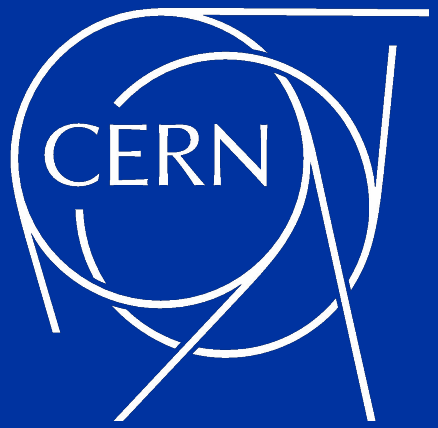
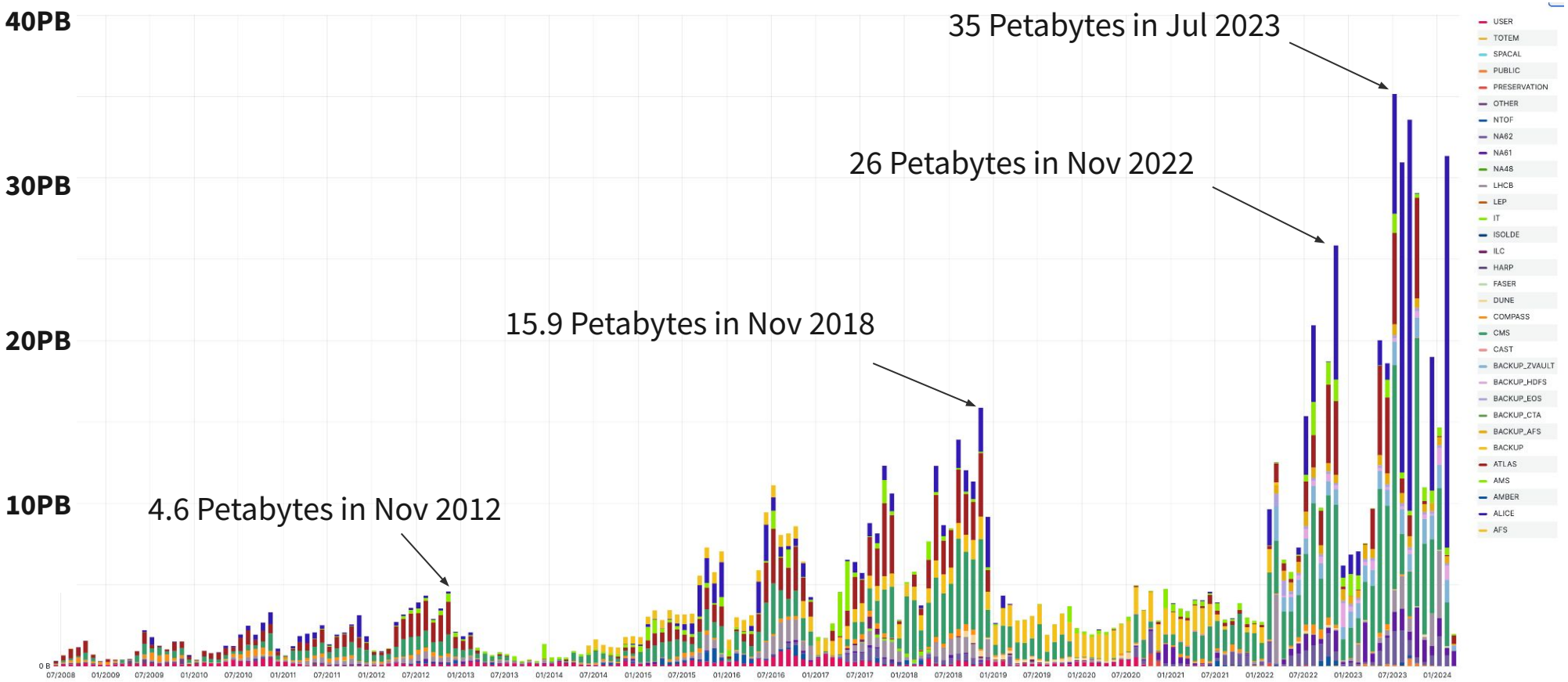


