Conference on Computing in High Energy and Nuclear Physics
19-25 October 2024
Kraków, Poland

Evolution of the CERN Tape Archive scheduling system

Dr. Jaroslav Guenther (CERN)
on behalf of the CERN IT CTA Team

Scheduling Evolution in CERN Tape Systems

~1990 SHIFT — manual scheduling

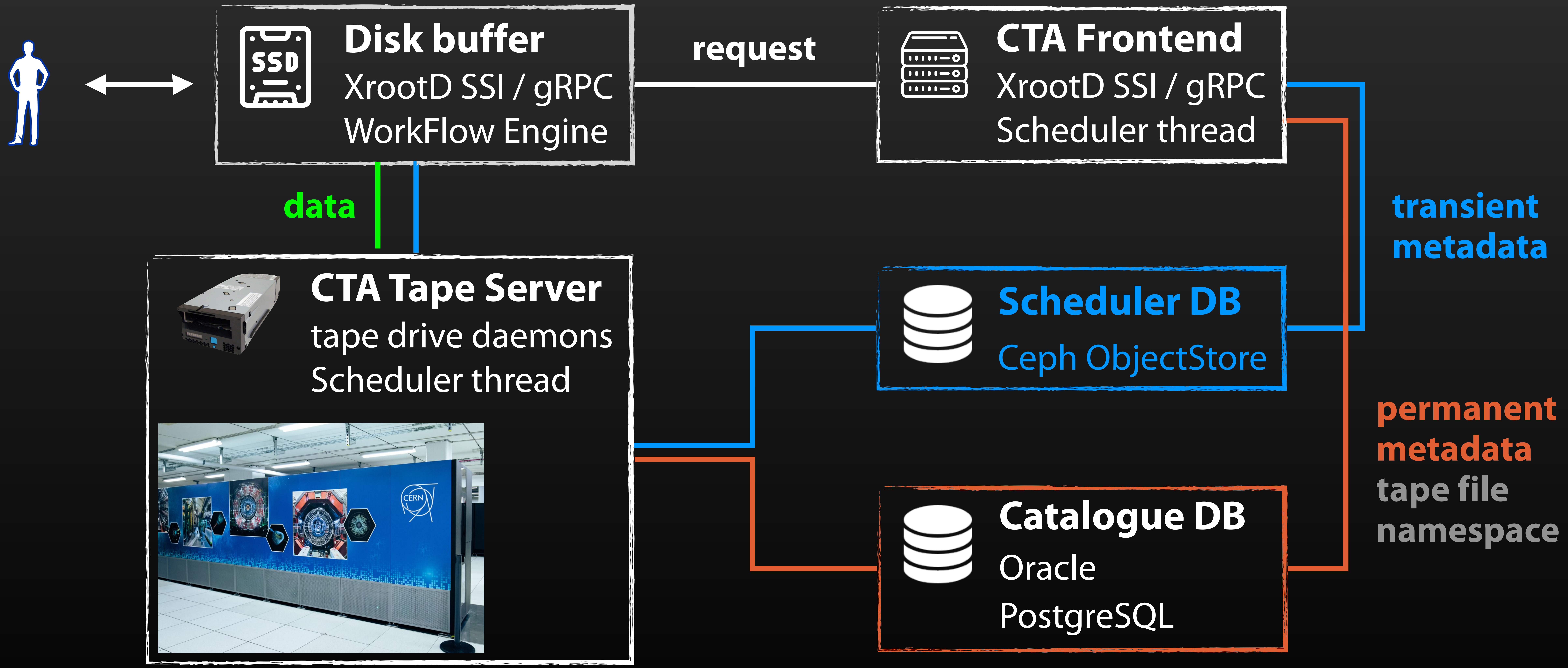
~1998 STK libraries — robotic scheduling, namespace → CERN IT
repack introduced !

next ~22 years CASTOR — centrally deployed scheduling

~2020 CTA — distributed multi-threaded scheduling
using ObjectStore technology

~ 2024
preparation for Run 4 — CTA adopted by wider community !
new challenges ahead

