

Fine-grained data caching approaches to speedup a distributed RDataFrame analysis

ROOT

Data Analysis Framework

<https://root.cern>



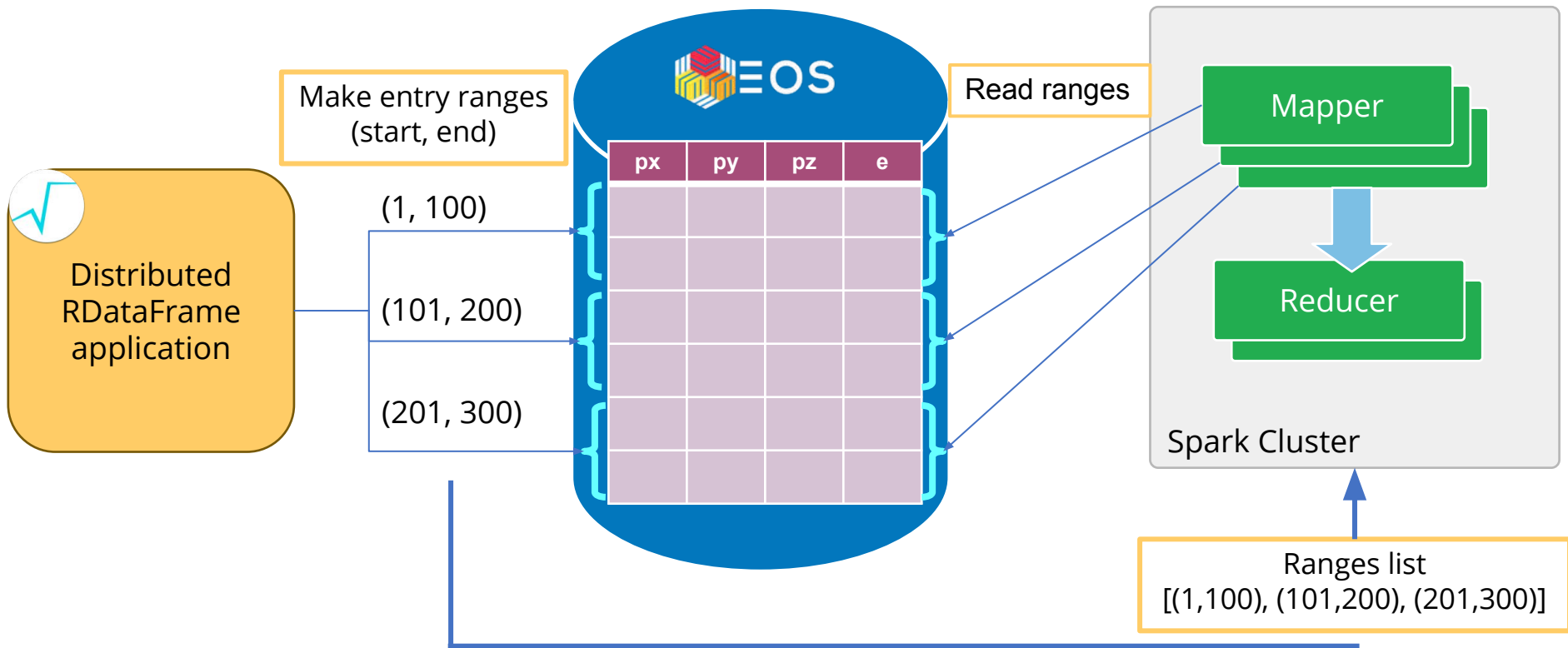
Target

- ▶ Interactive data **exploration**, repeat same operations with slightly different **parameters**.
- ▶ Data **filtering**, creating **histograms**...
- ▶ Still quite a lot of data to process (**GB** → **TB**)

- Enable **interactive large scale** data analysis
- From start to end in a single interface