Improving CTA performance by splitting user and repack scheduler backends

Presented by:
João Afonso
CERN, IT-SD-TAB

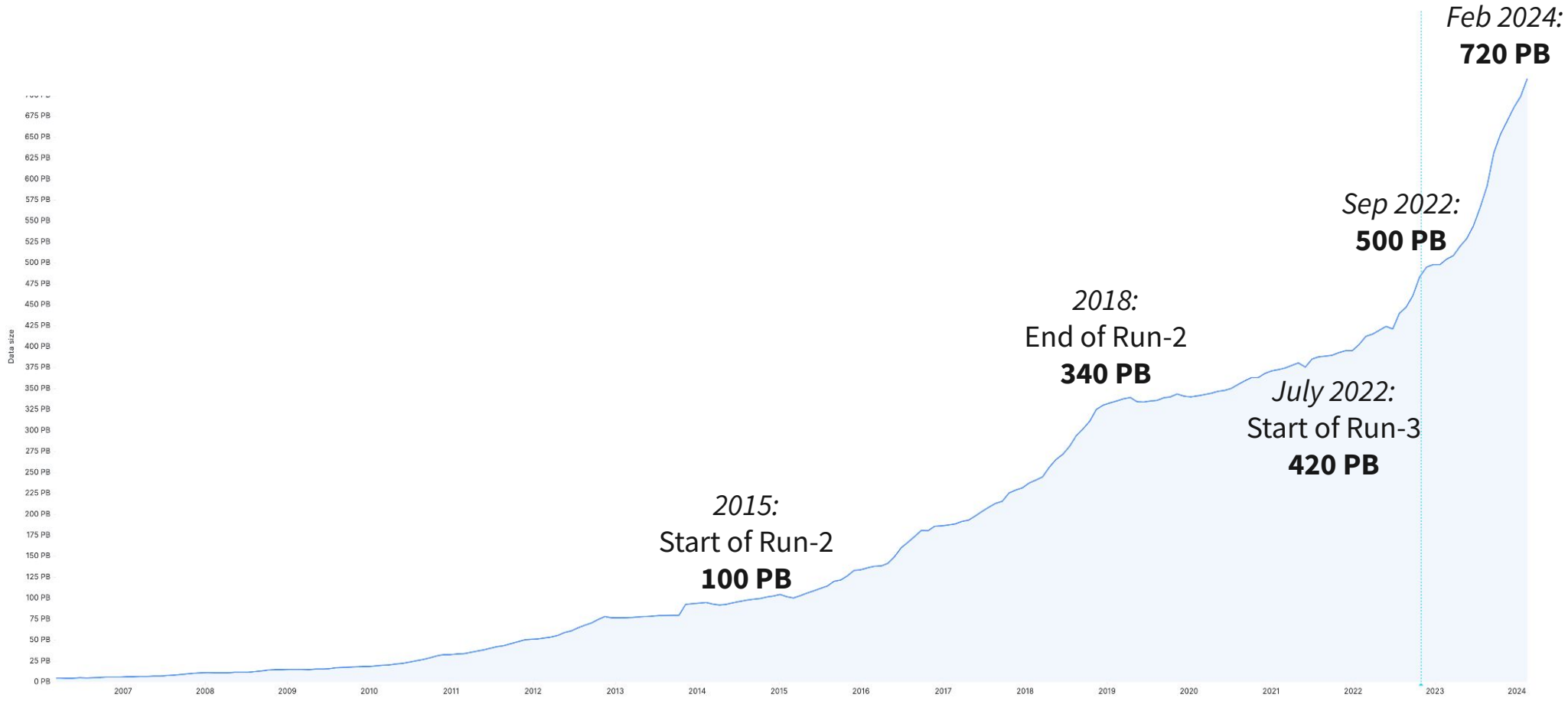
Data archival per month on CTA

- Amount of data to store increases every year



Accumulated data growth on CTA

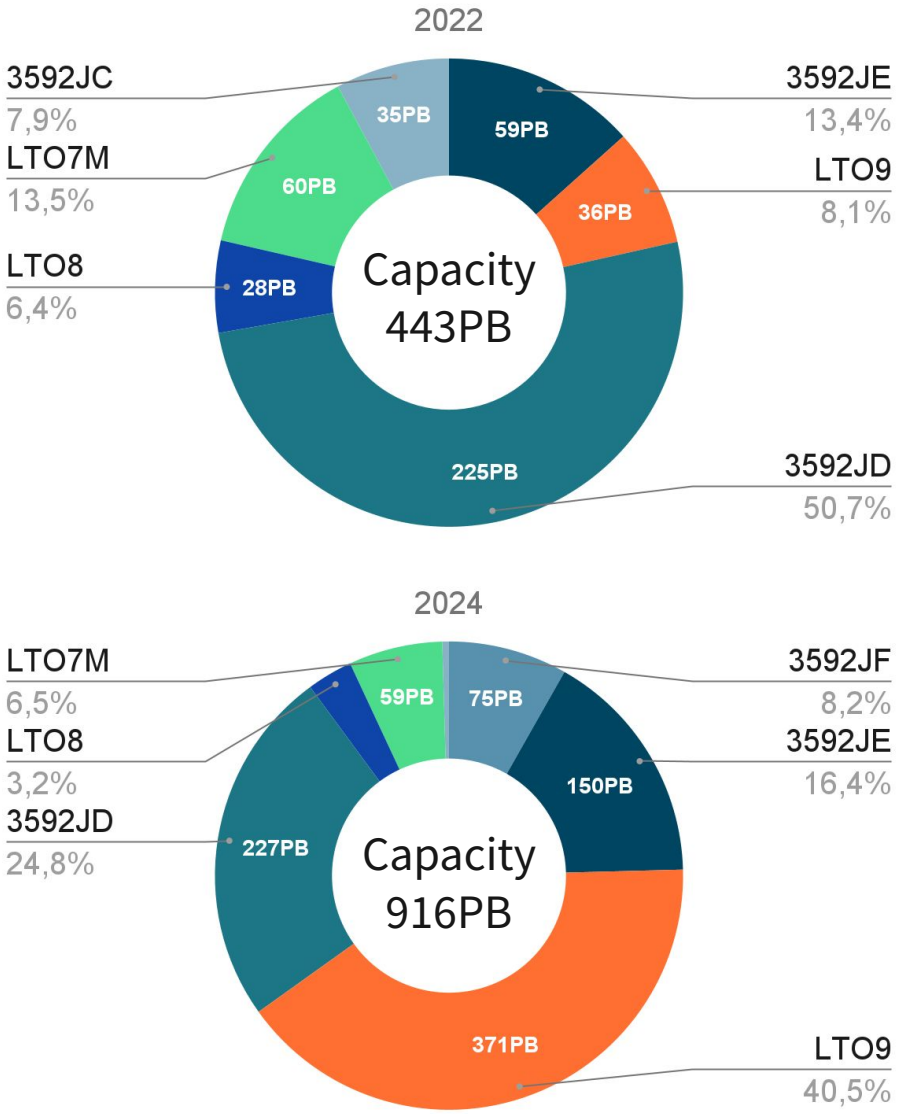
- +70% more data since start of Run 3
- Expected to increase further



Repack on CTA

- To match to the exponential increase of archival data, eventually we need to move it into newer, higher density media.
- The solution requires **repacking** older generation tapes in order to free slots for the higher density new generation media.