Scheduling DB Stores Transient Metadata



Scheduler DB Implemented as ObjectStore

Architecture

- motivated by performance for archival and retrieval queueing operations
- multi-threaded interface to Ceph
- protobuf serialised objects in key/value store
  (archive, retrieve requests and queues)
- code design ensures
  - high performance
  - scaling
  - reliability
  (despite > storage round trips than DB)
- delivered very well for Run 3

Motivation for Scheduler DB Evolution

Development Cost

- complex distributed transaction management system
- high maintainability cost
  - additional dependency
  - requires extensive learning effort
- scheduler logic is tightly coupled to ObjectStore implementation
  - lack of indexes is a serious constraint on implementation of scheduler algorithms (e.g. cancelling requests)
  - as we scaled up to ~200 tape drives, global locking caused scheduler contention
- workflows beyond original design proven difficult

but →

design allows multiple backends

and

code complexity reduction possible

and

we have off-the-shelf solution = Relational DB

Motivation for Scheduler DB Evolution 2/2

Operational Cost

- ObjectStore requires additional technology expertise → **we can consolidate on common technologies**
  - sites which do not run Ceph
  - FTS scheduling using Postgres
- complex backend object structure
  - object introspection, forensics and cleanup is difficult
  - "schema" updates difficult to manage
- high priority fixes still required several times a year (e.g. for object deletion, empty shard handling, infinite loops, global locking issues, repack exhausting resources, object size handling, etc.)

and

have Relational DB rows for operators

and

simpler operational tools

Advantages of Relational DB as the Solution

<u>New Scheduler DB</u>

- lower code complexity
  - no overhead of transactional management code

    MVCC, indexing (+sync) "for free" from DB
- straightforward schema updates
- multi-index queues → more flexibility to improve and extend Scheduling algorithms
- extensible to multiple database backends

