DRAFT

# Prototyping an Analysis Facility at CERN

Diogo Castro, Enrique Garcia, Giovanni Guerrieri, Ben Jones, Gavin McCance, A. Sciabà, M. Schulz, Enric Tejedor Saavedra

HEPiX Spring Workshop 2024, Paris 15-19 April 2024

# Introduction

- **Data Analysis at the LHC is evolving**
  - Very compact data formats (NanoAOD, PHYSLite) allow to replicate years of data on a single facility
  - New analysis frameworks (Coffea, RDataFrame) allow to use columnar data analysis concepts on both local and distributed resources
  - Services like Xcache or ServiceX can significantly reduce I/O latency and save processing time
  - Interactive analysis using notebooks adds extra convenience
- **Two different types of resources**
  - High performance nodes (many cores, lots of SSD storage) are used for interactive work
  - Analysis facilities as dedicated clusters with software and services to enable interactive distributed analysis
- **CERN providing the former to experiments, and just released a pilot service for the latter**

# 2022: first studies on the suitability of the CERN infrastructure for interactive analysis

- **Main observations**
  - Amount of interactive analysis still very small
    - O(1000) cores for batch analysis, O(100) for interactive
  - Average read rates rather low (O(10 MB/s))
    - Much below EOS saturation levels
  - No need for caching layers in front of EOS
    - Even HDD-based storage still good enough
- **Conclusions**
  - CERN Batch + EOS perfectly adequate for analysis

https://zenodo.org/records/6337728

| Experiment | Completed jobs | Running jobs | CPU time | Wallclock time |
|---|---|---|---|---|
| ATLAS | 100K | 2500 | 70 years | 80 years |
| CMS | 600K | 8000 | 140 years | 170 years |

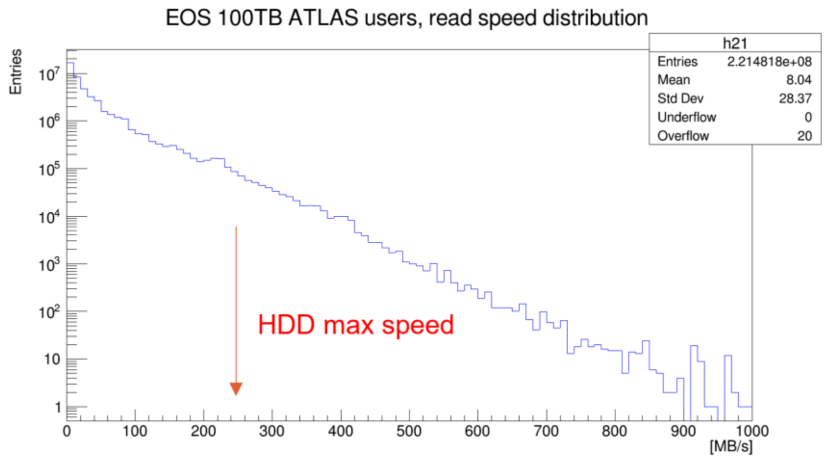
CERN Grid analysis in a 7-day period in October

| | LXBATCH | LXPLUS |
|---|---|---|
| ATLAS | 4400 cores | 100 cores |
| CMS | 8800 cores | 70 cores |

Average number of busy cores in a 7-day period

| | SWAN | Spark/YARN |
|---|---|---|
| ATLAS | 8.5K hours · cores | 18.5K hours · cores |
| CMS | 16K hours · cores | 22K hours · cores |