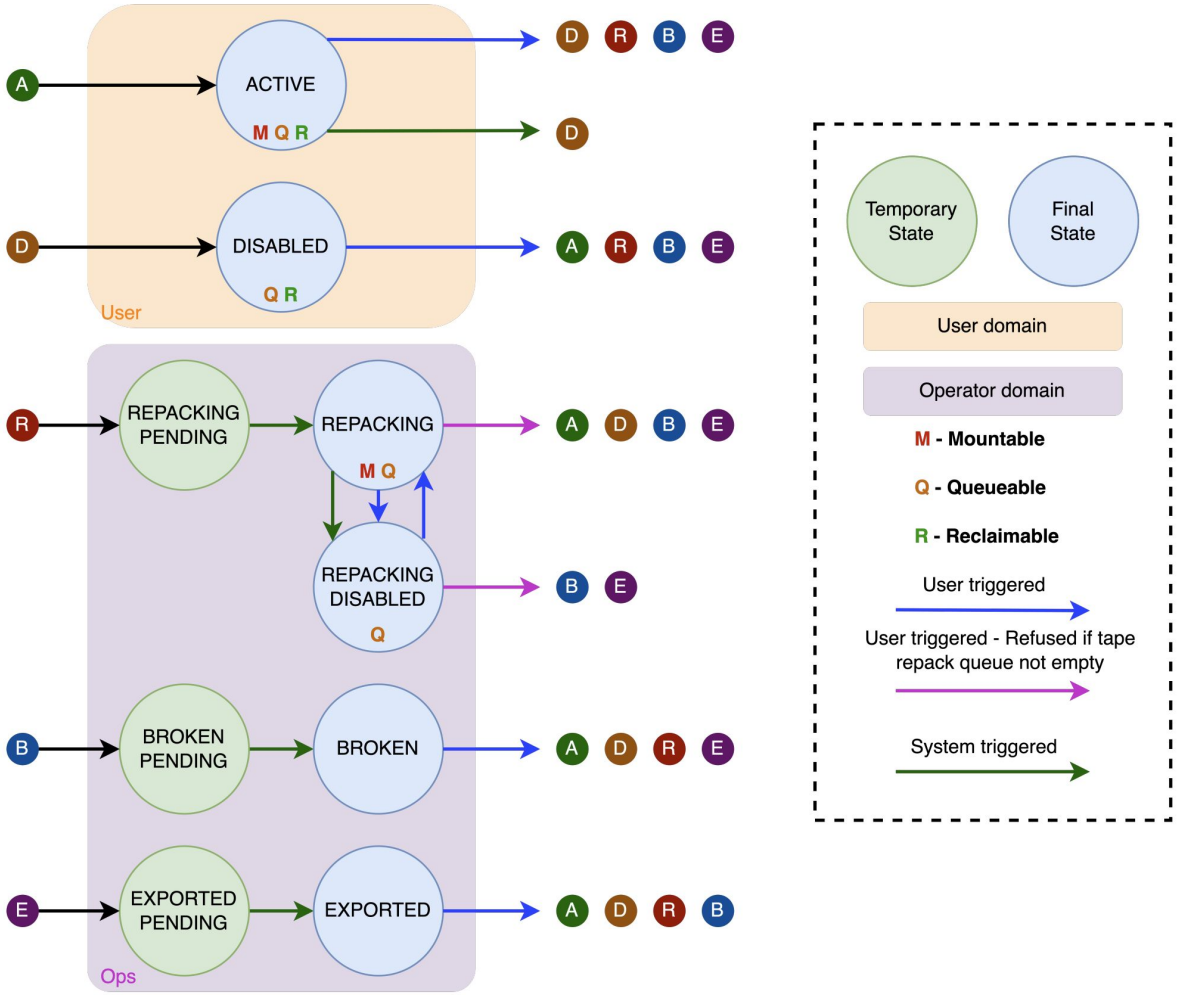
Model	Count	Capacity	Run 3
3592JF *New*	1504	50TB	Yes
3592JE	7499	20TB	Yes
LTO9	20618	18TB	Yes
3592JD	15125	15TB	No
LTO8	2410	12TB	No
LTO7M	6599	9TB	No
3592JC	604	7TB	No



New features and tools for repack operations

New CTA tape states

- Created new REPACKING, BROKEN and EXPORTED tape states.
- **Clear separation between user and operator domains**.
- Repacking can only be done on tapes with REPACKING state.
- Presented on EOS 2023 Workshop:
  - https://indico.cern.ch/event/1227241/contributions/5335996/



Dedicated Virtual Organisation for Repack

- CTA resources are allocated to the experiments by Virtual Organization (VO).

Problem:

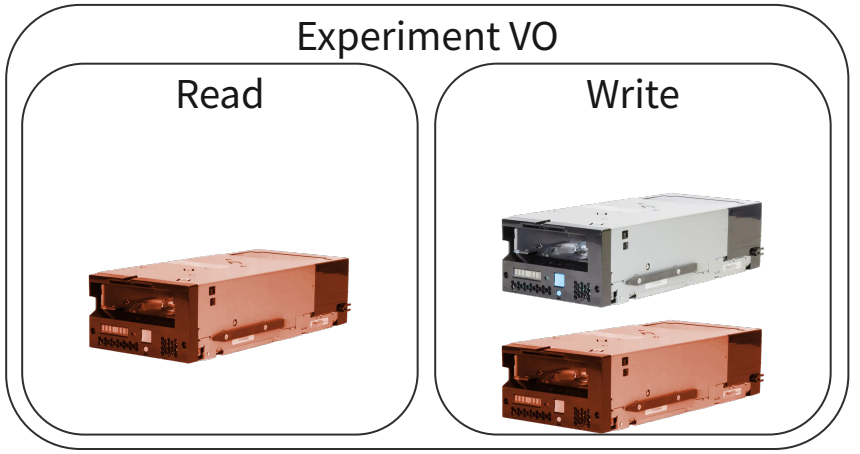
- Repacking was using drive quota that was meant for the experiments.
  - VO directly selected from user tape pool.
- As a result, effective user read/write throughput could be impacted by repack operations.

Solution:

- Operators can define a default VO for repack.
- This limits the number of tape drives for reading/writing repacking data, without affecting the user VO quota.