Distributing RDataFrame workloads

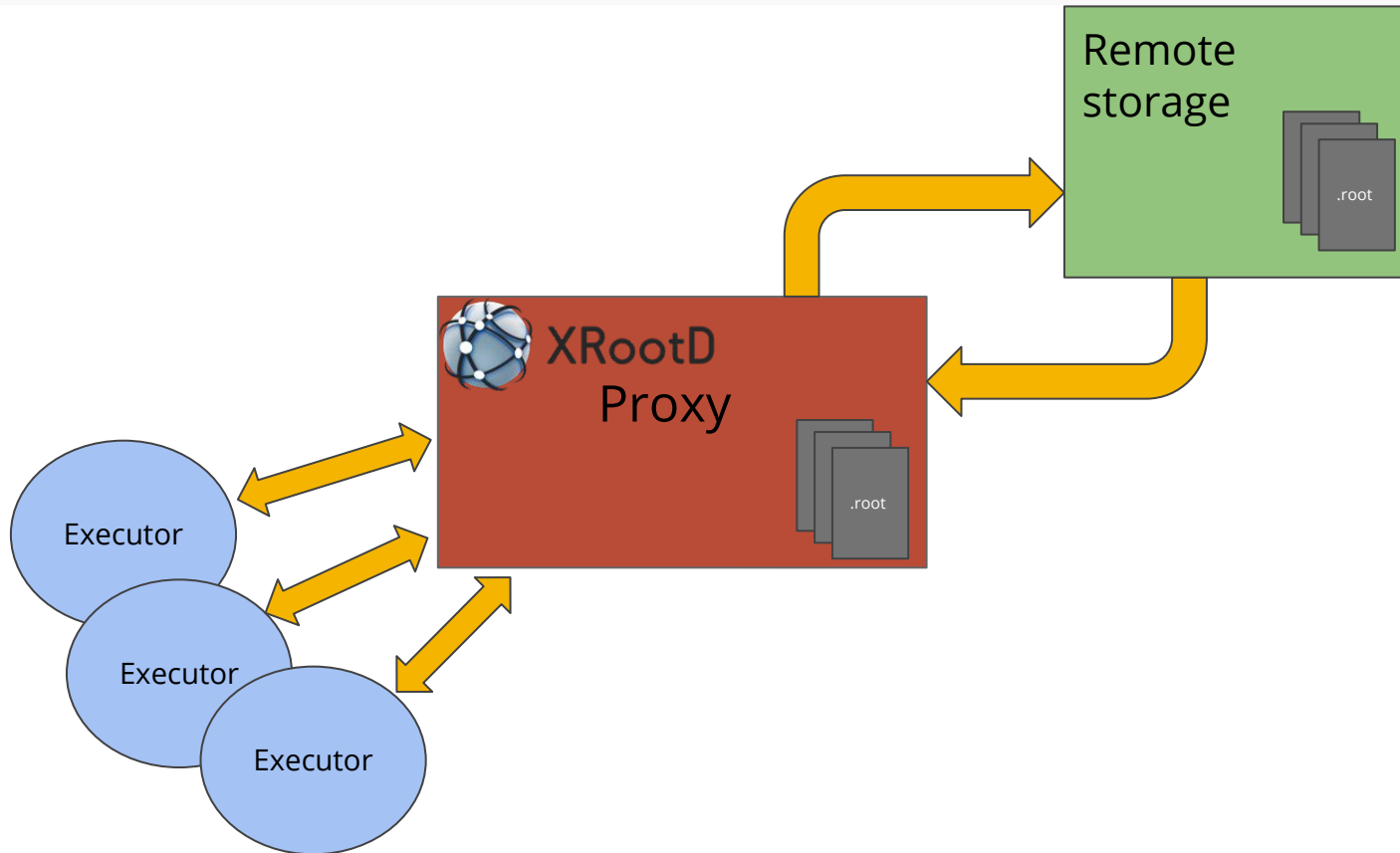




How/why data caching

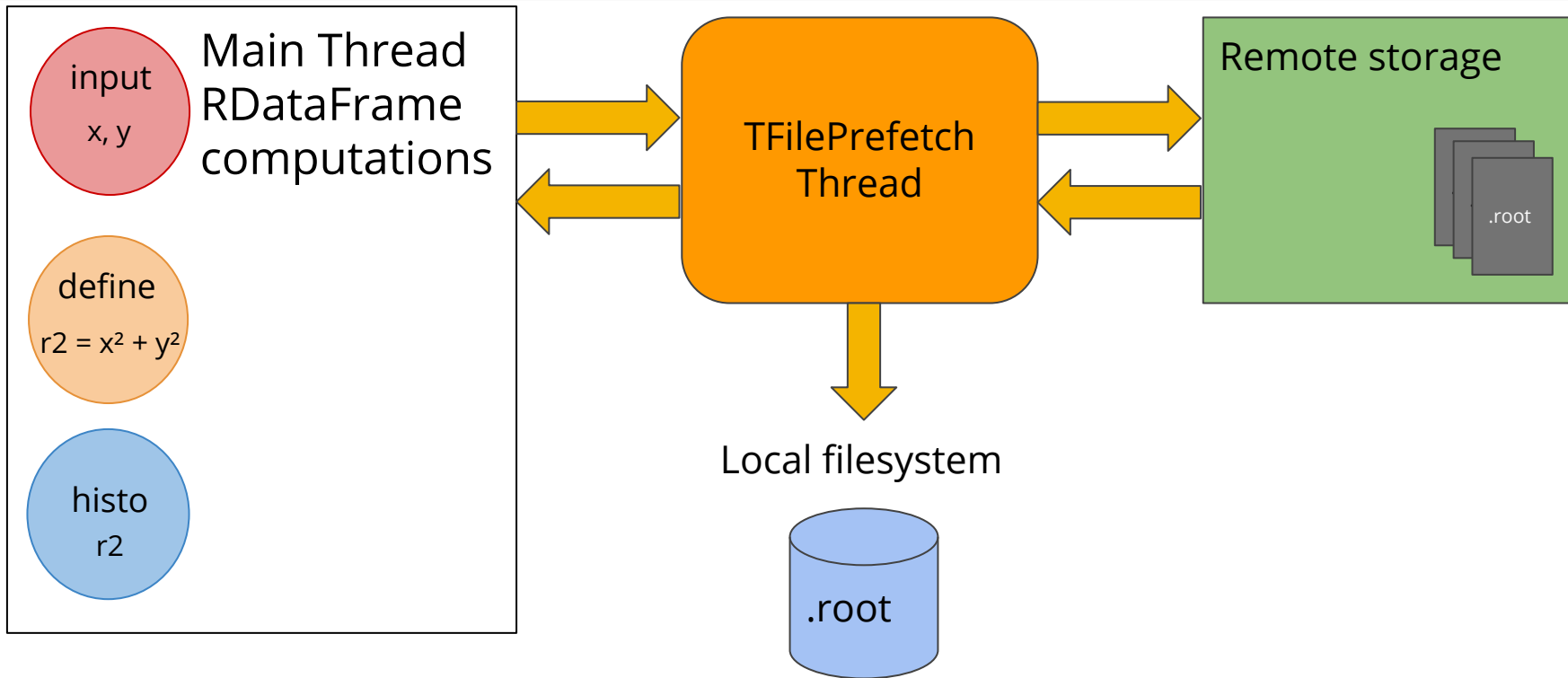
- ▶ Data is often remote, lots of time spent in IO → **caching**
- ▶ Sparse data access → **granularity** required
- ▶ Two approaches compared: **separate** cache **server** vs **local caches** on the nodes

XRootD proxy cache server





TFilePrefetch local cache





Test application

Reference dataset

- **1 file**, 100M entries, 5 columns
- File size: 1.8 GB
- Cached column size: **~700MB**
- Stored in **EOS**