Wallclock time in a 10-day period

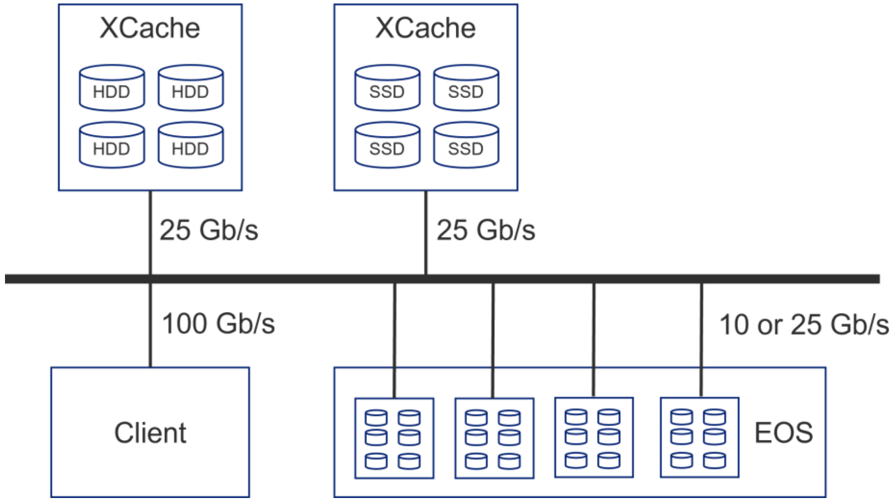


# 2023: Extensive studies of I/O performance for interactive analysis at CERN

- **Goal was to collect analysis workloads and tools to measure I/O performance in different storage configurations and levels of parallelism**
  - Seen as preliminary to understand how an Analysis Facility might work at CERN
- **Several workloads were used for measurements**
  - ROOT's *rootreadspeed* I/O benchmark
  - ROOT's RDataFrame benchmark
  - IRIS-HEP Analysis Grand Challenge, both Coffea and RDataFrame implementations
  - A real CMS analysis using Coffea
  - A real CMS analysis using RDF

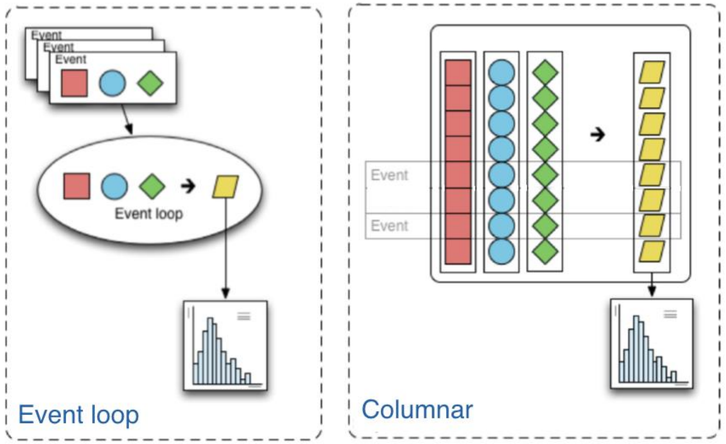
# Testbed setup and metric measurement

- **High performance client node**
  - Two AMD EPYC 7702 (128 cores)
  - 1 TB of RAM
  - 20 SSD of 4 TB each (of which 10 in RAID0)
  - 100 Gb/s connection
- **Two Xcache nodes**
  - Two Intel Xeon Silver 4216 (32 cores)
  - 192 GB of RAM
  - One with ~ 1 PB in HDD, the other with 32 TB in SSD
- **Storage system**
  - EOS at CERN (EOSCMS and CERNBOX)




- **HSF PrMon tool to measure performance**
  - Wallclock time
  - CPU time
  - Read bytes (from storage or network)
  - Time spent in data processing
  - CPU (pseudo) efficiency
    - CPU time / (wallclock time × workers)
  - Average read data rate
    - read bytes / processing time

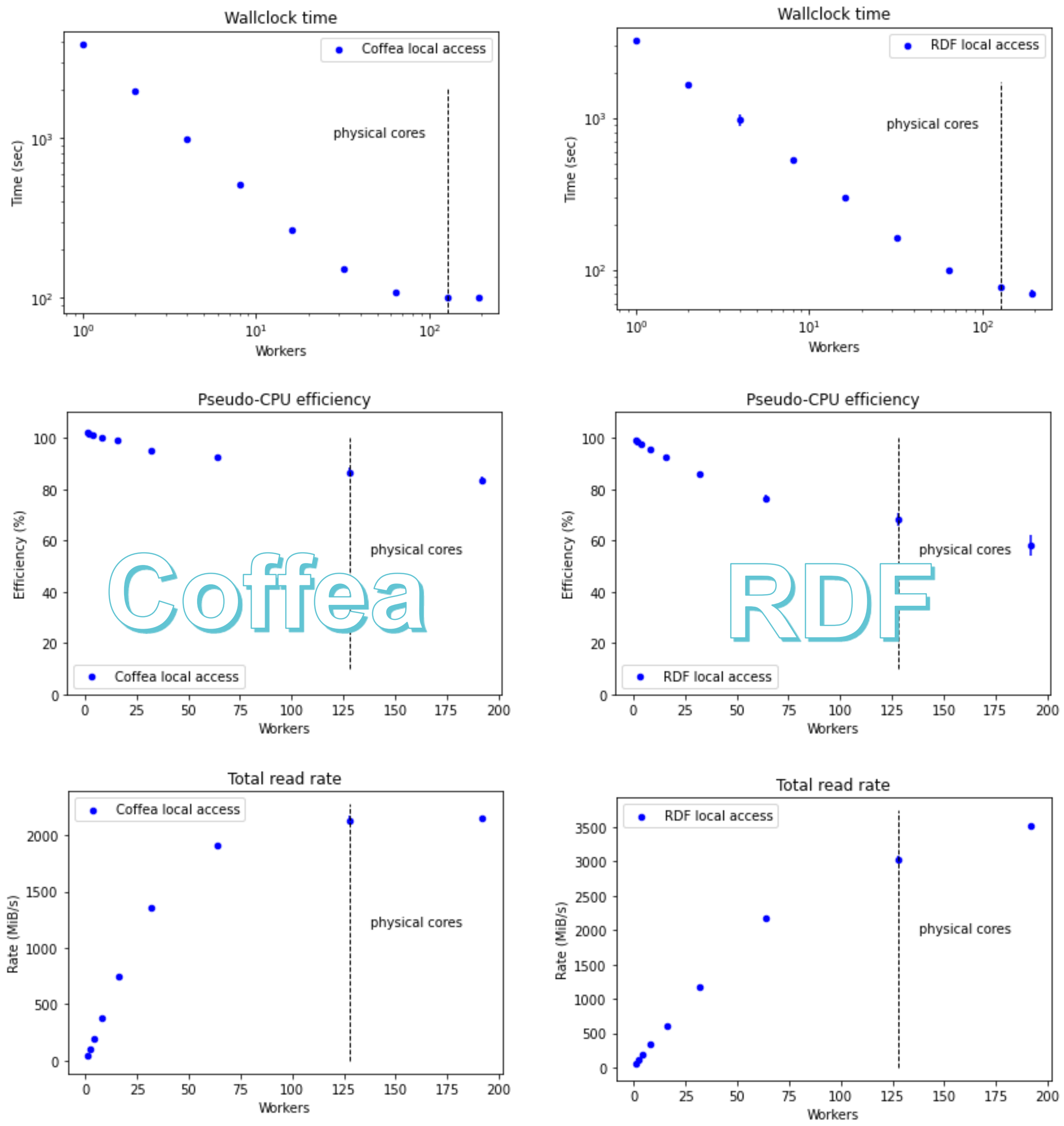
# Analysis Grand Challenge ttbar analysis