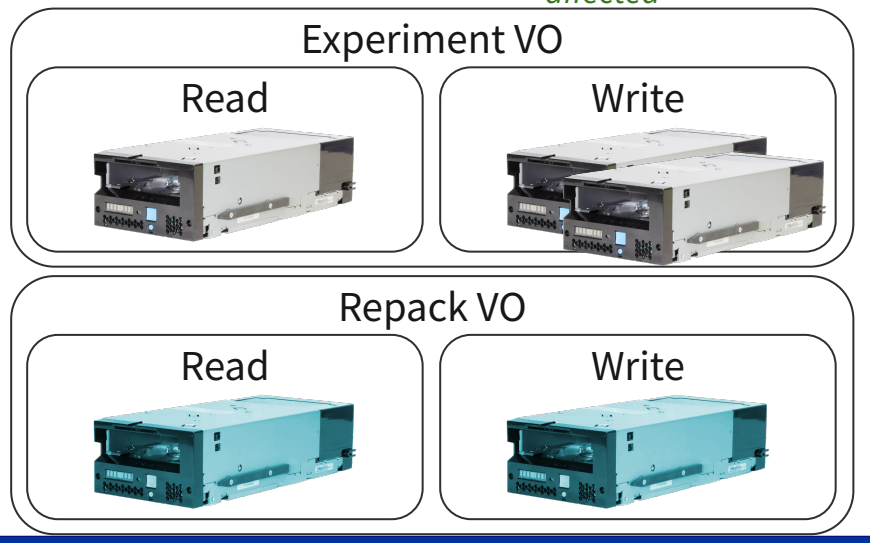
Repack would use same drives as experiments, blocking them from read and/or write

Before



User driver quota not affected

Now



Parameterize number of parallel repacks

Before:

- Number of parallel repacks was fixed to 2 tapes.

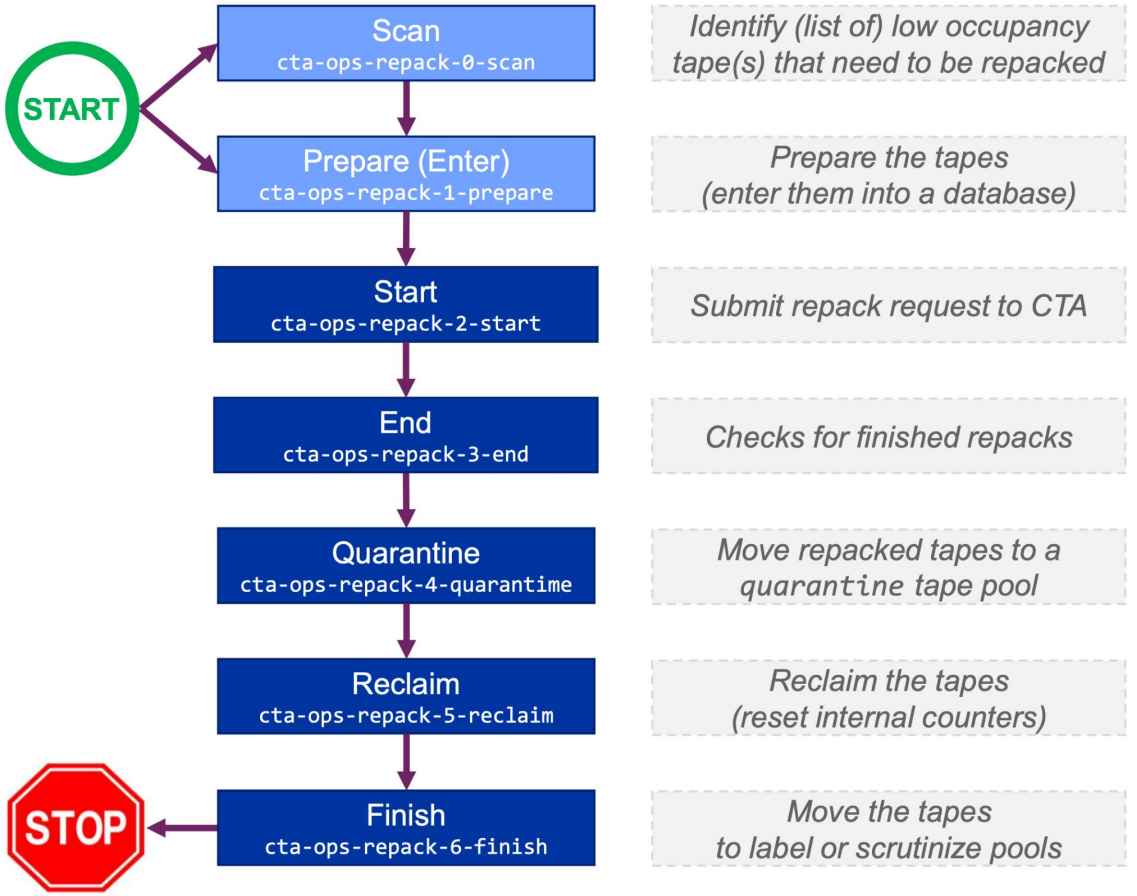
Now:

- Operators can configure, on the *cta-taped.conf* file, how many tapes can be expanded in parallel:
  - `# taped RepackMaxRequestsToExpand 20`
- Allows to scale up/down repack throughput, in accordance to available capacity.

ATRESYS

- **A**utomated **T**ape **RE**packing **SYS**tem
- Automatic orchestration of tape repacks
- Makes use of new REPACKING states.

- Presented on EOS 2023 Workshop:
  - https://indico.cern.ch/event/1227241/contributions/5366313/



Let’s start repacking!