  (PostgreSQL DB, Oracle DB)
- no global scheduling lock

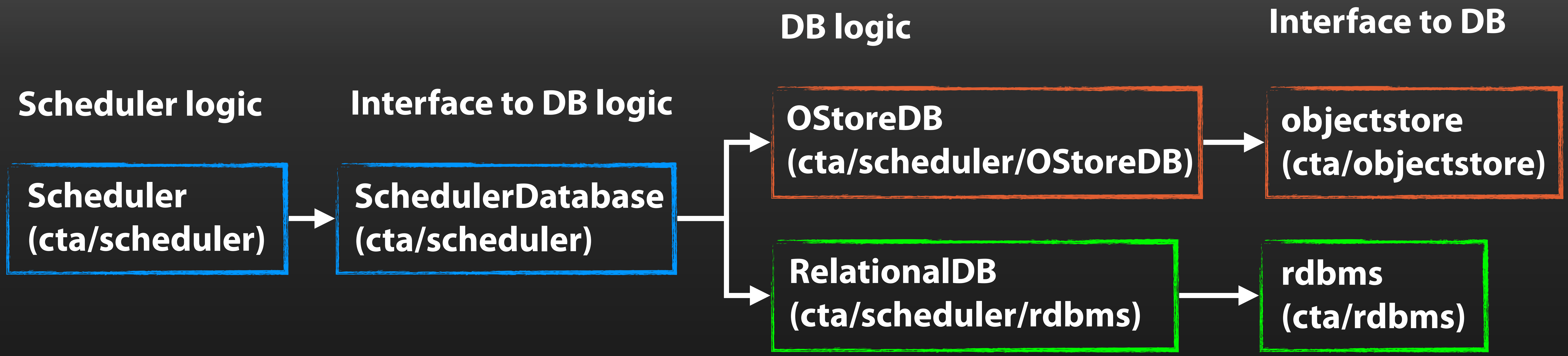
Overview of Scheduling Workflow 1/2

blue = scheduler DB tasks

Overview of Scheduling Workflow 2/2

blue = scheduler DB tasks

Scheduler DB Implementation Architecture

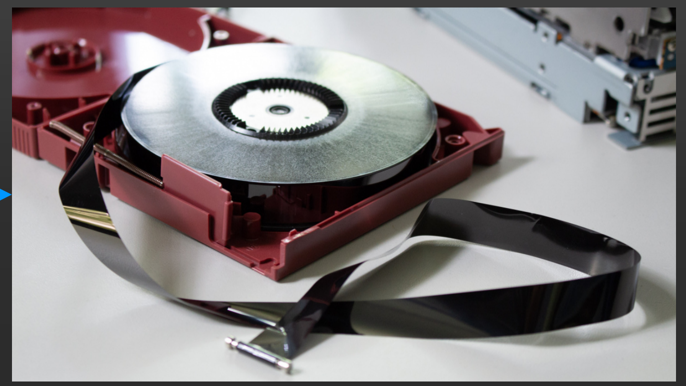


<u>**Design Allows Multiple Scheduler DB Implementations**</u>

- **use CTA generic RDBMS interface for Postgres DB implementation**
- **workflow oriented tables (Archive/Retrieve/Repack), views, sequences, etc.**
  (file transfer job = row in a table)

Finished Implementation of Archival Workflow

Disk buffer



<u>**Archival**</u>

- ✓ queueing
- ✓ job summaries per mount decision
- ✓ job fetching for transfer
- ✓ management of failures and retries
- ✓ transfer status reporting
- ✓ improvements in CTA rdbms layer

<u>**Lower Granularity Locking**</u>

- ✓ drive deciding to mount a tape
  - lock per logical library
    (prevents empty mounts)
- ✓ fetching the jobs from the queue
  - lock per tape pool per workflow
    (avoids interwoven row sets per drive)
- ✓ no global lock on scheduling anymore !

Functional Testing of Archival Workflow

Setup

- Full System deployment in Minikube (CTA+EOS)
  - 1 disk for buffer (EOS MGM & FST)
  - 1 disk for tape drive
- External Catalogue: Oracle DB
- External Scheduler: PostgresDB

Load

- 1M files, 128 B each
- 1 tape pool (30 MB tapes)

Kubernetes deployment for performance and scalability tests is in works !