- **Simplified analysis from CMS used as technical demonstrator in IRIS-HEP**
  - Input dataset 3.6 TB, 2300 ROOT files, 1.5 GB/file consisting of CMS 2015 Open Data
- **Columnar analysis paradigm**
  - Distributed using a map-reduce concept
- **Original Coffea implementation**
  - ROOT-less, parallelism via Python futures or Dask (multiprocess)
- **RDataFrame port**
  - ROOT-based, parallelism via implicit multithreading, or multiprocess via Dask
- **Measure performance and scalability**
  - Local parallelism on client node
  - Data read from
    - local node
    - directly from EOS via xrootd
    - via an XCache instance

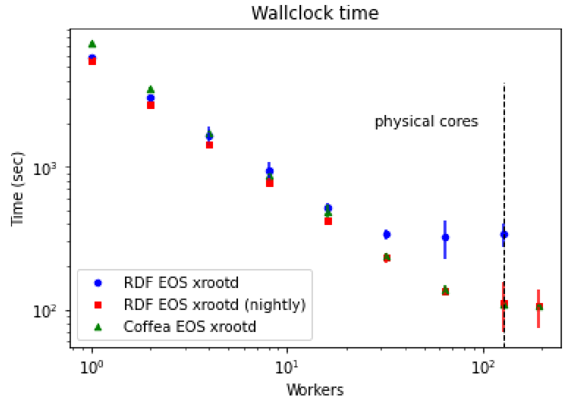
# Local access performance

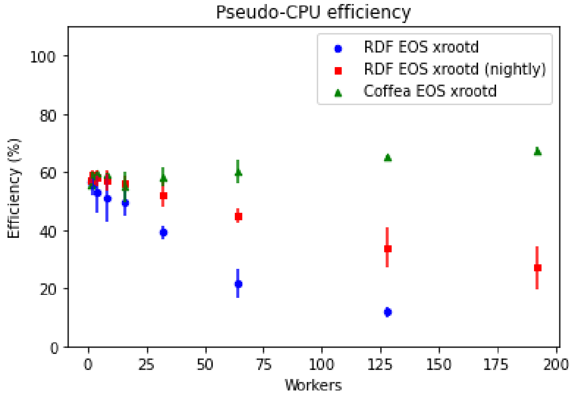
- **Scalability is excellent**
  - Some bottleneck appears for high numbers of workers
- **The CPU efficiency comparably high**
  - I/O not a strong bottleneck
- **Local, fast SSD storage is always going to work well**
  - Aggregate read rates up to 3 GB/s