Oops…

- With these upgrades we were ready to ramp up repacking!
- However, problems started pouring in and affecting user requests too…

Oops…

- With these upgrades we were ready to ramp up repacking!
- However, problems started pouring in and affecting user requests too…

“Unable to repack I43877 - message too long”

ops#1234 (8 Nov 2023)

```
[...] MSG="In BackendRados::atomicOverwrite, failed to assert existence or
write:
RepackRequest-Fronten
24-0-377467 Errno=90:
vid="I43877"
```

“Repack rm causes cta-frontend to crash”

ops#1258 (27 Nov 2023)

```
[root@ctaproductionfrontend02 ~]# cta-admin re rm -v I44010
231127 11:02:13 26098
/ctafrontend@localhost:
Error from XRootD SSI F
```

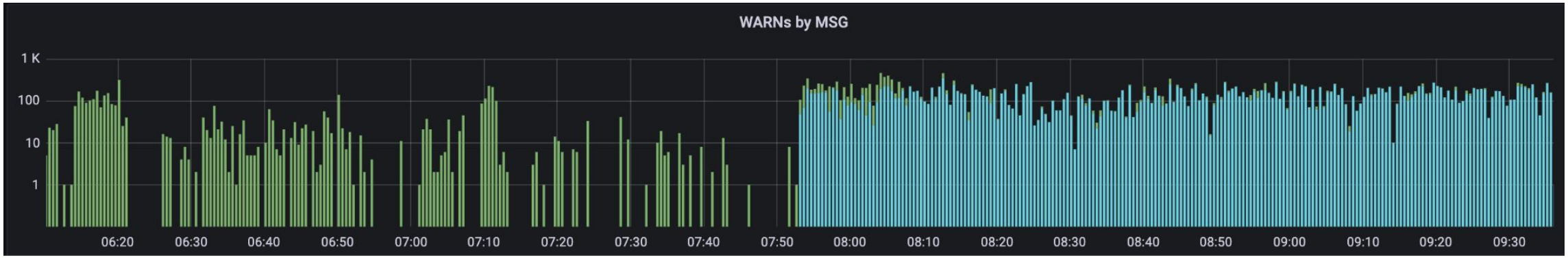
“Several lock-related errors in production taped servers”

ops#1186 (25 Sep 2023)

```
In OStoreDB::fetchMountInfo(): fetched a retrieve queue and that lasted more
than 1 second
```

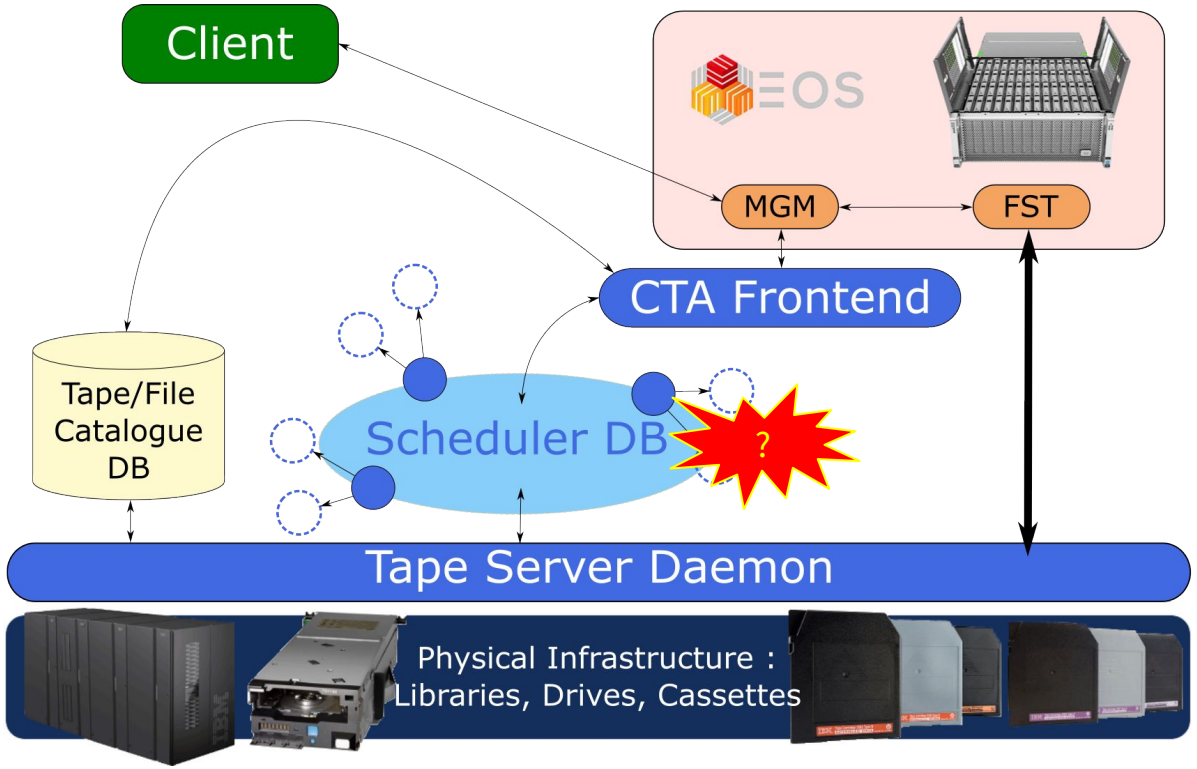
Oops…

- With these upgrades we were ready to ramp up repacking!
- However, problems started pouring in and affecting user requests too…



