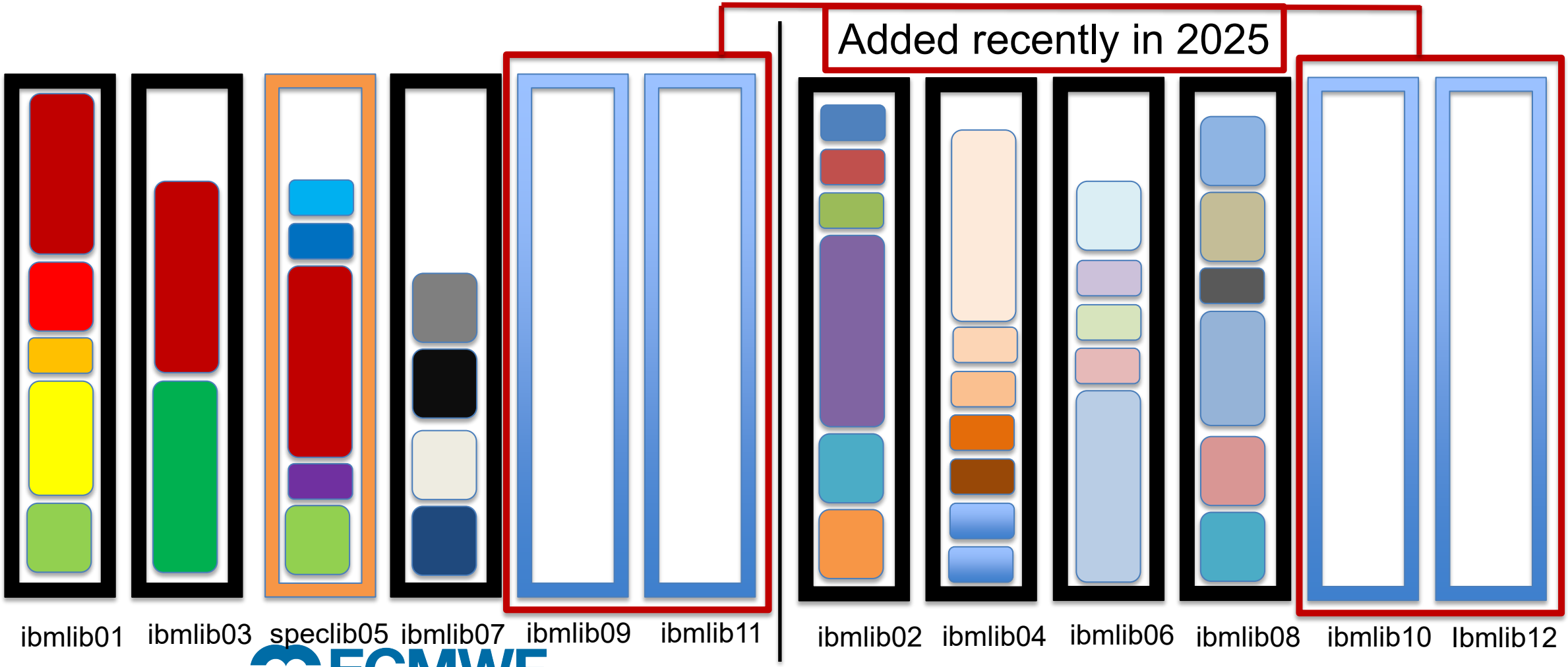
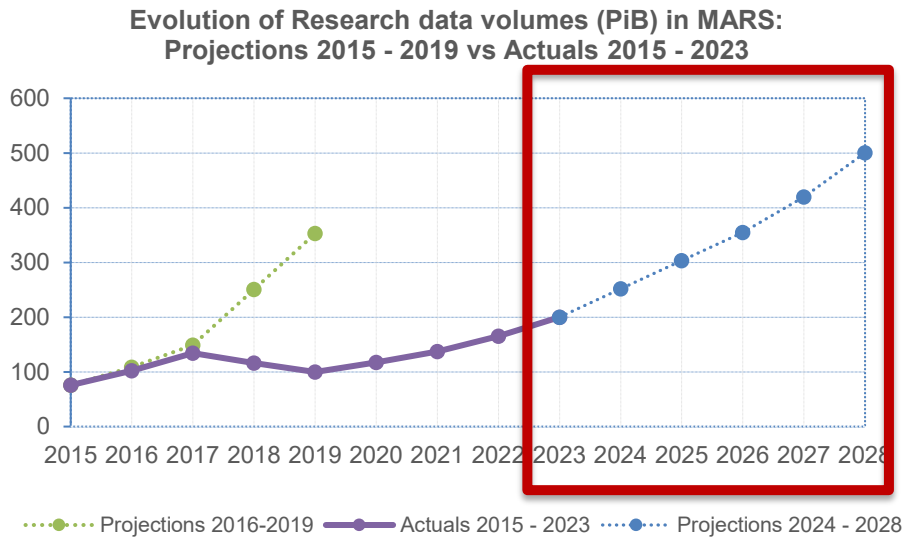
Load balancing accesses to tape libraries

