Analysis Grand Challenge benchmarking tests on selected sites

David Koch¹

Thomas Kuhr¹, Günter Dückeck¹, Dennis Noll², Benjamin Fischer²

¹Ludwig-Maximilians Universität München ²RWTH Aachen

CHEP 2023
- developed by the IRIS-HEP team
- effort to demonstrate feature-completeness and scalability of scikit-HEP tools
- main framework of the analysis: coffea, offers a high level interface for columnar analysis
- github, readthedocs

Scikit-HEP: Python ecosystem for HEP analyses
ttbar-Analysis includes

- 1-lepton event selection
- top reconstruction
- cross-section measurement
- on-the-fly evaluation of systematic uncertainties
Analysis Grand Challenge

ttbar-Analysis includes

- 1-lepton event selection
- top reconstruction
- cross-section measurement
- on-the-fly evaluation of systematic uncertainties
- total of 3.44 TB of CMS open data
- only $\sim$138 GB are actually read (4% of the total dataset)
- 948 mio events and 10 variables
- work is distributed across many workers with dask-jobqueue

... all this sits in a single Jupyter notebook $\Rightarrow$ analysis code is easy to use should also be scalable and fast

The AGC analysis is meant as a showcase of how a possible future HL-LHC analysis could look like
Talks at CHEP directly related to the AGC:

- Andrea Scabià, *I/O performance studies of analysis workloads on production and dedicated resources at CERN*, Monday 3pm
- Oksana Shadura, *Coffea-Casa: Building composable analysis facilities for the HL-LHC*, Tuesday 10am
- Alexander Held, *Physics analysis for the HL-LHC: concepts and pipelines in practice with the Analysis Grand Challenge*, Tuesday 5pm
- Vincenzo Padulano, *First implementation and results of the Analysis Grand Challenge with a fully Pythonic RDataFrame*, Tuesday 5:15pm
Benchmarks performed on three different sites:

**LMU** institute cluster at LMU Munich consisting of one very powerful node and desktop computers
job-scheduler: SLURM
reading of the data via xrootd from LRZ
Benchmarks performed on three different sites:

**LMU** institute cluster at LMU Munich consisting of one very powerful node and desktop computers
  job-scheduler: SLURM
  reading of the data via xrootd from LRZ

**LRZ** WLCG Tier-2 site in Munich
  job-scheduler: SLURM
  data is stored on regular Grid storage (HDD) as well as on a XCache server (SSD)
Benchmarks performed on three different sites:

**LMU** institute cluster at LMU Munich consisting of one very powerful node and desktop computers
- job-scheduler: SLURM
- reading of the data via xrootd from LRZ

**LRZ** WLCG Tier-2 site in Munich
- job-scheduler: SLURM
- data is stored on regular Grid storage (HDD) as well as on a XCache server (SSD)

**Vispa** analysis facility operated by RWTH Aachen; provides a web-based terminal, code editor and jupyter hub: [https://vispa.physik.rwth-aachen.de](https://vispa.physik.rwth-aachen.de)
- job-scheduler: HTCondor
- data is stored locally on SSDs and read directly; Vispa also has a very dedicated caching-system (*arXiv*) that I did not test with AGC yet
Measurements

Runtime

Only the part of the analysis which is run distributed is used for the benchmark → fetching some metadata and reading and processing the data.

- the total runtime
- the total processing time
- the sum of all process times across the workers (the sum of all blue rectangles here) via coffea's tooling
Measurements

Runtime

Only the part of the analysis which is run distributed is used for the benchmark → fetching some metadata and reading and processing the data

I can measure directly
• the total runtime
• the total processing time
• the sum of all process times across the workers (the sum of all blue rectangles here) via coffea’s tooling
Measurements

Runtime

The diagram shows the total runtime in s for different numbers of workers, labeled as lmu, lrz, and vispa. The x-axis represents the number of workers, ranging from 20 to 140, while the y-axis represents the total runtime in seconds. The chart indicates how the total runtime decreases as the number of workers increases.
Measurements

Runtime

metadata-fetch time (\(=\) total - process time, also contains waiting and communication between dask workers) and process time
Measurements

Runtime

measure for the amount of overhead relative to the runtime

\[
\frac{t_p}{\sum t_{p_i}/n}
\]

\(\sum t_{p_i}/n =: \bar{t}\) is the average process time per worker – *pure computing time without overhead*
measure for the amount of overhead relative to the runtime

\[ \frac{t_p}{\sum t_{p_i}/n} \]

\[ \sum t_{p_i}/n =: \bar{t} \text{ is the average process time per worker} - \textit{pure computing time without overhead} \]
Measurements

XCache

runtimes at LRZ with and without XCache enabled: makes no significant difference ⇒ with this setup, the analysis is hardly I/O limited
Questions?
Backup
Measurements

Runtime

The graph shows the cumulative runtime of all processes across all workers for different numbers of workers. The y-axis represents time in seconds (s), while the x-axis shows the number of workers. Three different systems are compared: lmu, lrz, and vispa. The data points indicate the summed process time across all workers for each system.

The graph highlights how the runtime changes with the number of workers, demonstrating the impact on performance for each system.
Measurements

Runtime

Scaling of processing speed

Scaling of process time

David Koch
CHEP 2023
May 8 2023
Measurements

XCache

XCache load during two consecutive runs with 50 workers
example Dask dashboard with long tail
total runtimes during the day and night @Vispa