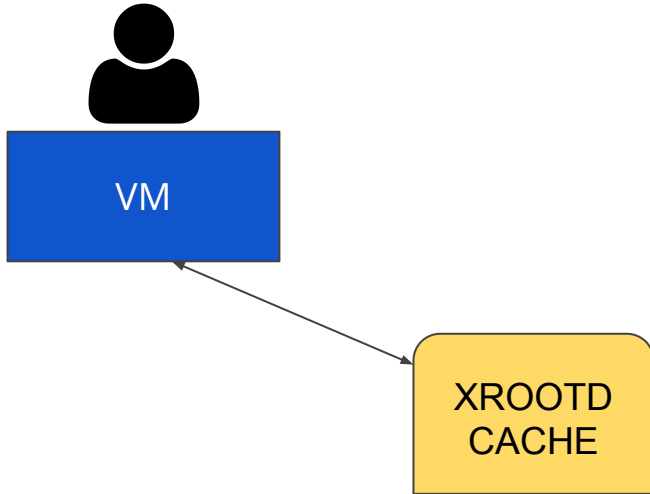
Test runs

- Simple operation on a column
- Single node / distributed
- Remote data / cached data



[Github repository](#)

Setup: Single Node

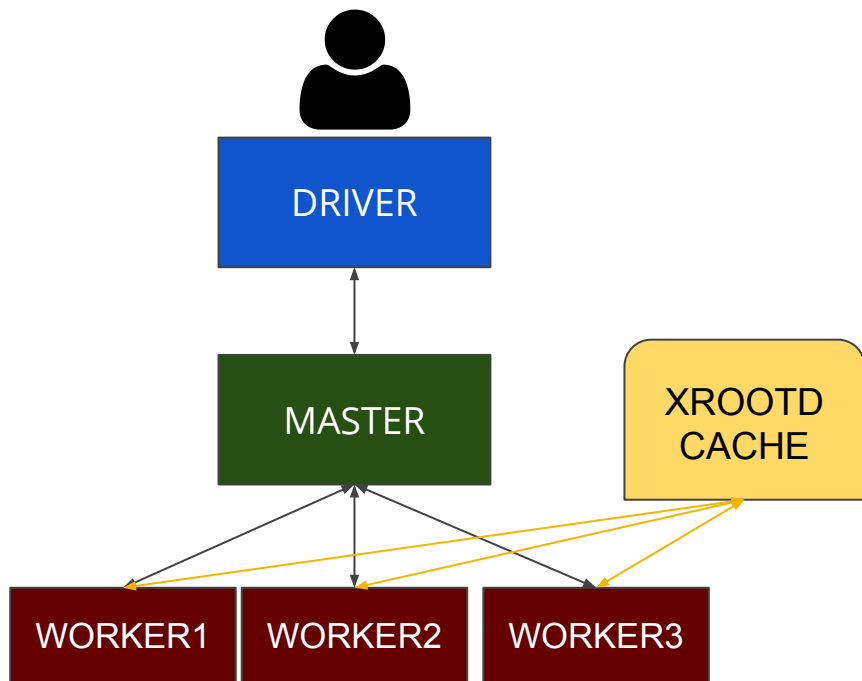


Test setup is composed of:

- 1 VM (1 core, 1024MB RAM, 10 GB spinning disk)
- 1 physical node (8 cores, 16 GB RAM, 256GB SSD)



