PHYSLITE dask tests (on Fernando's jupyter hub)

Nikolai Hartmann

LMU Munich

June 9, 2021, ATLAS - Google Technical Meeting

Reminder: Data set and analysis

- 100 TB dataset with ATLAS LHC Run2 data in derived format
  $\rightarrow$ DAOD_PHYSLITE: small analysis format, calibrations applied
- Distributed across 260k files, 18e9 events in total
- Stored in ROOT format, columns split
  $\rightarrow$ potential for conversion to parquet
- Example analysis using uproot and awkward array:
  - Apply selection criteria for analysis objects: Electrons, Muons, Jets
  - Perform overlap removal (involves combinatorics)
  - Can then calculate simple observables, fill histograms
  - Reads $\approx$ 10% of the stored data
    $\rightarrow$ but rather scattered reading: basket (compressed block) sizes in the order of 5-50kb
  - Maximum throughput when reading from memory: 10k events per second
    (still mostly dominated by decompression/deserialization)

$\rightarrow$ already ran sucessfully on google panda queue for 1% and 10% of the dataset

Data access

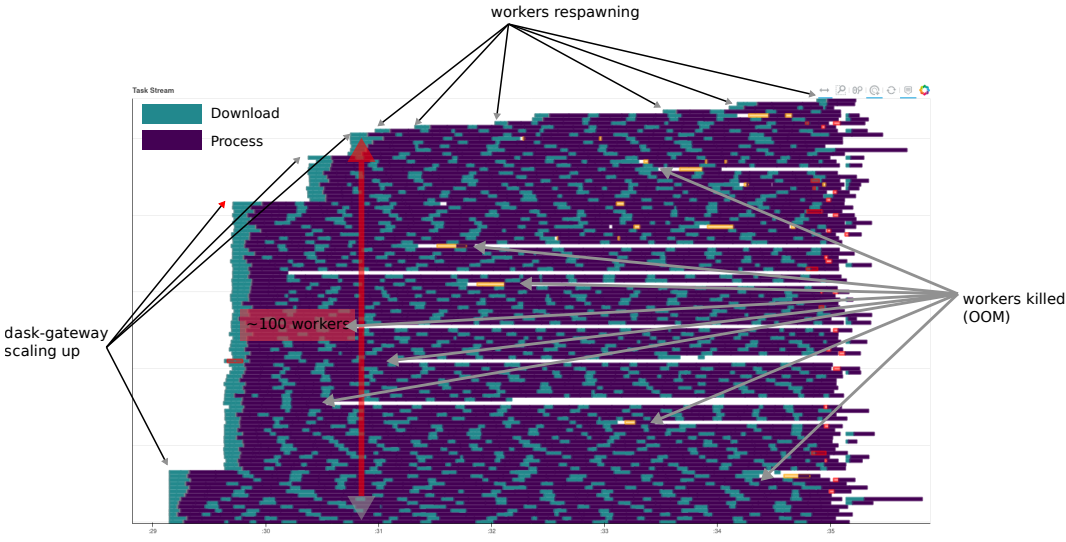
- Use signed urls via rucio query
  → takes a bit of time
  → run `list_replicas` in 10 threads for the 10% dataset (25k files)
  → set `signature_lifetime` manually
  (default seems to be 10min for some 1h for others??)
- In jupyter: Upload x509 proxy certificate
- HTTP multirange not supported
- HTTP single range requests work
  → uproot does not do connection pooling by default
  → experimental implementation worked more or less
  (using 10 concurrent connections and asyncio loops)
  → still slower than whole file download

→ use whole file download (download into memory) for now
(also used on the panda queue)