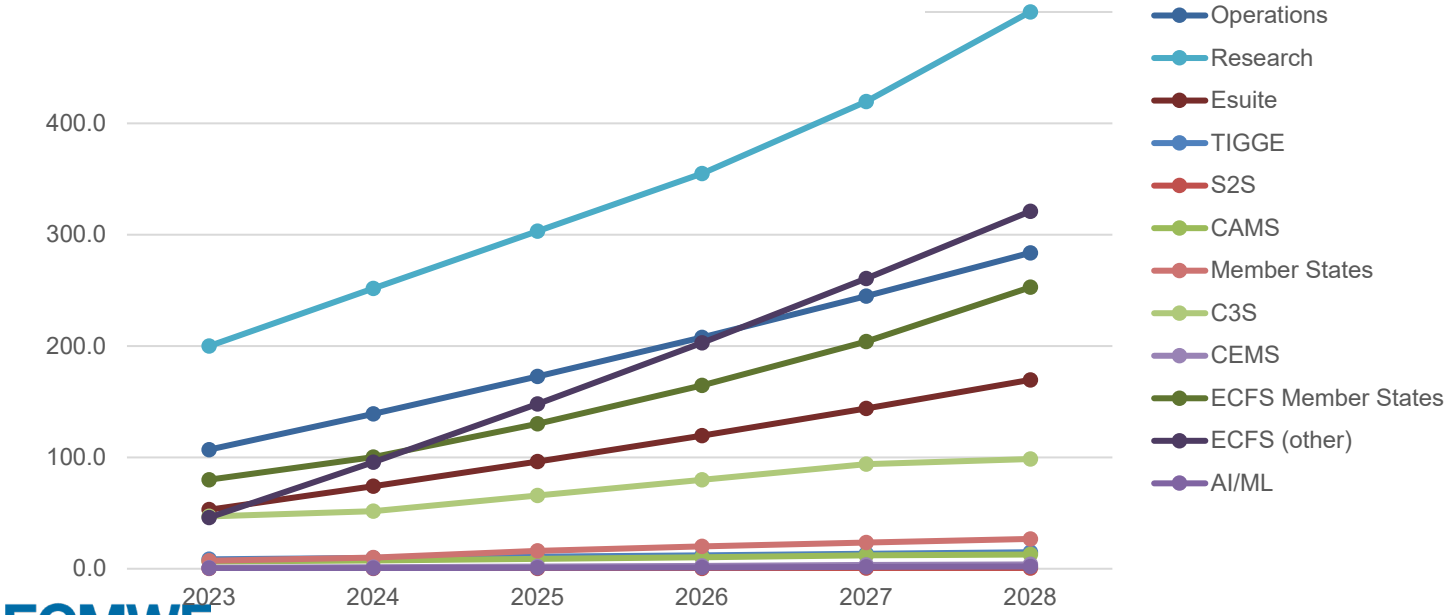
Sebastien Denvil and ECMWF colleagues.