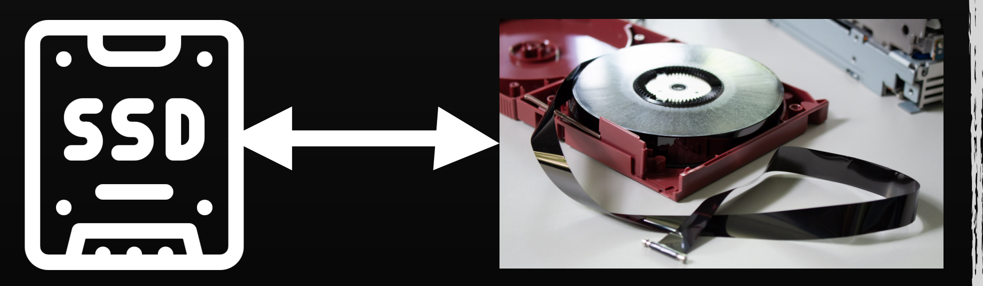
Functional Test of Archive Scheduling

Scheduler DB

fetching jobs →

Tape Server Mount

transferring data →

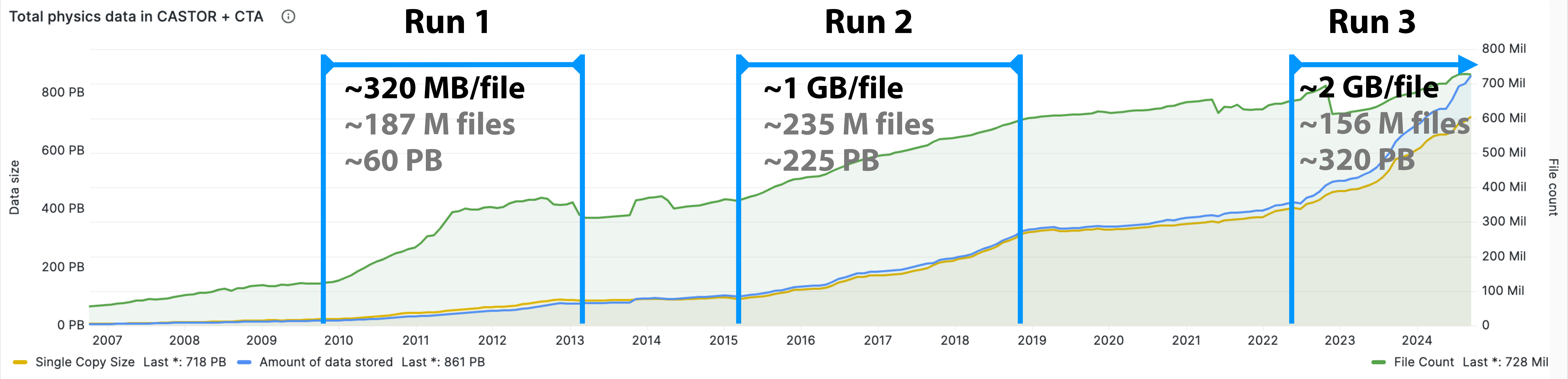


Queueing requests

(@ 400 Hz with 1 disk for r/w)

De-queueing only

Minimal File Rate Requirements



Conservative estimates

- file sizes grow

  → keeps minimal rate down

	RUN 3	RUN 4
CTA SLA	50 GB / s	125 GB / s
Avg. File Size	2 GB	2 GB
Avg. Throughput	25 Hz	**63 Hz**

Postgres DB Scheduling

several times faster than ~ 100 Hz = peak rate of Run 3

Implementation and Deployment Roadmap

Implementation:

- Retrieve and Repack

Code Optimisation:

- Performance & Scaling Tests

Deployment:

- Kubernetes Test Cluster
- Repack Instance
- Production

Q4 2024 | Q1 2025 | Q2 2025 | Q3 2025 | Q4 2025 | Beyond Q1 2026

Not planning to replace ObjectStore on a timescale of Run 3

Summary

Next Scheduler DB Evolution

- ✓ meets with Run 4 challenges
- ✓ satisfies the needs of larger CTA Community
- ✓ decreases future development and operational costs

Implementation Status

- ✓ PostgreSQL as CTA Scheduler DB for Archival Workflow is functional
- ✓ Retrieve and Repack are coming next
- ✓ improved locking granularity

Long Term Prospects

- ✓ tape supply logic part of scheduling
- ✓ refactoring and cleaner handling of several workflows: (leveraging DB features)
  - request/file deletion
  - multiple retrieves per file
  - priority queues

Thank you !

Thank you for your attention

… special Thanks to few of my CERN colleagues for their help
(David S., Elvin S., Joao A., Julien L., Michael D., Steven M., Pablo O. C., Vlado B.)

I welcome any questions or comments !

Backup

Tape Infrastructure (September 2024)

CERN Tape Archive

- Archive of the physics data
- Provisioned capacity: ~1.18 EB

- Libraries:
  - 4 x IBM TS4500
  - 2 x Spectra Logic TFinity
- Drives:
  - 40 x TS1170, 46 x IBM1160
  - 88 x LTO-9, 10 x LTO-8
- Media:
  - 150 PB on 3592JF, 150 PB on 3592JE, 227 PB on 3592JD
  - 551 PB on LTO-9, 17 PB on LTO-8, 59 PB on LTO-7M

IBM Spectrum Protect

- Backup of the business data
- Licensed capacity: ~15 PB

- Libraries:
  - 1 x IBM TS4500 (partitioned)
  - 1 x Spectra Logic TFinity (partitioned)
- Drives:
  - 10 x LTO-9
  - 10 x LTO-8
- Media:
  - 12 PB on LTO-8
  - 11 PB on LTO-7M

