PREPARATION FOR ALICE DATA PROCESSING AND ANALYSIS IN LHC RUN 3

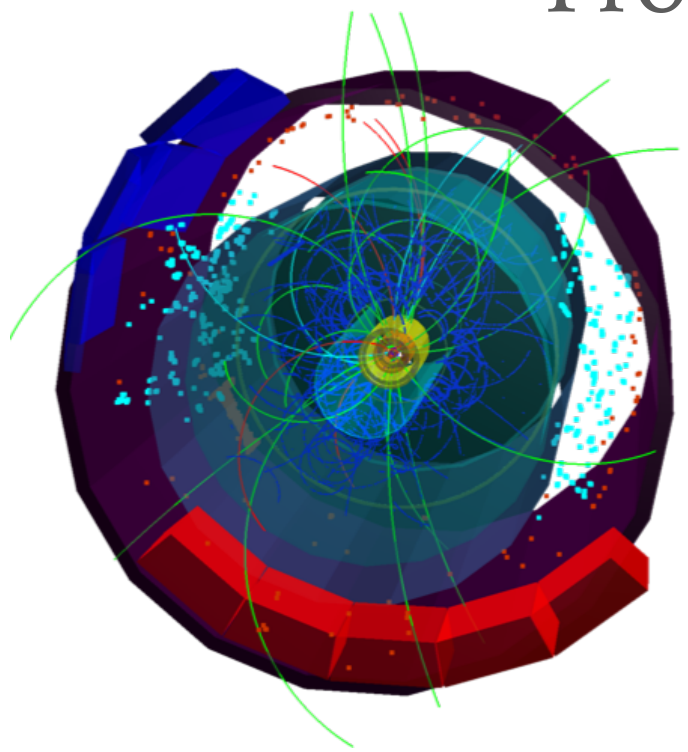
Giulio Eulisse (CERN EP-AIP) for the ALICE Collaboration

ALICE

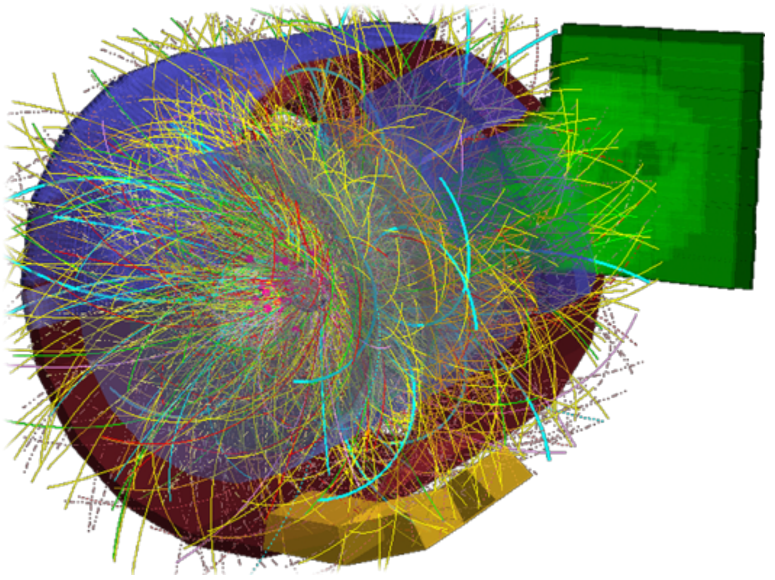
Challenges for ALICE in Run 3

- ***Reconstruct 100x more*** *events online.*
- ***Store 100x*** *more events (needs factor 36x for TPC compression).*
- *Reconstruct TPC data in* ***continuous readout*** *in combination with triggered detectors.*
- ***Completely new detector readout*** *and* ***substantial detector upgrades:*** *new ITS, MFT, FIT. New GEM for TPC readout.*
- *WLCG "flat budget" scenario (4x more resources over 10 years, for 100x more events).*