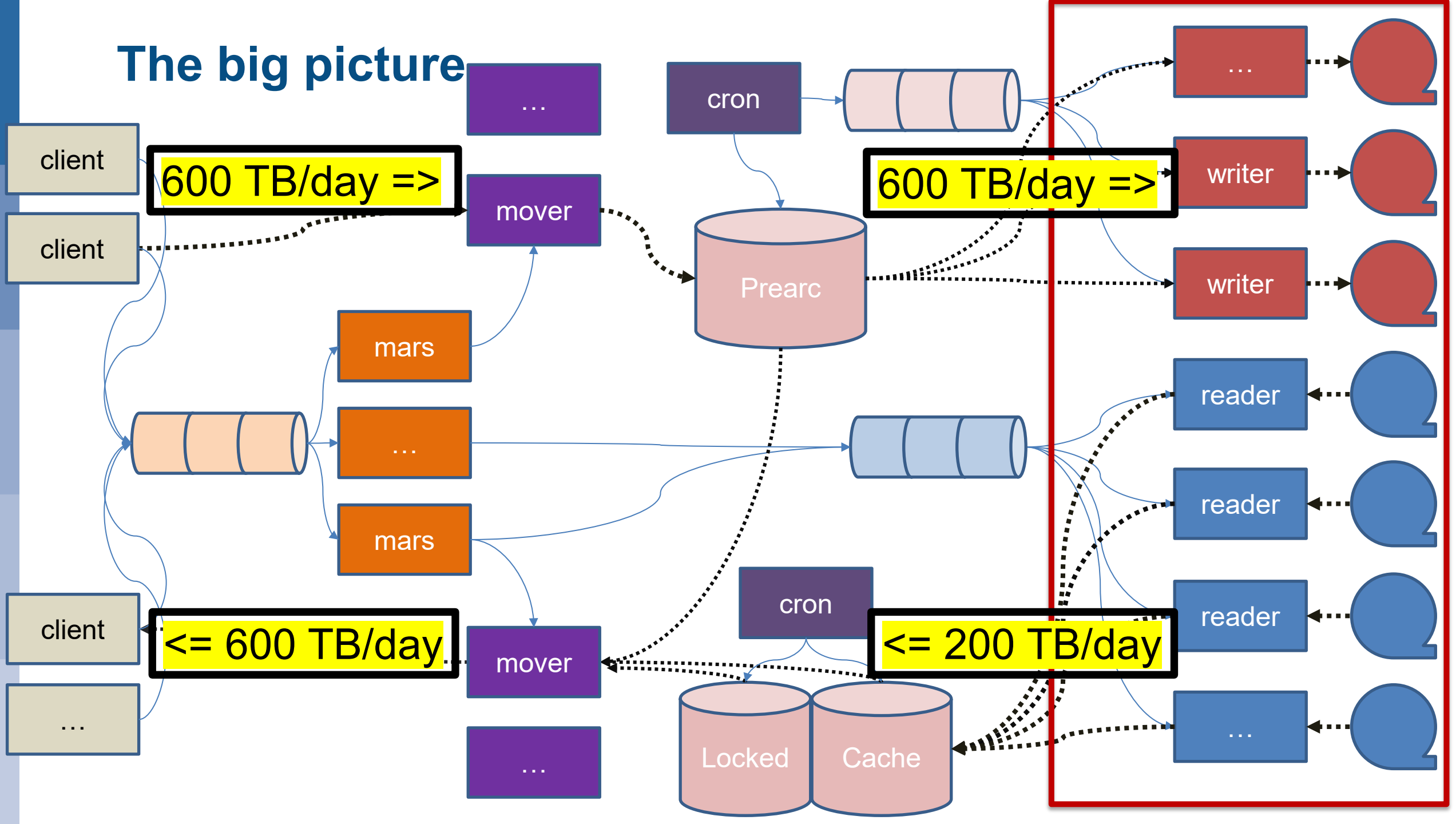
ECMWF

Archive holdings and projections



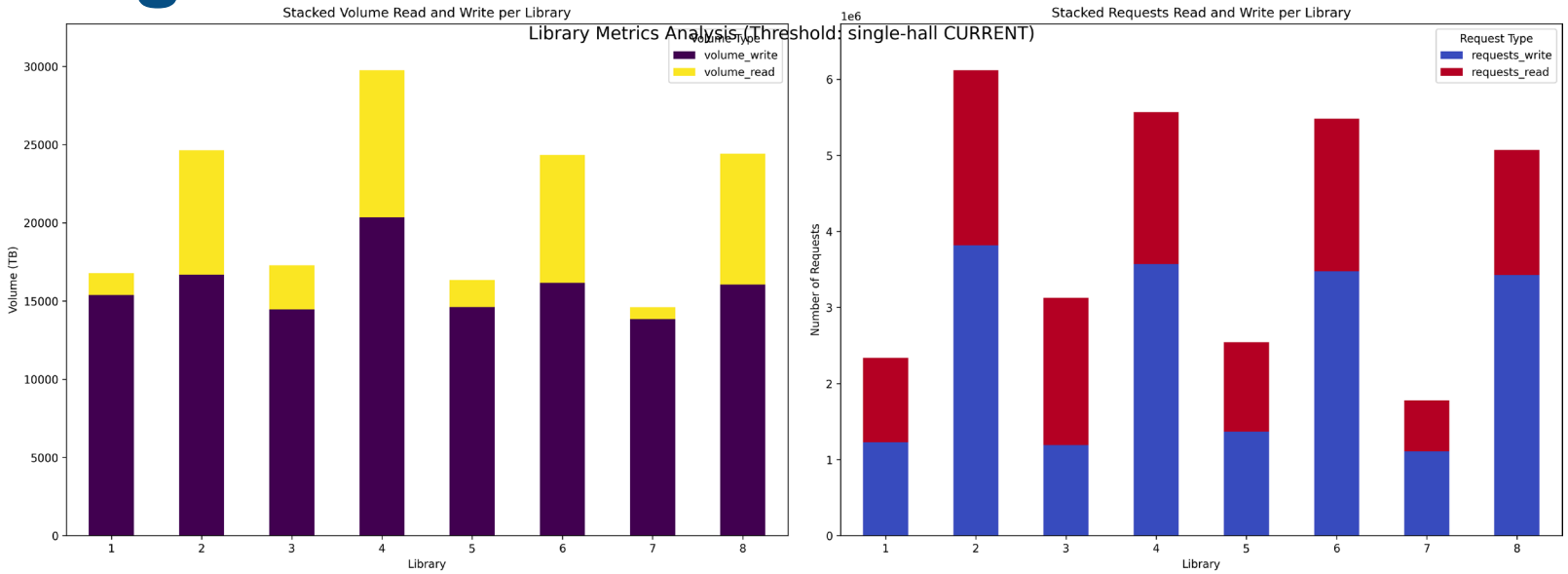


The big picture

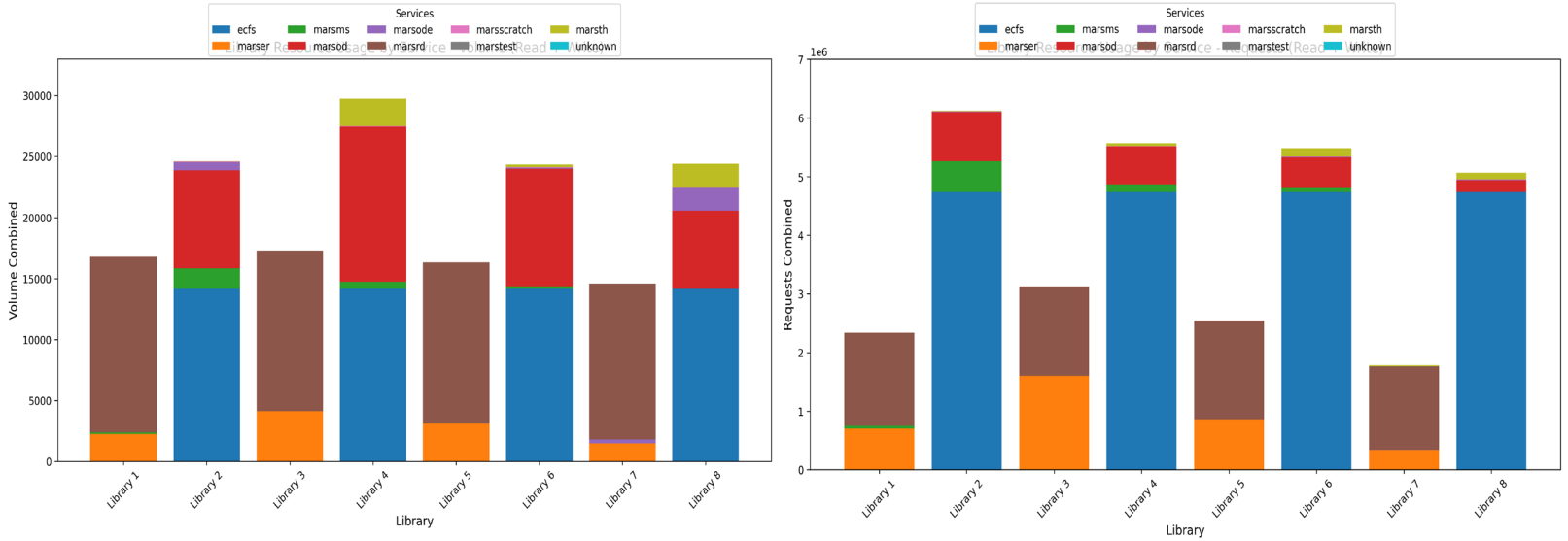


Current balancing across libraries

- Volume read
- Volume write
- Requests read
- Requests write



- ECFS
- MARSOD
- MARSRD
- MARSER
- MARSRD
- MARSTH
- MARSODE
- MARSMS



What is a field

Class (Operations)

Type (Forecast)

Stream (Daily Archive)