Setup: Distributed

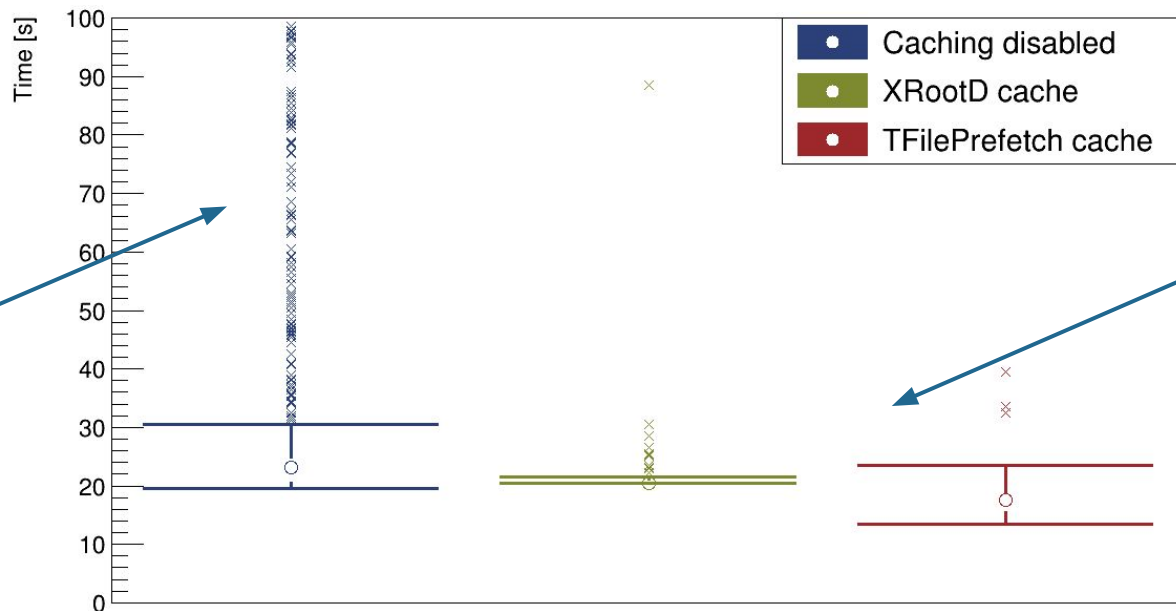


Test setup is composed of:

- 5 VMs (1 core, 1024MB RAM, 10 GB spinning disk)
 - driver: launches user applications
 - master: schedules work
 - worker[1-3]: execute jobs
- 1 physical node (8 cores, 16 GB RAM, 256GB SSD)



Single Node application



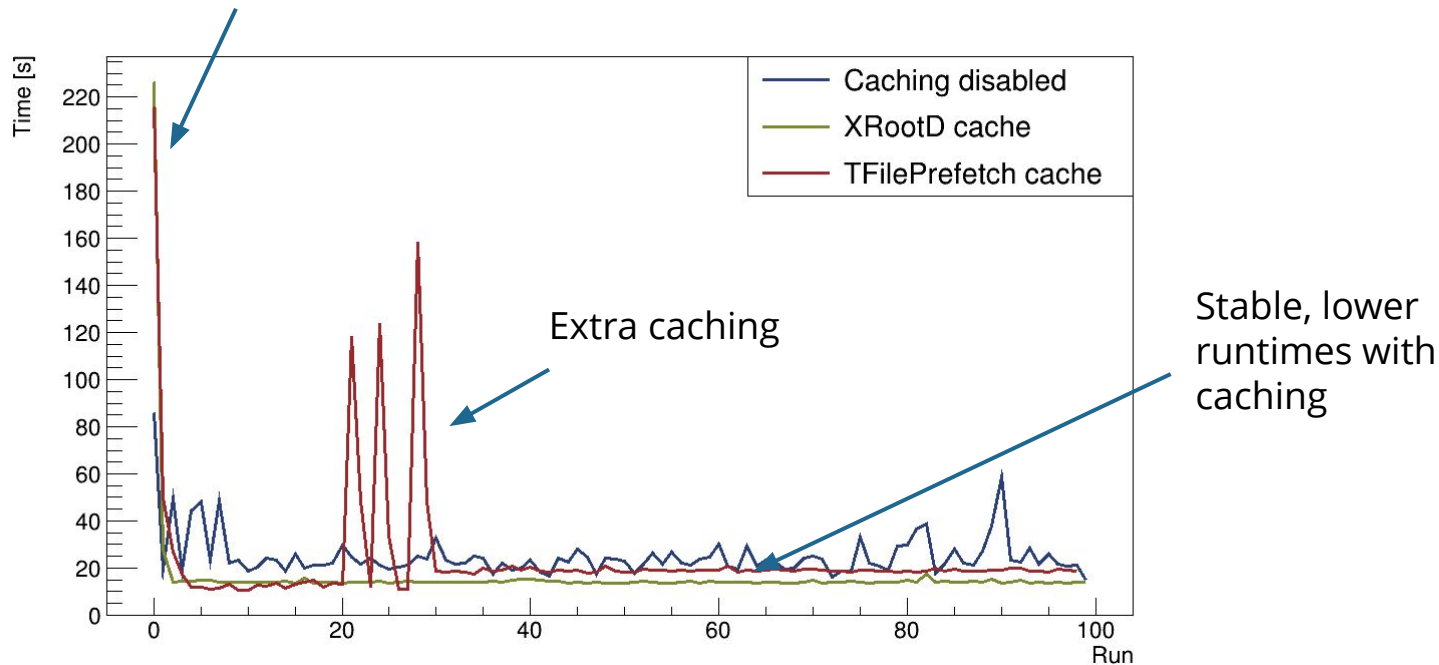
Very high runtime variability due to remote IO