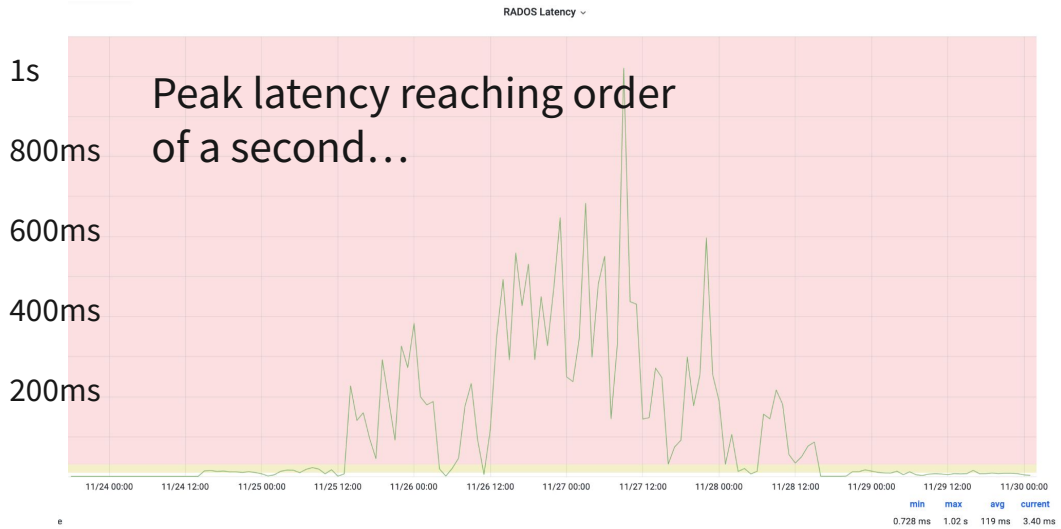
Investigation

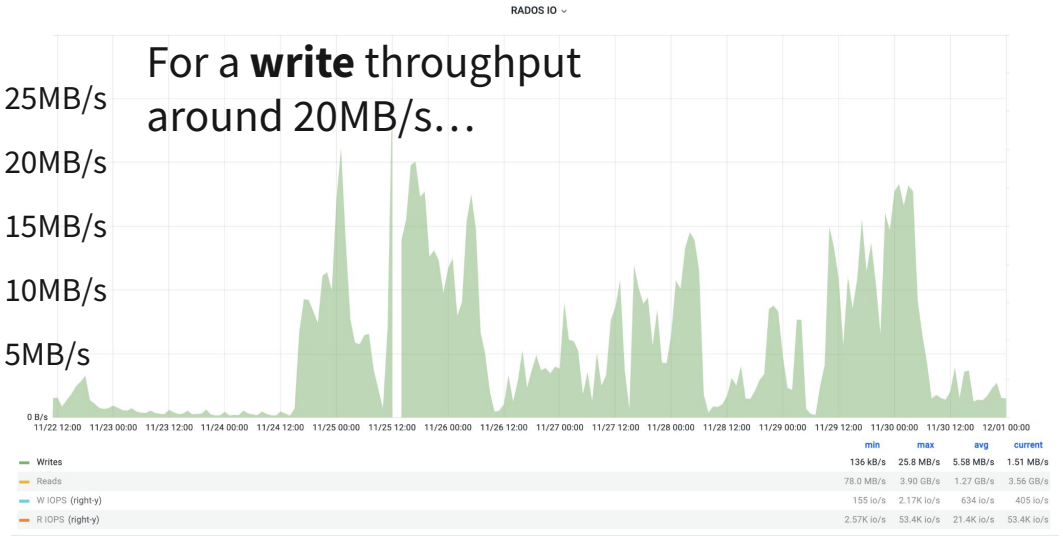
- These problems were clearly caused by performance limitations on the Scheduler DB.
- But understanding why exactly required an investigation…



Investigation

- The problem was related with the Ceph cluster, used by the Scheduler DB for object storage.
- Repack was causing a huge amount of data to be transferred to/from Ceph. This was overloading the cluster, which then became unable to respond quickly to new requests.
- This high latency was dragging the CTA frontend and tape servers, sometimes causing them to fail.





Investigation conclusion

We were hitting a bottleneck with repack, with real consequences on the health of CTA

Main reason:

- Repack objects (repack metadata) were reaching **sizes over 100MB**

1. Each repack object contains a list of all subrequests, one for each file on the repacking tape.
2. Existing tapes may contain millions of files.
3. This resulted into very large objects being created for each repacking tape.

4. **Several servers will try to modify these files – as part of the normal repack workflow – multiplying the bandwidth pressure over the Scheduler DB.**

Examples:

Tape	Nr files	Repack object size
L76199	2725278	~272 MB
I00146	2605639	~260 MB
I00837	2571847	~257 MB
I75773	2286214	~228 MB

Partial repack object dump:


```
{
  [...]
  "subrequests": [
  {
    "address":
"RepackSubRequest-Maintenance-tpsrv679.ce
    "fseq": "1",
    "retrieveAccounted": false,
    "archiveCopynbAccounted": [ 1 ],
    "subrequestDeleted": false
  },
  {
    "address":
"RepackSubRequest-Maintenance-tpsrv679.cern.ch-23471-20240304-11:22:43-0-10",
    "fseq": "2",
    "retrieveAccounted": false,
    "archiveCopynbAccounted": [ 1 ],
    "subrequestDeleted": false
  },
  {
    "address":
"RepackSubRequest-Maintenance-tpsrv679.cern.ch-23471-20240304-11:22:43-0-11",
    "fseq": "3",
    "retrieveAccounted": false,
    "archiveCopynbAccounted": [ 1 ],
    "subrequestDeleted": false
  },
  [...]
```


++100K's entries per repack request!

Path to solution

Mitigating the effects

- An quick fix was done by increasing the Ceph-Rados object limit from 100MB to a larger value.
  - This prevented the CTA scheduler from failing when a repack object was too big.
  - However, it does not solve the performance issues. May actually make them worse.

Limiting the number of files to repack per tape

- A medium-term fix was implemented that allows us to set a limit on the number of files to repack:
  - This allows operators to put an upper bound on the size of the repack object (for example 200K files → 20MB), keeping it under control.
  - As a result, repacking a tape may require several iterations until it's complete.

CTA frontend conf:

```
# cta.repack.repack_max_files_to_select 100000
```

Or cmd line:

```
> cta-admin re add --vid I12345 --maxfilestoselect 100000
```

Mid term solution

For an effective mid-term solution, we need to further isolate user requests from tape operations.