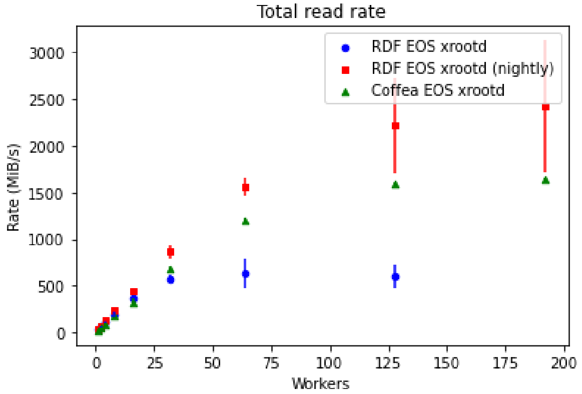


# Direct access to EOS

- **Scalability still good when parallelism is via multiprocess**
  - RDF multithreading did not perform well with xrootd and many threads
  - The cause was a significant lock contention, later fixed
- **CPU efficiency practically constant around 60% with multiprocess parallelism**
  - I/O time is not negligible anymore but no bottlenecks
- **Two EOS instances tested**
  - EOSCMS and EOSUSER, similar results



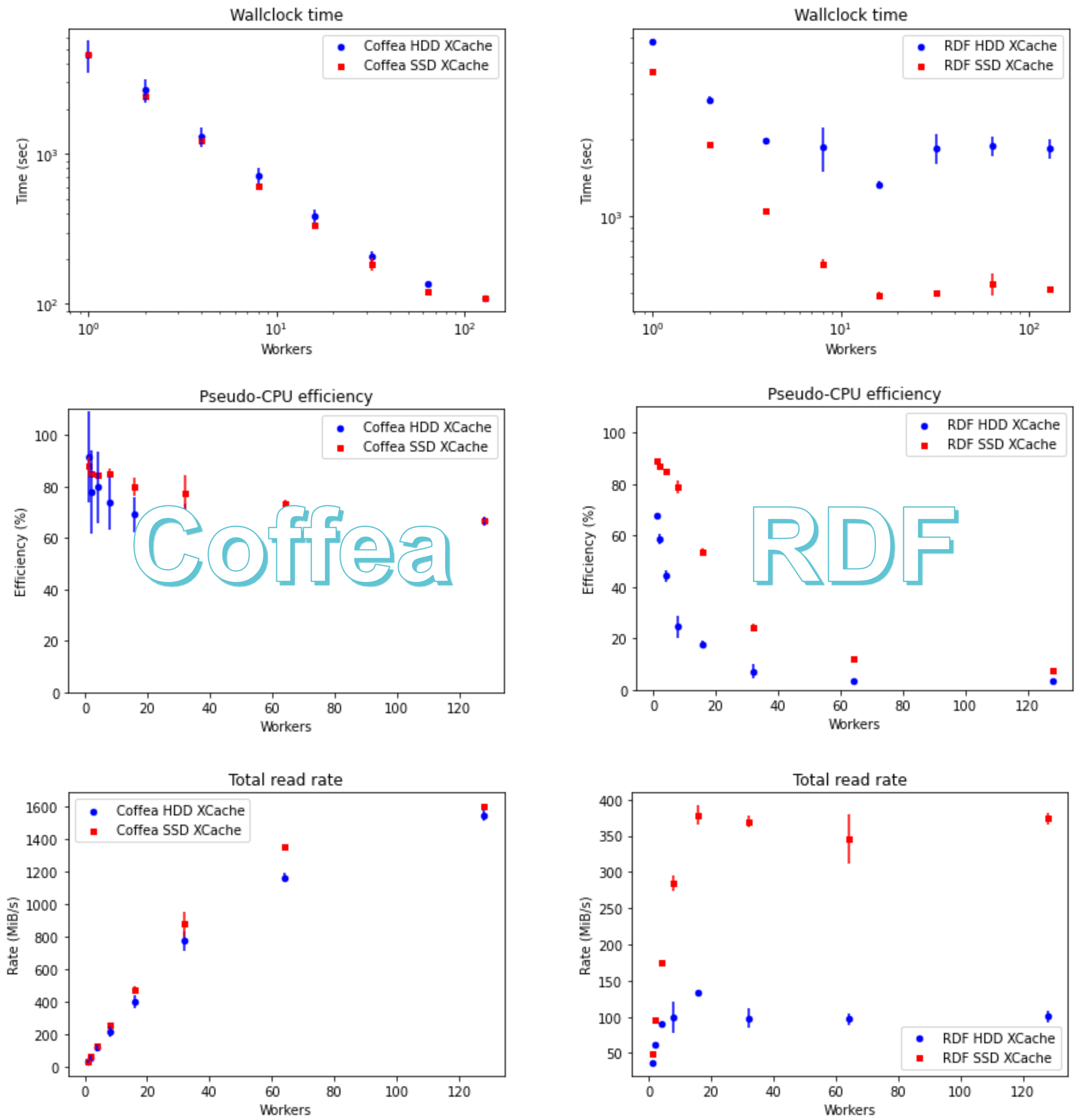
# HDD/SDD-based XCache

- **Compared performance of direct access to Nebraska and CERN, cold cache and warm cache**