XRootD cache shows the best consistency



Distributed application: without data locality

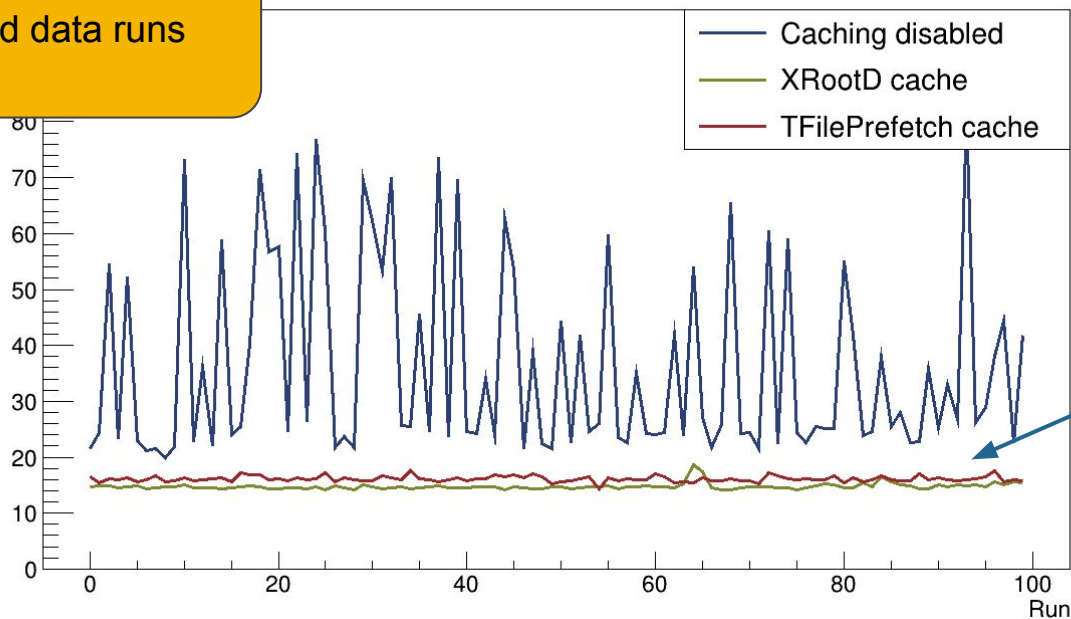
First run, caches are populated





Distributed application: data locality

Here only showing cached data runs



Both mechanisms lead to similar runtimes



Conclusions and next steps

- ▶ HEP analysis can benefit from caching
- ▶ Need to be careful: granularity, cache system layout, data locality
- ▶ Performance of XRootD proxy and TFilePrefetch is comparable on the data and workflow analysed (with smart reutilisation of data ranges)

For the future:

- Bigger datasets and more computations
- different compression algorithm, filesystem cache, TTree vs RNTuple
- Investigate issues with more latency (geographical distance)
- Ad-hoc caching mechanisms for RDataFrame analysis