Solution:

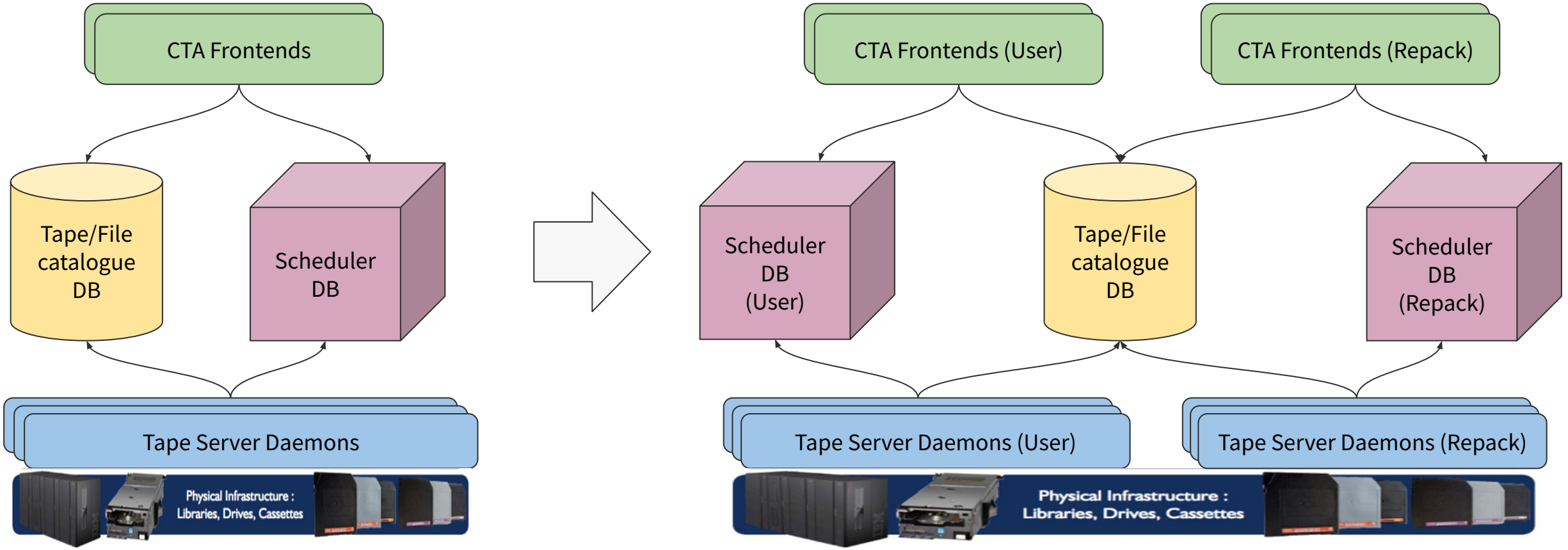
- **Split user and repack scheduler DB backends.**

✓ Guarantees that repack operations performance no longer impacts user requests.

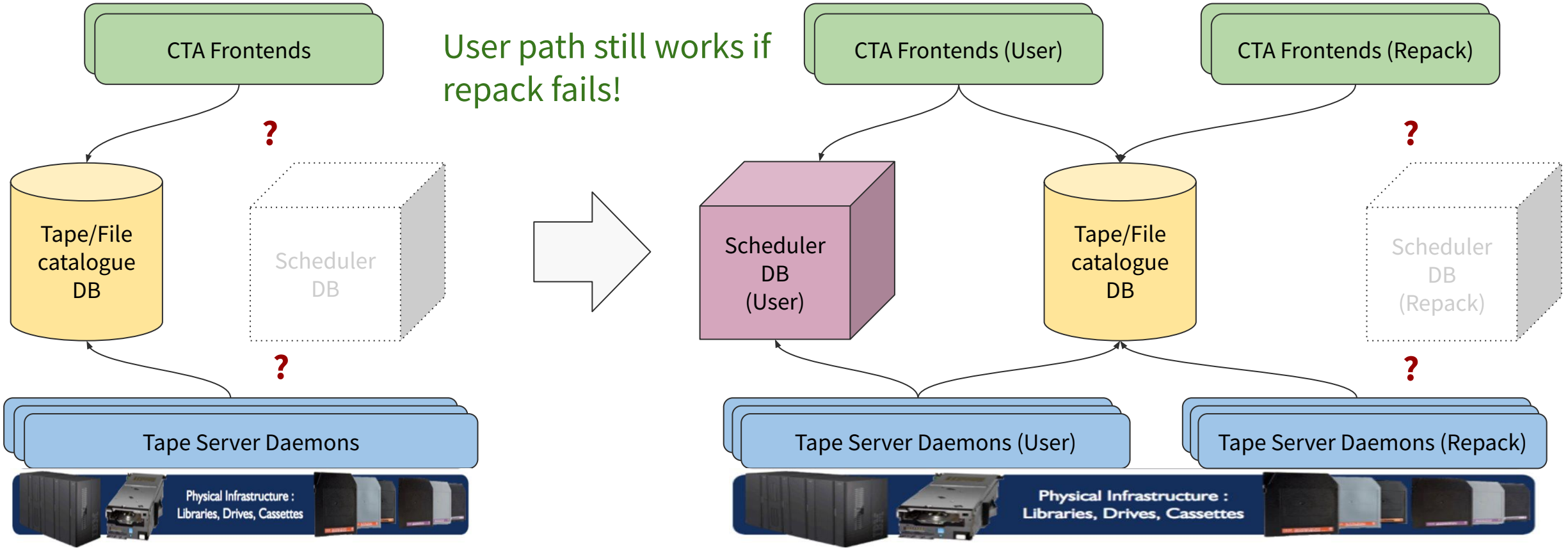
✓ In practice, the repack scheduler DB can go down without affecting user requests.

✓ By having a separate Scheduler DB instance for repack, we can use it to test the new Postgres Scheduler DB, without any risk for the users.