Dask setup/configuration

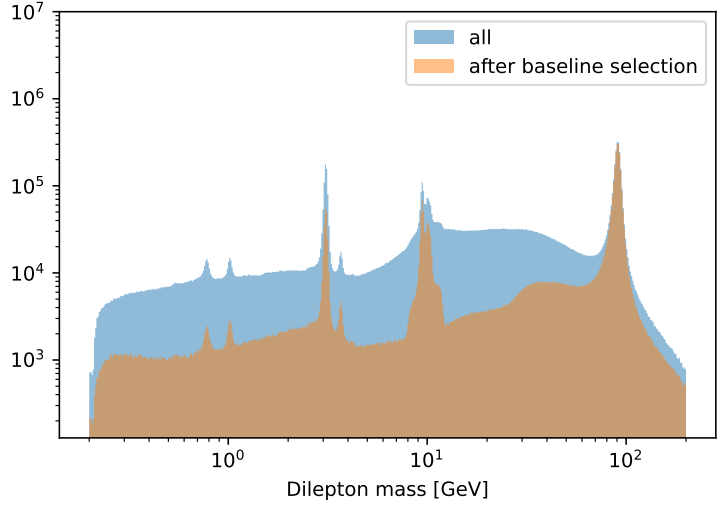
- Using Fernando's dask gateway setup
  $\rightarrow$ works nicely, workers come up fast
- One dask worker per core, one thread per worker
  $\rightarrow$ faster than with multiple threads (lot's of python stuff going on)
  (reconsider, also means less memory per worker)
- 4-5 GB memory per worker (need to be careful with GB and GiB)

1% Test

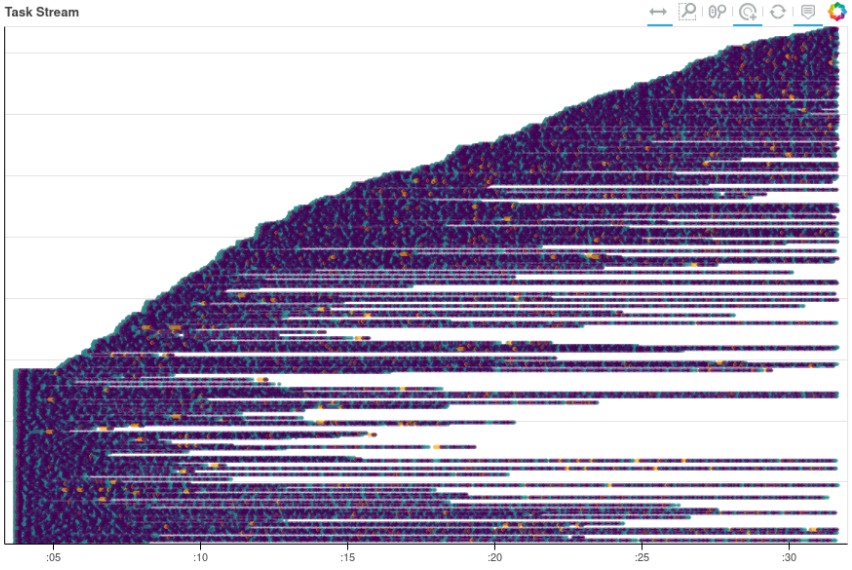


$\rightarrow$ runs fine in $\approx$ 6 minutes on 100 workers ($\approx$ 2500 files)

The “history of particle physics” plot



10% Test



→ too many OOM failures ...

Conclusions and next steps

- Dask gateway setup works well
- Nice bandwidth to google cloud storage
  $\rightarrow$ no degradation with 100 parallel dask workers
- Need to get memory (leaks?) under control
  $\rightarrow$ probably mainly related to uproot
- Unfortunately no clue where to start (fixes for all things i found so far are already included)
  $\rightarrow$ need to dig in further, try to come up with reproducible examples
  (to show Jim Pivarski)
- Dask does not allow to spawn python subprocesses on workers
  $\rightarrow$ can't do quick & dirty fix by running uproot loading in subprocess
  $\rightarrow$ although some super ugly workarounds (call python script) might be possible
- Maybe part of the problem are large files
  (some have up to 1.5GB, few outliers with $\approx$ 2GB)
  $\rightarrow$ try to download to disk instead (do the dask nodes have disks?)
- Maybe try in parallel to run on parquet files
  $\rightarrow$ Can we run a bulk conversion on the panda queue and store output on google cloud?