| Coffea + HDD XCache: wallclock time (s) | | | |
|---|---|---|---|
| Site | Direct | Cold | Warm |
| Nebraska | 440 ± 20 | 600 | 130 ± 5 |
| EOSCMS | 140 ± 10 | 320 | 137 ± 3 |

| RDF MT + HDD XCache: wallclock time (s) | | | |
|---|---|---|---|
| Site | Direct | Cold | Warm |
| Nebraska | 5500 ± 1000 | 20000 | 1530 ± 80 |
| EOSCMS | 320 ± 100 | 8000 | 1600 ± 400 |

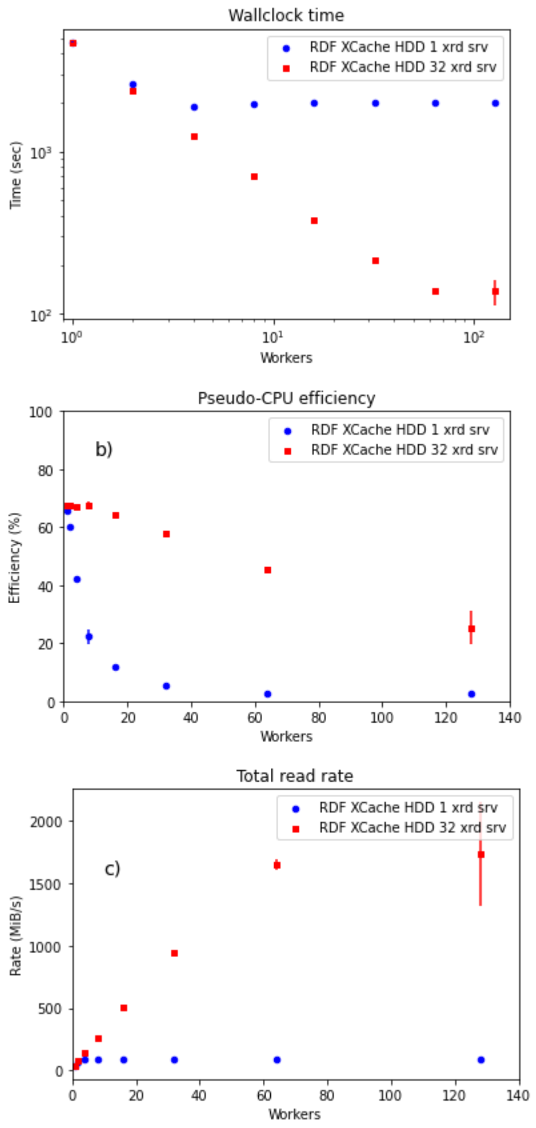
- **Performance results**
  - A cold cache is slower than direct access!
    - Due to sparse file access and network latency
  - Multiprocess scales very well
    - HDD XCache almost as good as SSD XCache
  - RDF multithreaded scales very poorly "out of the box"
    - All connections multiplexed into one ⇒ bottleneck!
    - SDD XCache helps a lot, but scalability is still broken



# Fixing the multithreaded RDataFrame performance on XCache

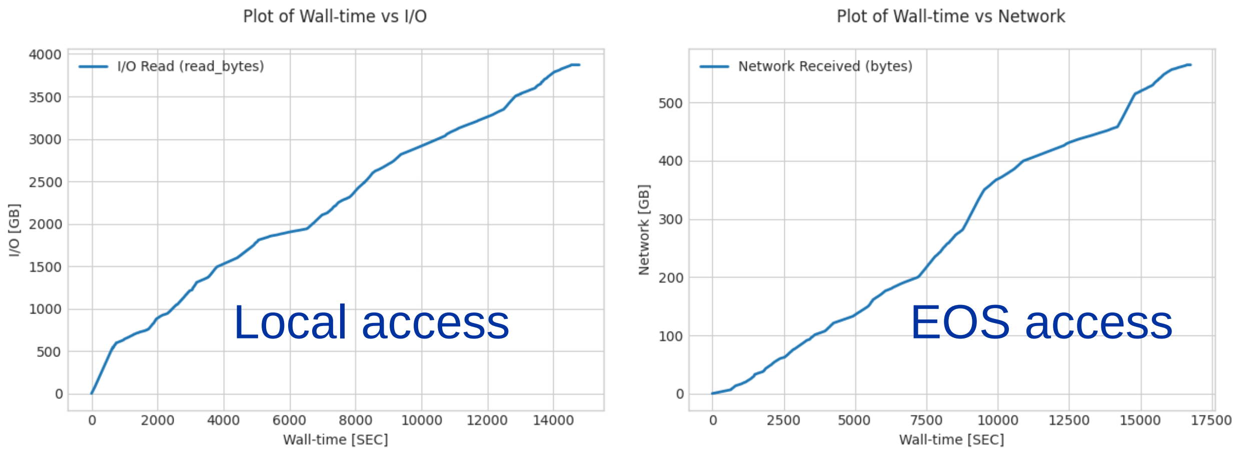
- **Scalability with ROOT multithreading and XCache can be improved**
  - XRD_PARALLELEVTLOOP=10 on the client largely improves Xrootd performance