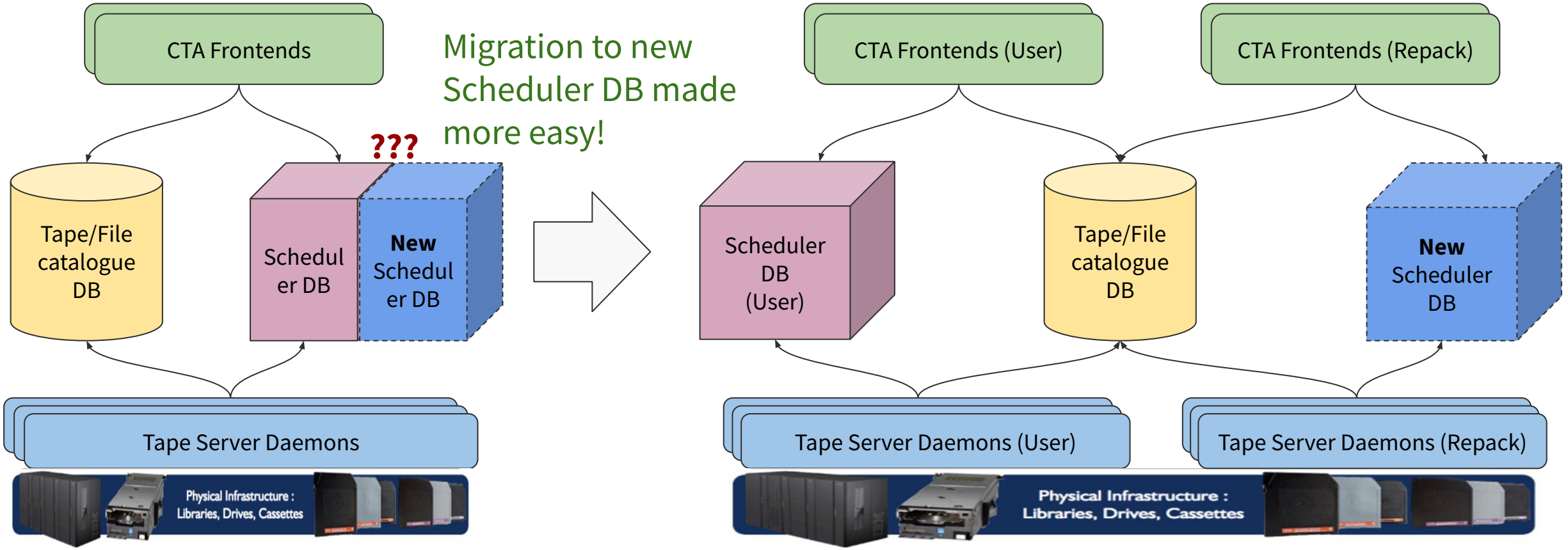
New architecture



New architecture



New architecture



Current work

This required a coordinated work between development and operations:

- **Development:**
  - Mechanism to switch between user and repack object store configurations. ✓
  - Enable the disabling of repack requests on the user side and of user request on the repack side. ✓

- **Operations:**
  - Setup a new repack Scheduler DB instance, on a new Ceph cluster.
  - Migrate servers for the new configuration.

To be ready for the LHC run

Results coming soon!

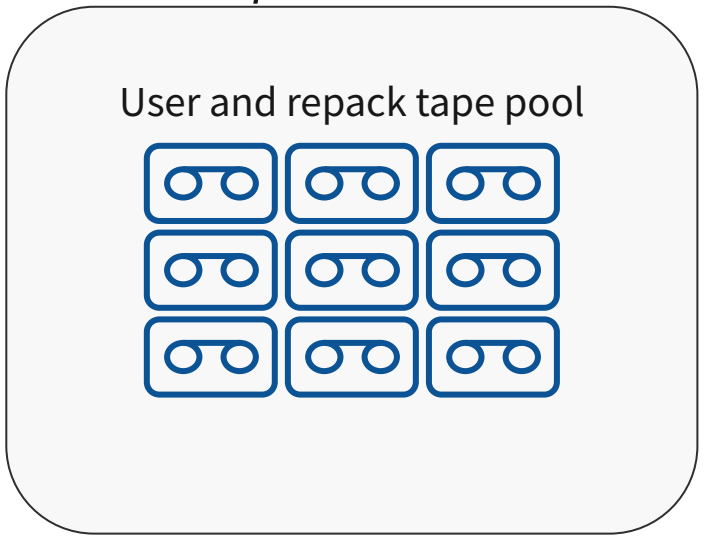
Future work (draft)

Separate archive repack tape pools from user tape pools

- *TBD on a future release*

Now

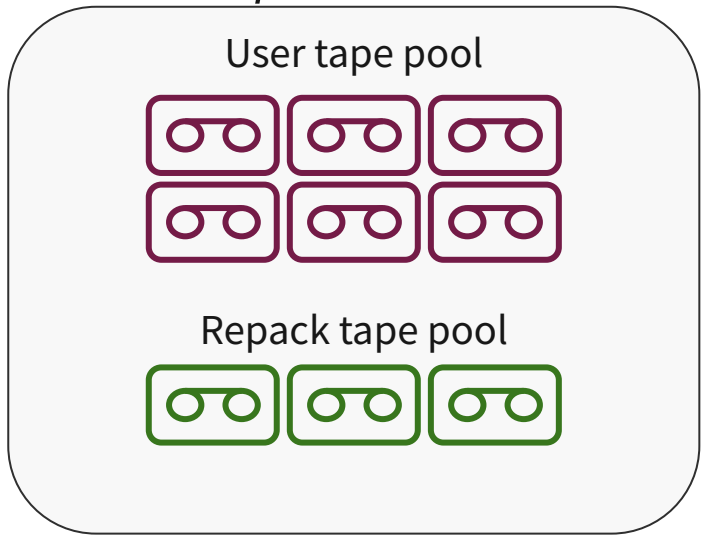
Experiment A



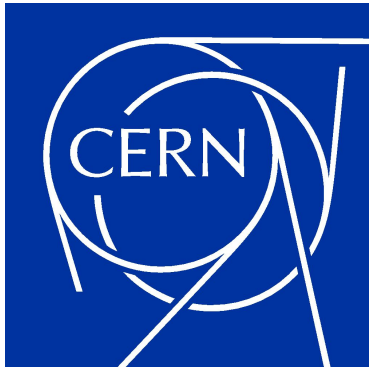
- Same tape pool for user and repack data archival
- Old and new data can get mixed → Bad data colocation

Future

Experiment A



- Separate tape pools for user and repack data archival
- Old and new data do not get mixed → Better data colocation



home.cern