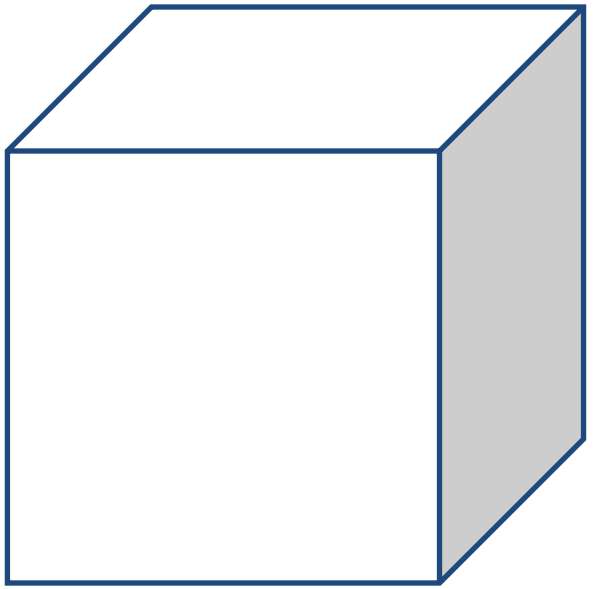
Date (1978-01-12)

Base Time (12 UTC)

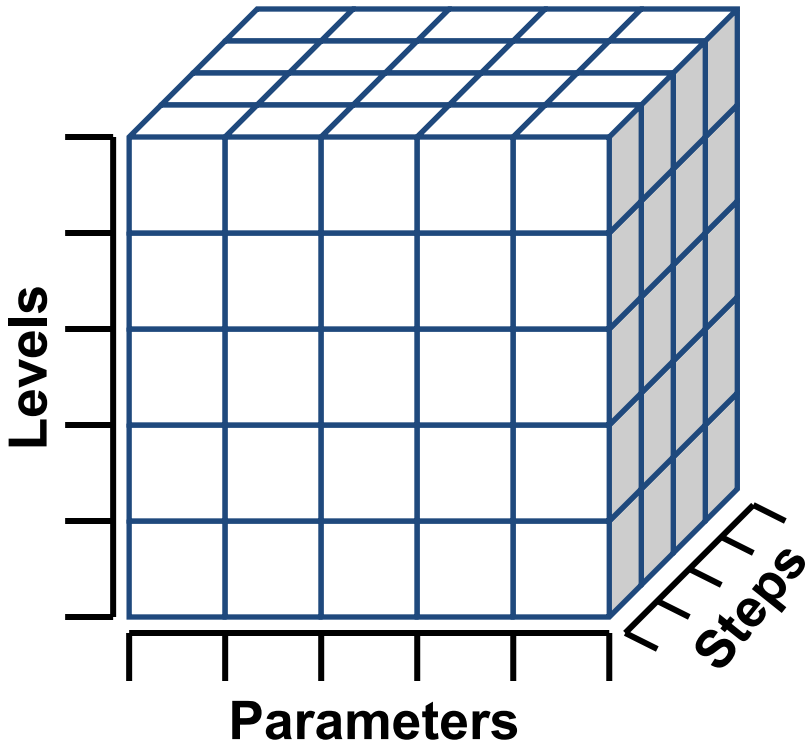
Step (240 hours)

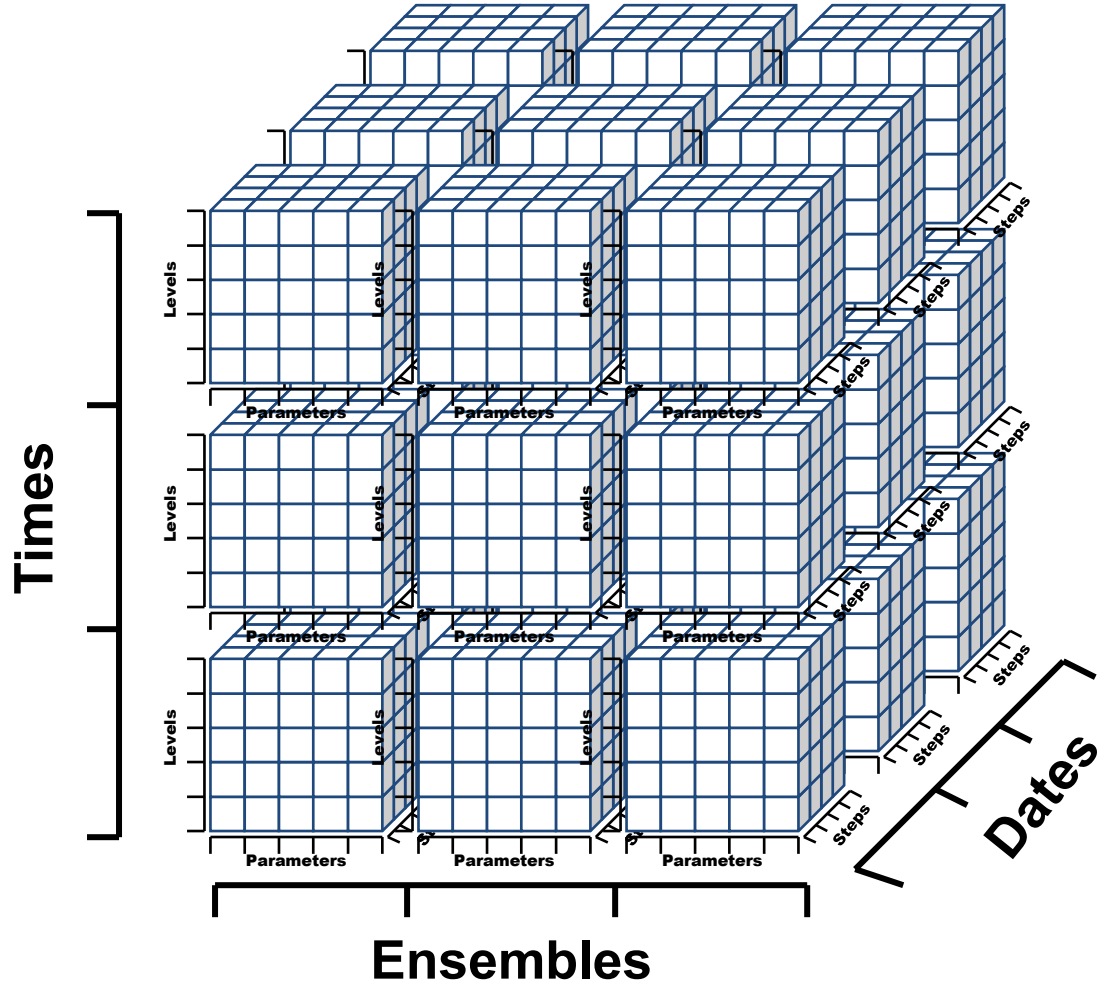
Parameter (Temperature)

