

# Rucio DB overview

Cédric Serfon (BNL)

November 10, 2022

# Introduction

- We have now many instances of Rucio serving multiple communities
- All these instances are operated on different databases
- The purpose of this talk is to give an overview of the different deployments. Aims at :
  - Helping new communities in making some decisions (e.g. backend, partitioning, etc.)
  - Identifying common issues, potential bottlenecks
- Thank you to all the communities that provided feedback

# DB backends

- Thanks to SQLAlchemy, Rucio supports multiple backends
- Regarding current deployment, mostly Oracle and PostGres are used :
  - Oracle : ATLAS, CMS
  - PostGres :
    - 11 : SKAO
    - 12 : Belle II
    - 14 : FNAL/Rubin/DUNE
- FNAL/Rubin/DUNE :
  - FNAL DBs are hosted by Postgres DB provision service, with multiple Rucio databases residing on a single cluster.
  - Rubin's databases are deployed with CloudNativePG for K8s
- From the communities who replied, none use MySQL or MariaDB. Anyone in the room ?

# Size of the DB

- Very different range if one compares the number of rows of DID table. To be fair, some collaborations (\*) are in data taking mode, whereas others not yet :
  - ATLAS\* : 1.3B
  - Belle II\* : 104M
  - CMS\* : 92M
  - Dune : 3.3M
  - SKAO : 266k
- Table partitioning :
  - ATLAS : All “active” tables partitioned by scope, archived tables partitioned by time
  - CMS : History table + bad\_replicas tables partitioned
  - Belle II : History tables recently partitioned

# History tables

- There are a few history tables (most of the name are self-explanatory) :
  - deleted\_dids
  - requests\_history
  - contents\_history
  - account\_usage\_history
  - rules\_hist\_recent and rules\_history
  - messages\_history
  - subscriptions\_history
- More are defined into model.py (e.g. replicas\_history) but are not filled by any workflow

# History tables

- If nothing is done, these tables can only grow and occupy a significant fraction of the DB space
  - They can be useful for debugging and to prevent reuse of DID names (deleted\_dids)
  - Common practice : They are partitioned and old partitions are dropped
- Some requests :
  - Have the possibility to disabled these tables if wanted
  - Evaluate if it's possible to have the history tables living on a different DB than the active one and still can use them for API/CLI (e.g. list-rules-history)

# Scopes

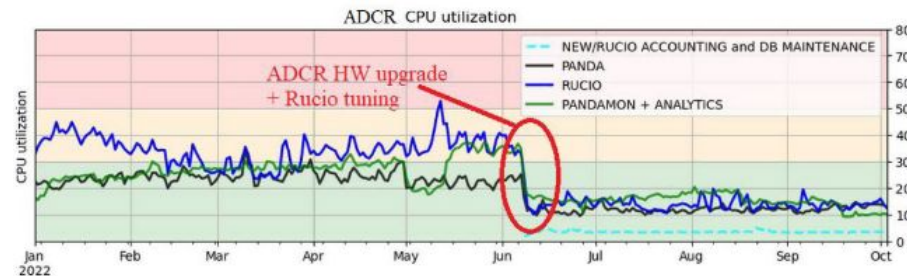
- Scopes are used to partition the namespace. Here again, large range of numbers
  - ATLAS : 8345
  - Belle II : 940
  - CMS : 10001
  - Dune 37
  - SKA : 9
- Big numbers of scopes are driven by number of users
- Number of DIDs in each scope can be very unbalanced e.g. in ATLAS :
  - 2 biggest scopes have 274M and 237M DIDs
  - More than 3k without dids

# Rules

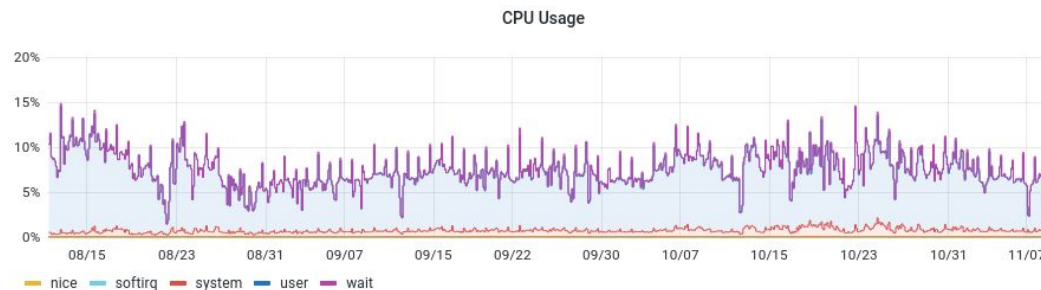
- Number of rules - #dids/#rules ratio :
  - ATLAS : 10M - 130
  - Belle II : 9M - 11
  - CMS : 7M - 13
  - Dune : 267k - 12
  - SKAO : 95 - 2.8k
- Surprisingly the number of rules for the experiments already taking data is of the same order of magnitude

# DB performances

- No communities reported any bottlenecks
- From ATLAS experience :
  - Rucio scales for at least O(1B) dids, O(10M) rules
  - Recent DB hardware + Rucio tuning (e.g. move to temporary table) allowed to significantly reduce the CPU utilization



ATLAS



Belle II

# Things to improve

- We have no recommendations how to setup a DB for the new communities (e.g. partition your tables from the beginning)



- Some collaborations use PL/SQL procedures to perform accounting or other operations (e.g. fixing some inconsistencies). We should try to collect/share them

# Acknowledgements

Thanks to everyone who provided information : Brandon White, Eric Vaandering, Hironori Ito, Mario Lassnig, Rob Barnsley, Yuyi Guo, Boromir