New Scheduler DB: Relational DB

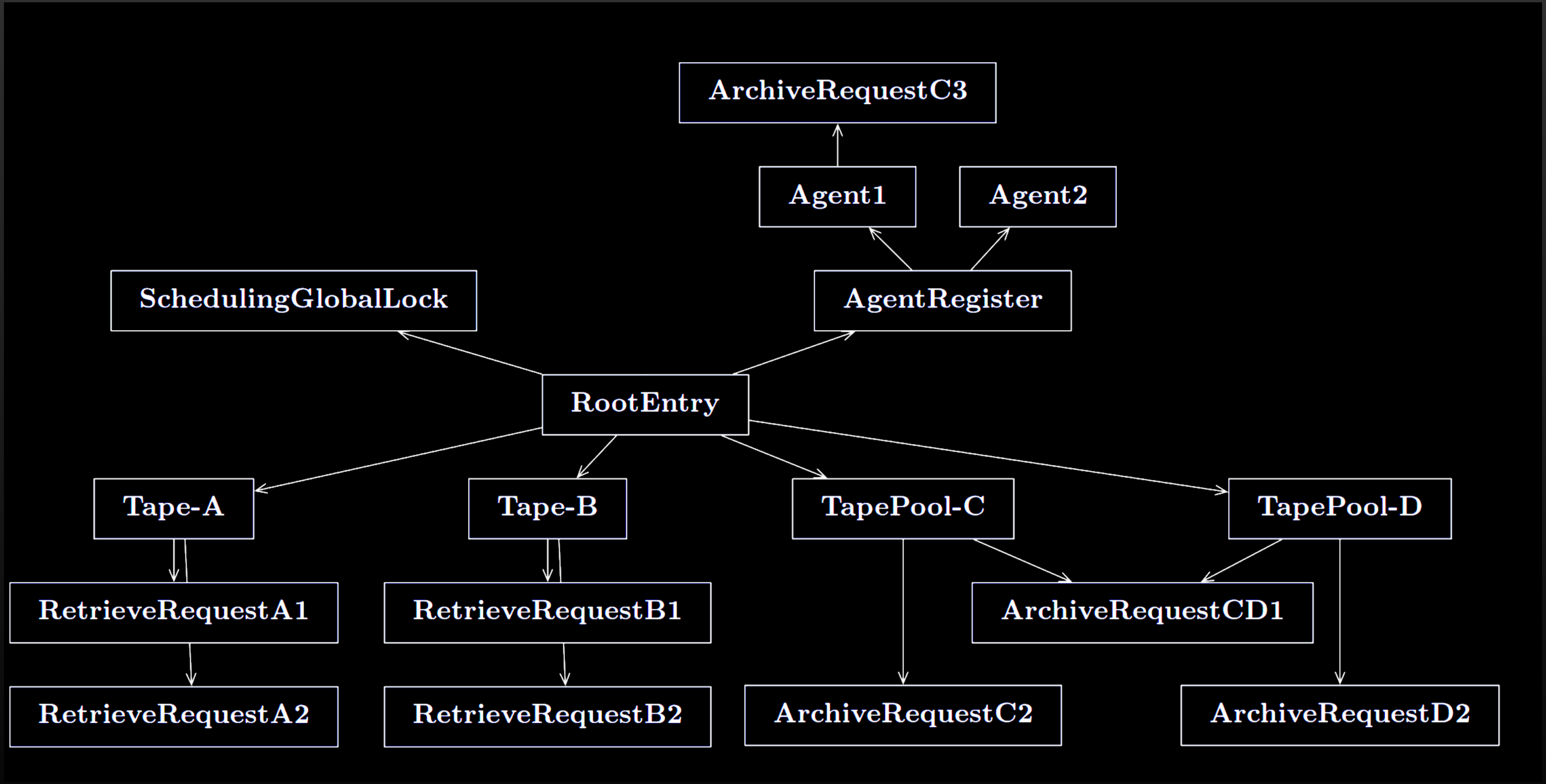
Challenges

- independent solutions for ObjectStore-coupled scheduler logic
- high performance, reliability and scalability IF
  - DB features exploited smartly !

    (e.g. not counting all rows for every query)
  - requires optimisation efforts per use-case
  - relies on diligence of developer with DB queries and DB configuration
- LHC Run 4 !

Scheduler DB Implemented as ObjectStore

Architecture



IO Limited Sanity Check

1 thread queueing

2 threads writing to tape