Level (1000 hPa)



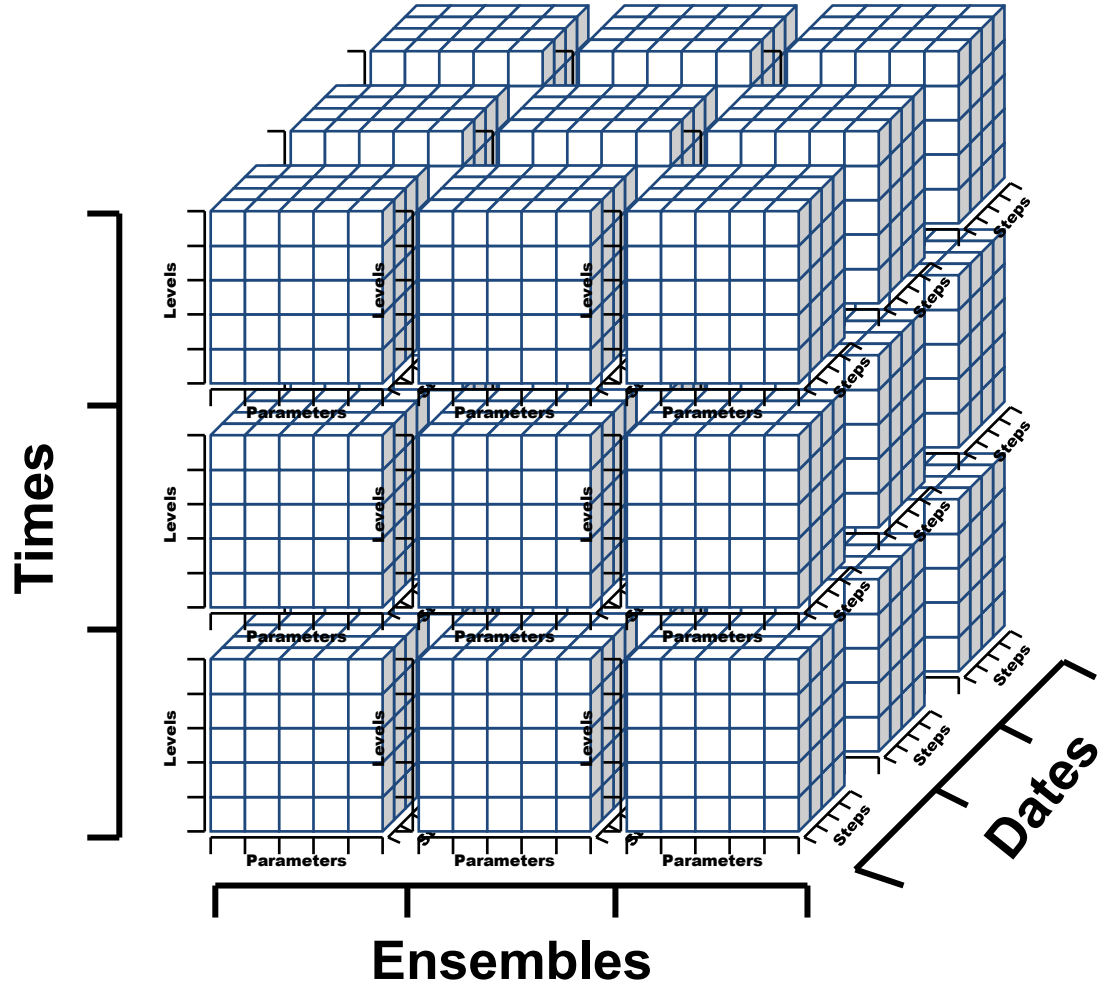
Archive objects are cubes and hyper-cubes