- **In a default configuration, XCache is heavily bottlenecked by the only server process**
  - Changed to have 32 processes (each serving three disk servers)
  - Scalability becomes excellent!



# CMS NanoAOD analysis using Coffea

- **Real world Higgs analysis**
  - First observation of ggH → cc
  - Running on NanoAOD
    - Tests using 2017 data, 6 TB, average file size 160 MB
- **Performance comparison in different scenarios**
  - Local access
  - Direct EOS access
  - HDD XCache




| | Wallclock time (h) | CPU efficiency (%) | Read rate (MiB/s) |
|---|---|---|---|
| Local access | 4.1 | 103 | 270 |
| Direct EOS access | 4.7 | 87 | 34.5 |
| Cold HDD XCache | 4.6 | 89 | 34.7 |
| Warm HDD XCache | 4.8 | 82 | 34.1 |

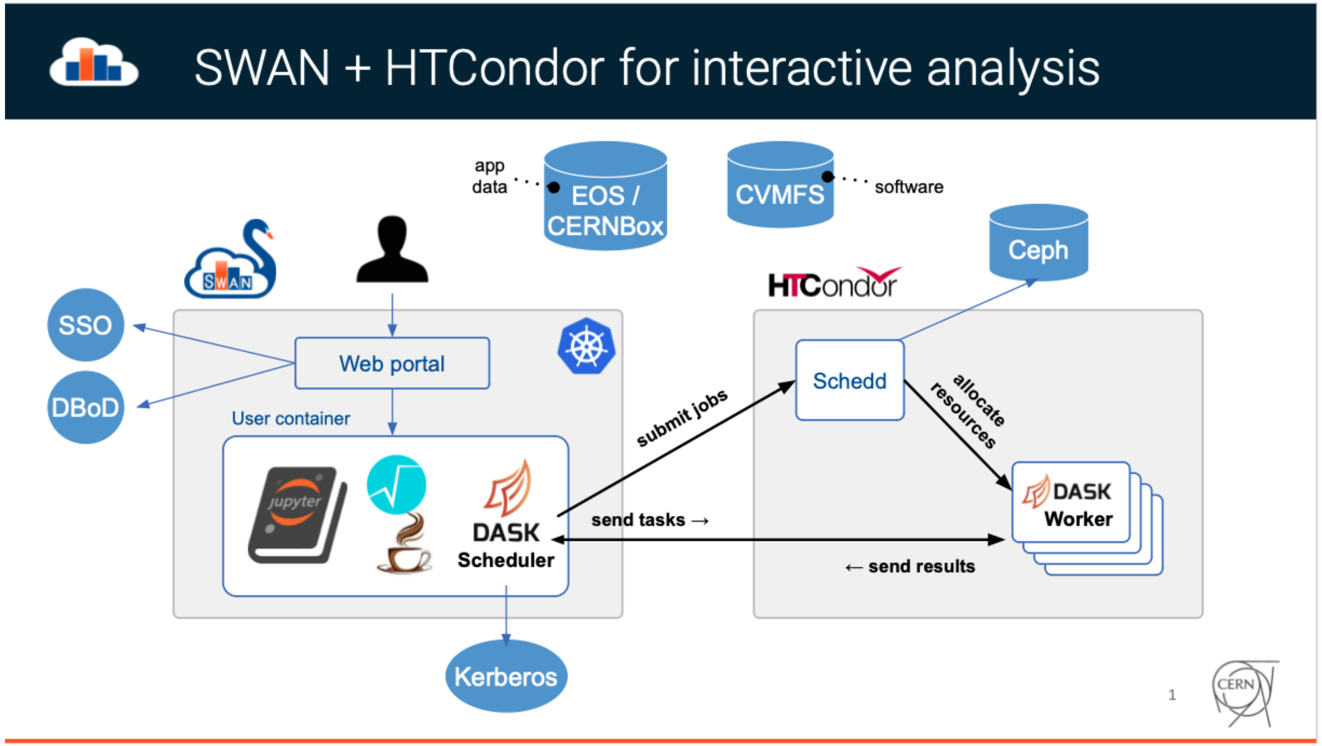
- **Results for 64 workers**
  - CPU limited
  - Caching layer irrelevant for performance
  - I/O is modest