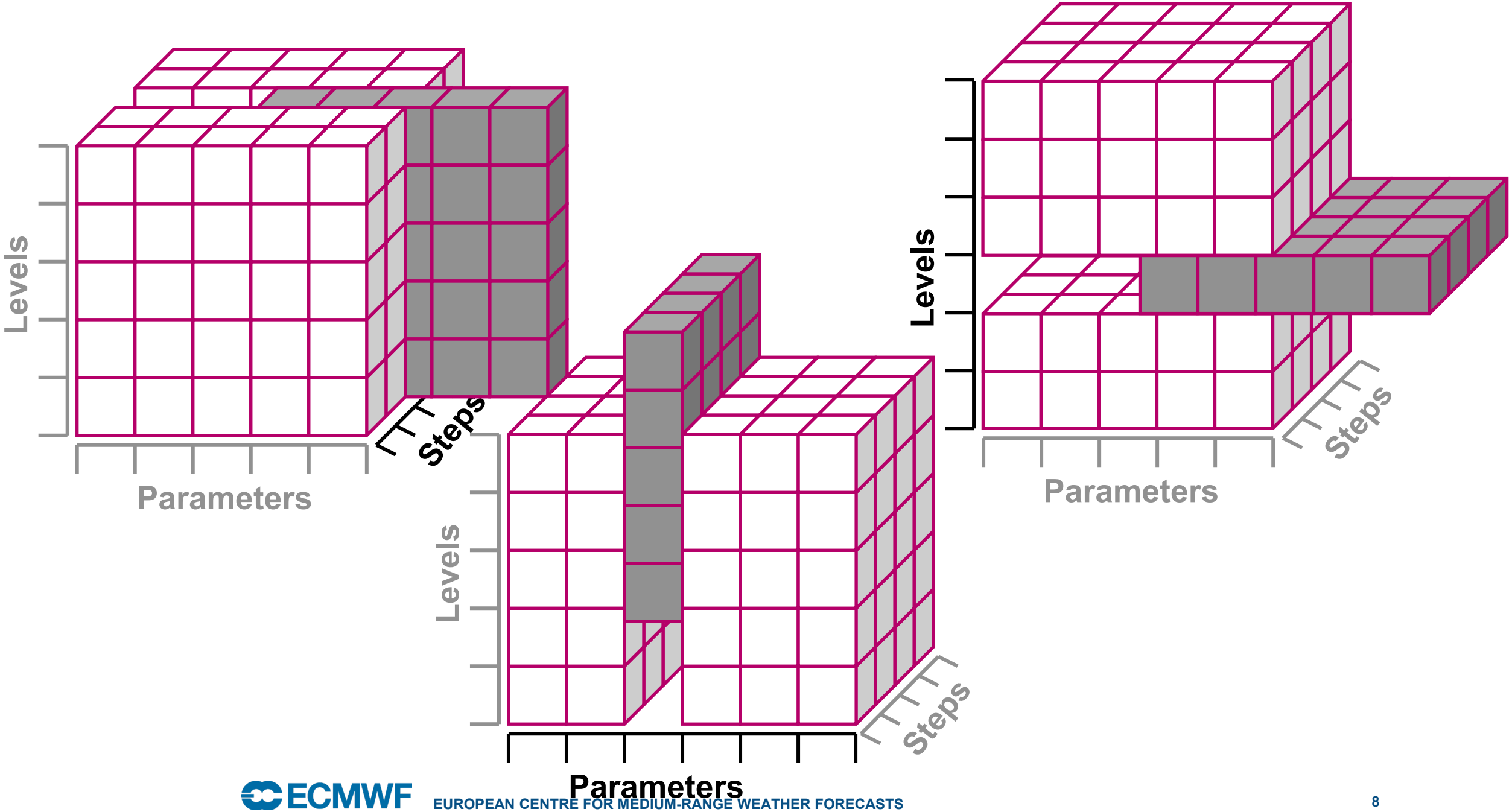


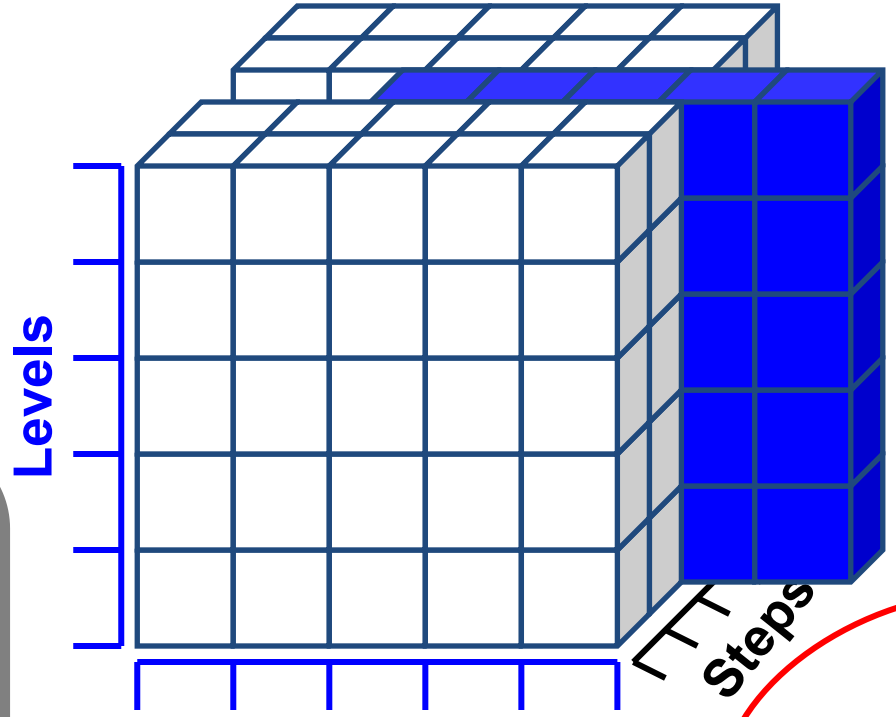


Archive objects

- Logical grouping
  - One forecast
  - One month of analysis
- Grouping
  - Hundreds of thousands of fields
  - Size of a tape
  - Reduce number of files
- Natural Collocation
  - **Tape families**
  - **Minimise tape mounts**




Levels
Parameters
Steps
Levels
Parameters
Steps
Levels
Parameters
Steps



MARS Server Meta-Data *"Shapes"*

Levels

850 | 500 | 200 | 100 | 50

Parameters

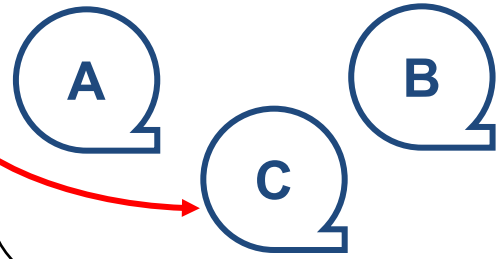
Z | T | U | V | W

Steps

12 | 24 | 30 | 36 | 48

Data Server Meta-Data *"Layouts"*

Offset	Length	Files
0	10000	A
10000	10000	A
0	8000	C
0	15000	B
15000	15000	B
8000	8000	C
…	…	…



Strategies to load balance accesses across libraries

- We assign about 500 tape families (ECFS + MARS) to one or more libraries, within one hall, or across two halls.

- Tape families have their own write and read access pattern we need to account for.

Strategies to load balance accesses across libraries

- To achieve this across an arbitrary number of libraries we aim to spread tape families in such a way that these totals are as equal as possible across libraries.

- Read volume (in TB) from libraries.
- Write volume (in TB) from libraries.
- Number of Read requests reaching each library.
- Number of Write requests reaching each library.

- This is a multidimensional optimisation problem under constraints where we want to minimise the sum of the imbalance (or deviation) of the four metrics across libraries.

How to solve this?

Several techniques can be used to solve this problem (NP-hard):

- If we need an optimal solution → Mixed Integer Linear Programming (MILP)
- If we need a fast approximation → Greedy Heuristic
- If strict constraints apply → Constraint Programming
- If the problem is too large for MILP → Simulated Annealing/Genetic Algorithm
- If the problem is huge → ML can provide a warm-start solution for MILP

Greedy Heuristic (Fast Approximation)

Best for: Quick solutions when optimality is less critical.

Approach:

- Sort families by volume or requests.
- Assign each family to the library with the current lowest load.
- Repeat until all families are assigned.

Pros: Simple and efficient.

Cons: May not find the most balanced solution.

Mixed Integer Linear Programming

Best for: Finding an optimal solution when computation time is not a constraint.

Approach to formalise the problem:

- Define decision variables for family-to-library assignments.
- Set up constraints on decision variables to ensure:
  - Even distribution of read/write volumes and requests.
  - Operational or resilience constraint you must meet.
- The objective function must be a linear combination of decision variables, minimising imbalance across libraries.
- Solve using an LP solver like HiGHS, Gurobi (commercial), or CPLEX (commercial).

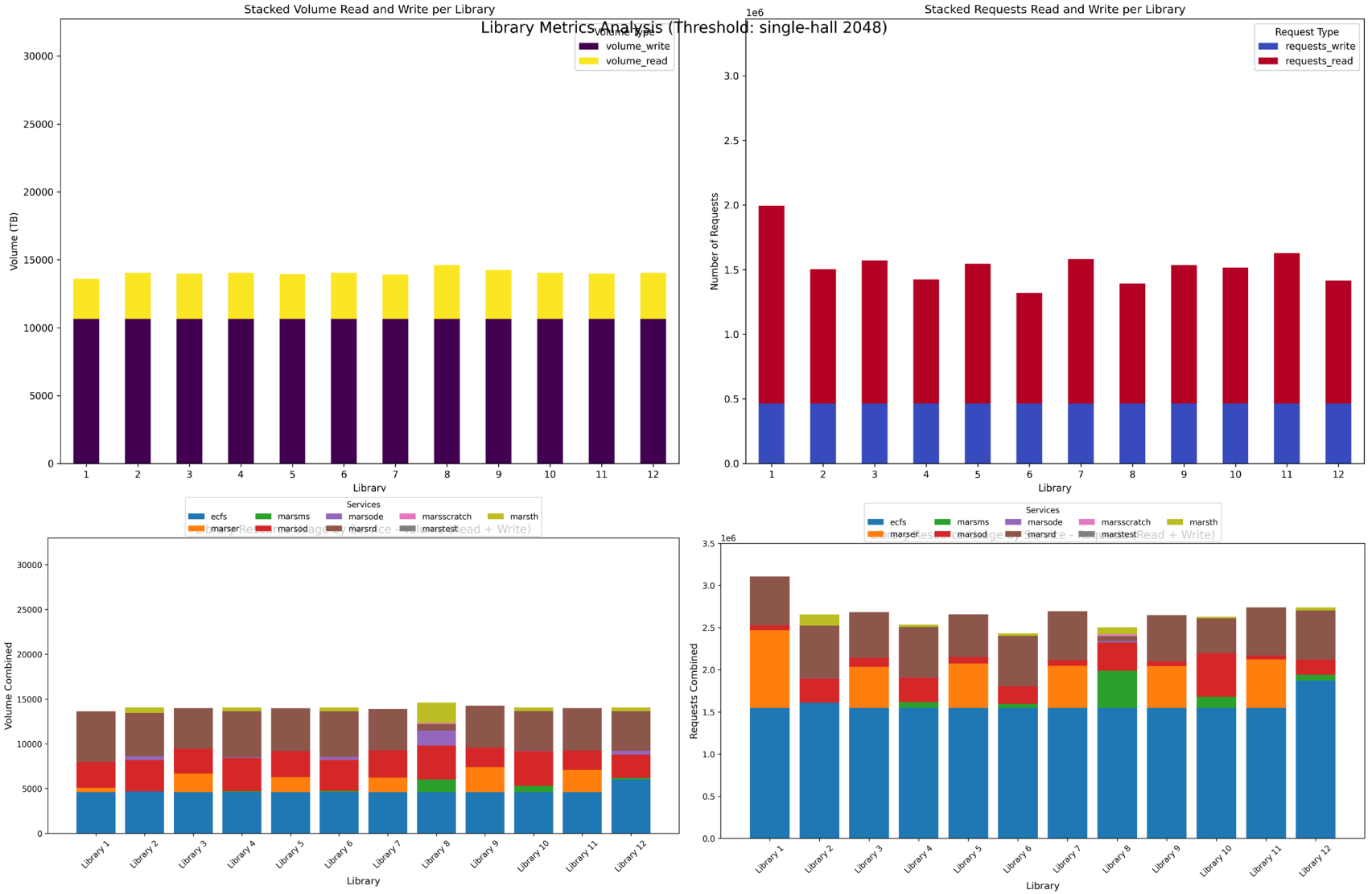
Pros: Guarantees an optimal or near-optimal solution.

Cons: Can be slow for very large datasets.