# Next step: a CERN Analysis Facility Pilot

- **Our definition**
  - An infrastructure that enables users to run their columnar analysis code (hence based on Coffea or RDF) using primarily Jupyter notebooks as an interface and transparently using batch computing resources, where data can be accessed primarily from a local storage system, but when needed from external sites, possibly taking advantage of a local caching layer
- **Other work**
  - HEP Software Foundation Analysis Facilities White Paper sets the general features
  - CERN operating since years a Spark/HADOOP cluster for interactive analysis
  - Several AFs already existing elsewhere for LHC experiments
- **How to do it**
  - We already have (almost) everything in place

# Software components

- **SWAN**
  - In production since several years, integrates JupyterHub and now JupyterLab with LCG software stack and CERNBOX
  - Users log in to a web portal and their session is created in a Kubernetes cluster
- **DASK**
  - A library for distributed computing that support different batch systems among which HTCondor and is supported by Coffea and RDataFrame for creating and managing workers
- **HTCondor**
  - Used to manage all batch nodes
- **CVMFS**
  - To access software libraries, clients, etc.
- **EOS**
  - To access data

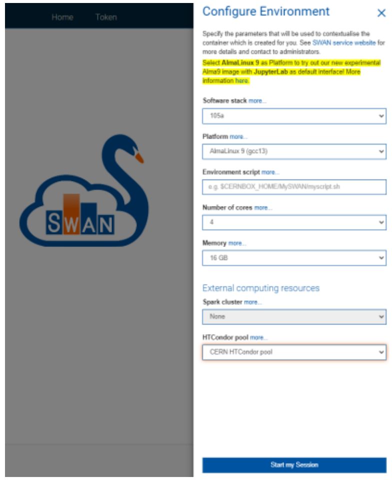


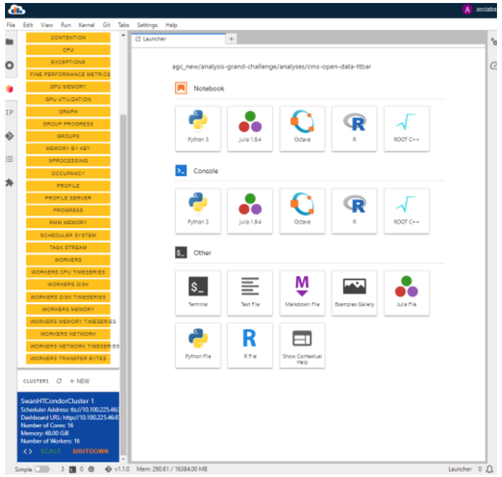
# Other motivations for a pilot

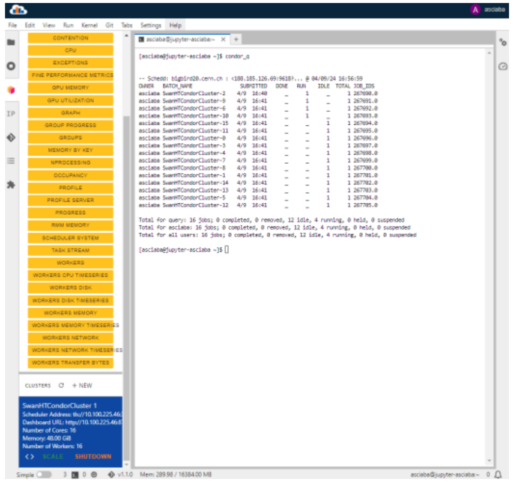
- **Optimise batch system utilization**
  - Overprovisioning of batch slots to be used by analysis workers (often idle)
- **Gauge the real demand for an AF**
  - Open to an initially small number of users
- **Put some assumptions to the test**
  - For example, that having a guaranteed buffer of CPU resources to accommodate simple requests and use current job priorities for users is already good enough
- **Find unforeseen limitations**
  - Never really tested in a multiuser environment