Selected tape distribution

- Volume read
- Volume write
- Requests read
- Requests write

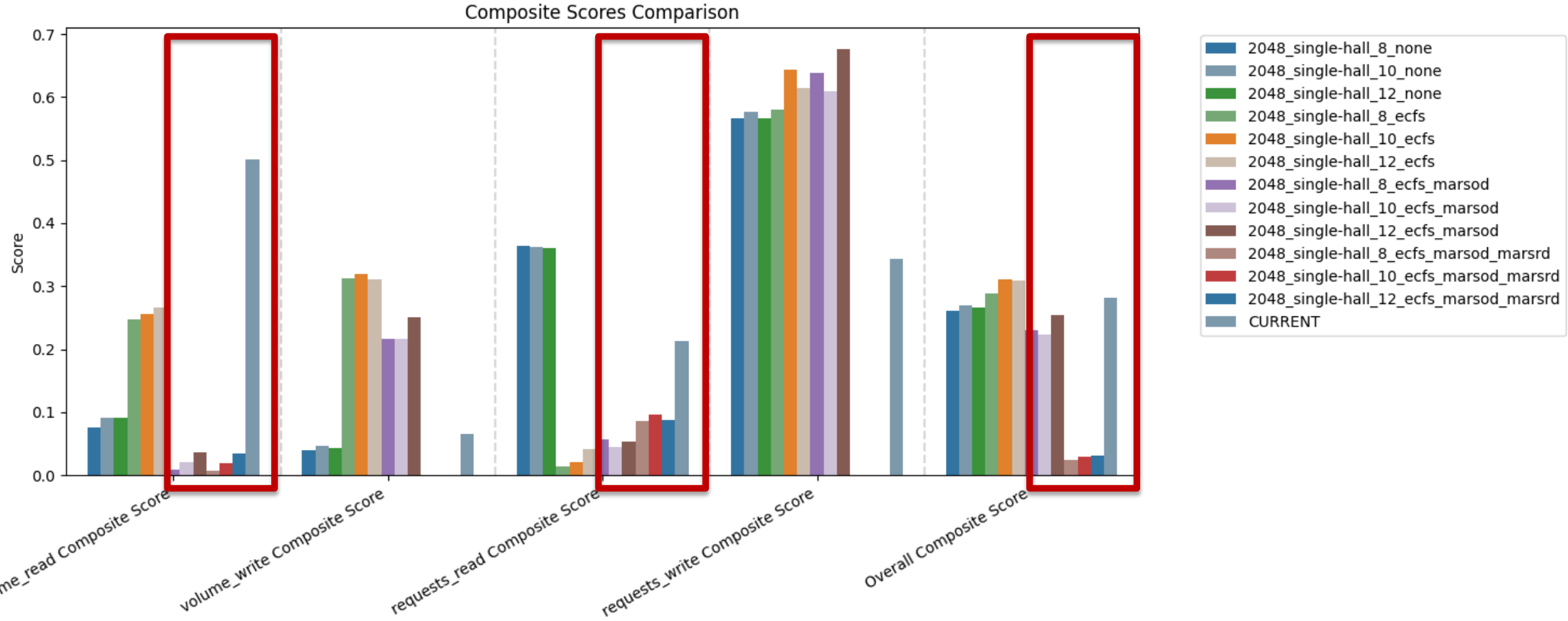
- ECFS
- MARSOD
- MARSRD
- MARSER
- MARSRD
- MARSTH
- MARSODE
- MARSMS



Validation

A mixture of normalised standard deviation and Jain's Fairness Index to assess the balancing.

The lower the better



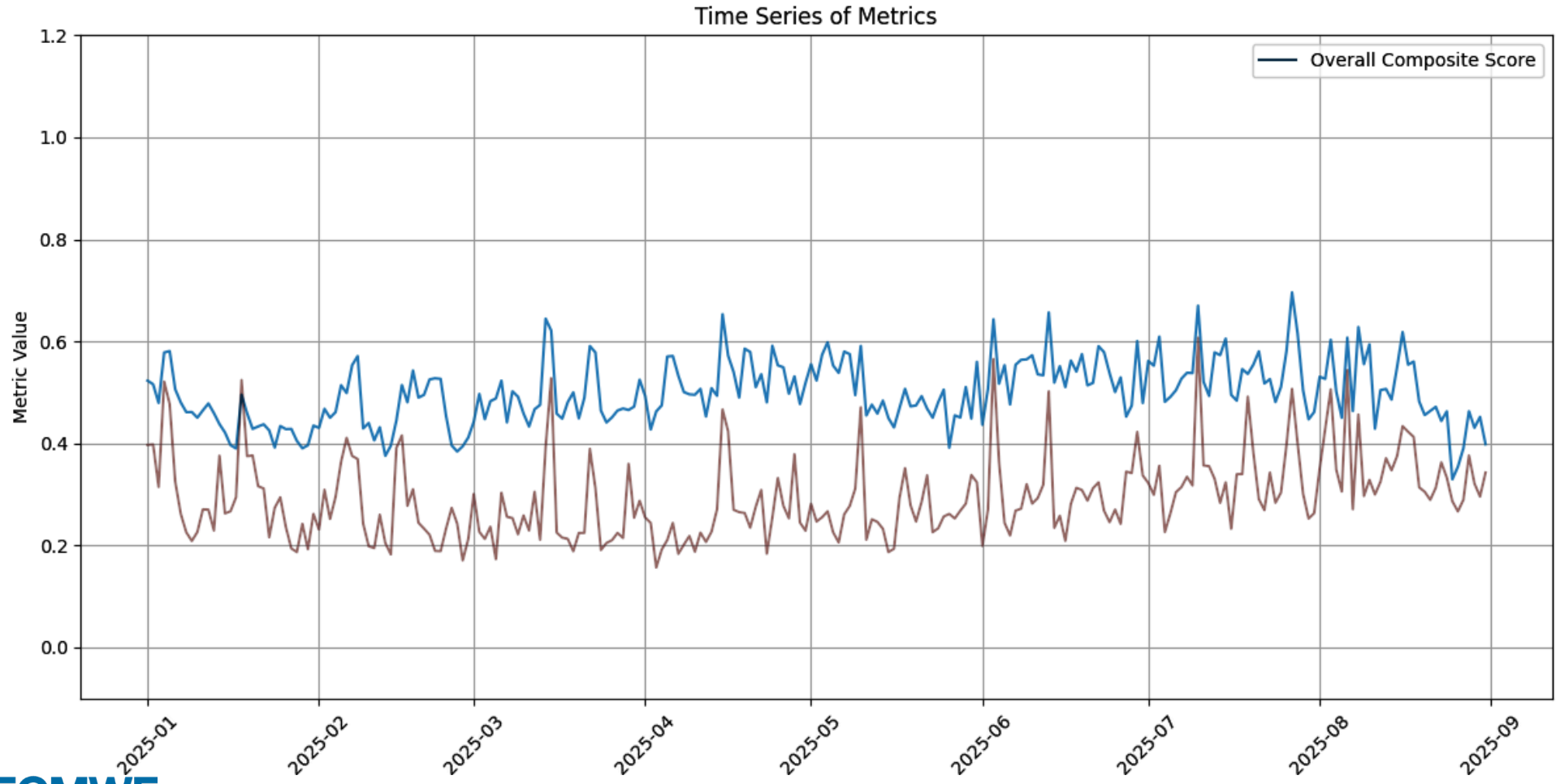
Validation

Is distribution made from 2024 activity leading to a good daily balance over 2025?

A mixture of normalised standard deviation and Jain's Fairness Index to assess the balancing.

Current

New balance



Next steps

Redistribute tape families across our 12 tape libraries

Plan for future tape families that will likely disrupt the balancing (ERA6 for example)

Better link load balancing score variations to user behaviour and changes in disk caching strategies or hardware incidents

Modify the objective function of the MILP to also minimise the number of tape movements involved in future rebalancing