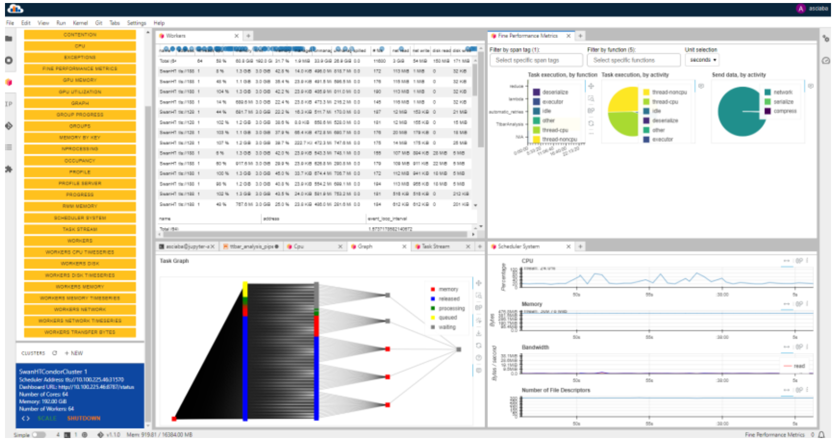
# Example workflow

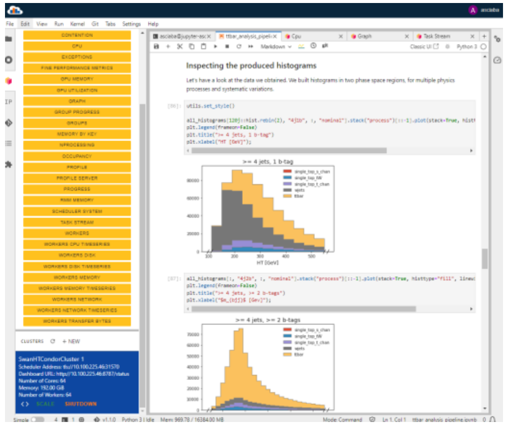
1. **Create a SWAN session**
   1. Select desired LCG software stack
2. **Create a DASK cluster via HTCondor**
   - and scale up to the desired number of workers
   - it will submit HTCondor jobs, which might take some minutes to start
3. **Execute analysis as notebook or code**
   - Run cells or Python script
   - Coffea or RDF will assign tasks to workers
4. **Monitor code execution**
   - Lots of metrics to show
5. **Check the results**
   - And repeat at will



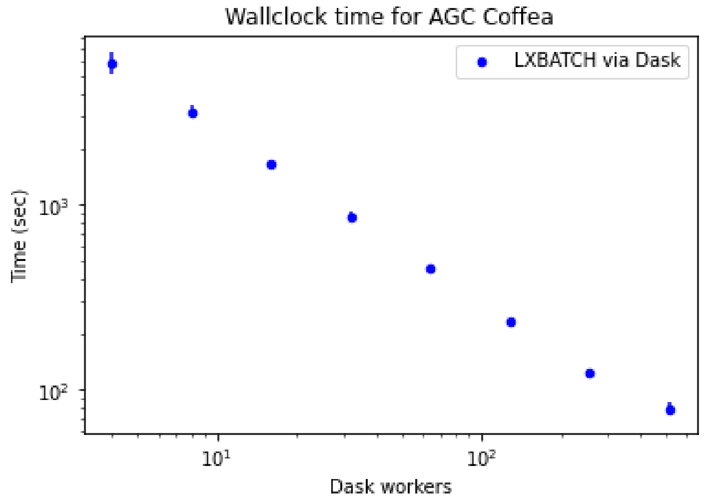









# Features and known limitations




- **Features**
  - Possible to disconnect and reconnect to a session without losing information
    - Within a few hours
  - Possible to convert a notebook into a script
  - Scalability with number of workers is excellent
  - Analysis software provided out-of-the-box by curated software stacks
    - Will investigate how to allow for custom software versions
- **Limitations**
  - Authentication/authorization requiring explicitly getting a Kerberos token but solution using Oauth tokens identified
    - Now need to manually get a Kerberos token to access compute and storage
  - XCache service to access datasets outside CERN not yet provided
    - Will be eventually if there is demand

# Conclusions

- **CERN computing infrastructure well suited for analysis, both batch and interactive**
  - As shown by extensive investigations using analysis workloads, EOS and HTCondor logs, etc.
- **Scale of interactive analysis still low but expected to increase**
  - Expectations for analysis facilities are much better defined than in the past
- **Time was right to set up an AF pilot service at CERN based on existing components**
  - Leverages on years of work and improvements of the DASK-HTCondor integration
  - Effort needed for user support unclear, but no issues seen so far
- **Some users have been invited to the pilot**
  - Online documentation provided (https://swan.docs.cern.ch/condor/intro/)
- **Will assess in ~6 months time based on user feedback**
  - Not yet a production service as of today

# References

- Analysis for LHC experiments at CERN: https://zenodo.org/record/6337728
- PrMon: https://github.com/HSF/prmon
- Analysis Grand Challenge: https://github.com/iris-hep/analysis-grand-challenge
- Coffea: https://coffeateam.github.io/coffea/
- AGC RDF implementation: https://github.com/andriiknu/RDF/