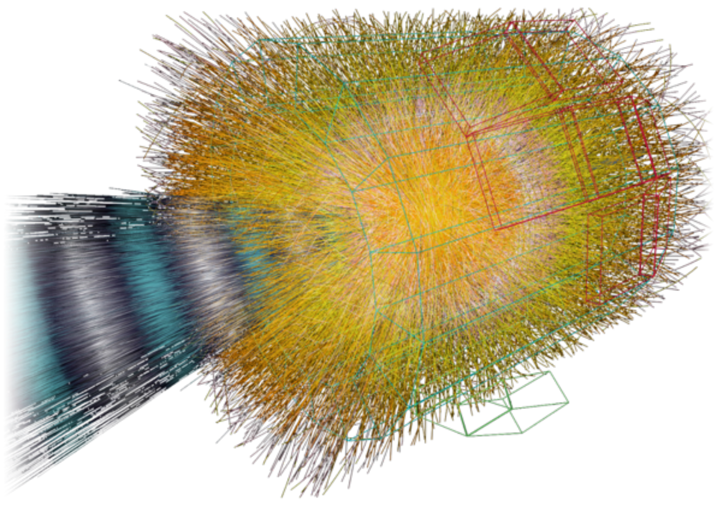
From < 1 kHz single events…



pp

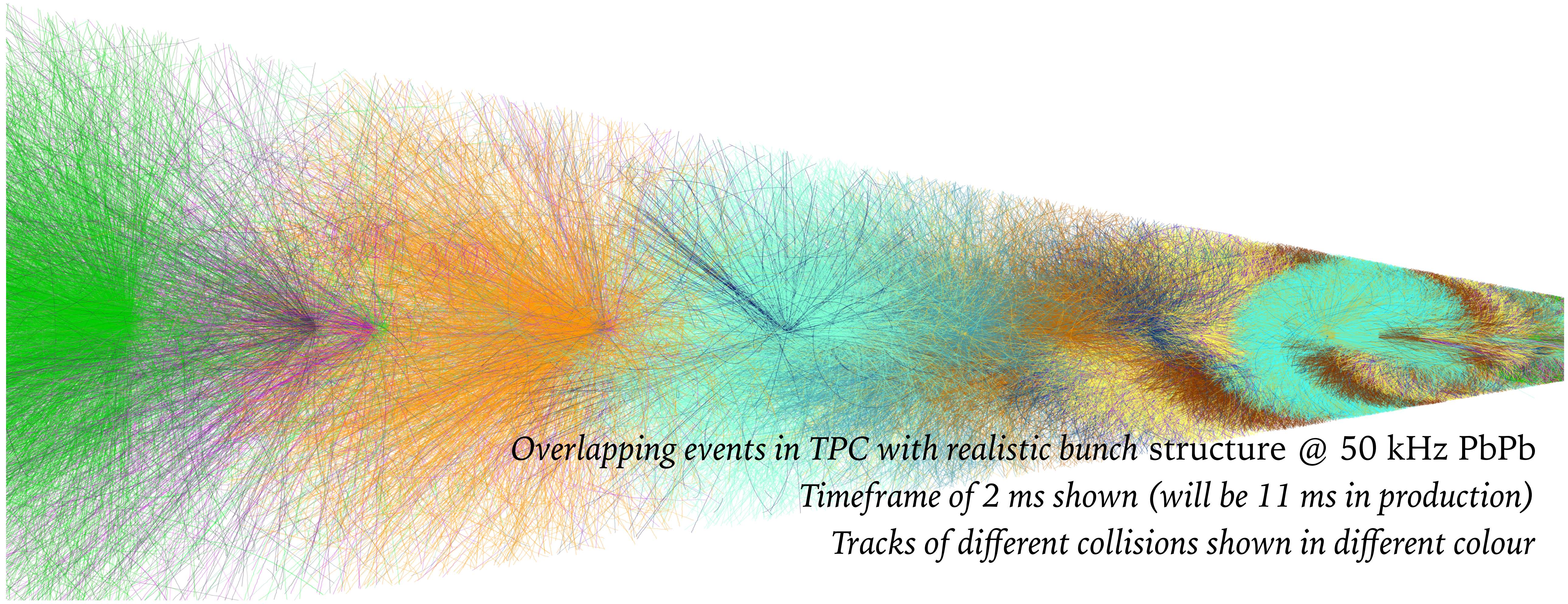


pPb



PbPb

…to 50 kHz of continuous readout data (in PbPb).



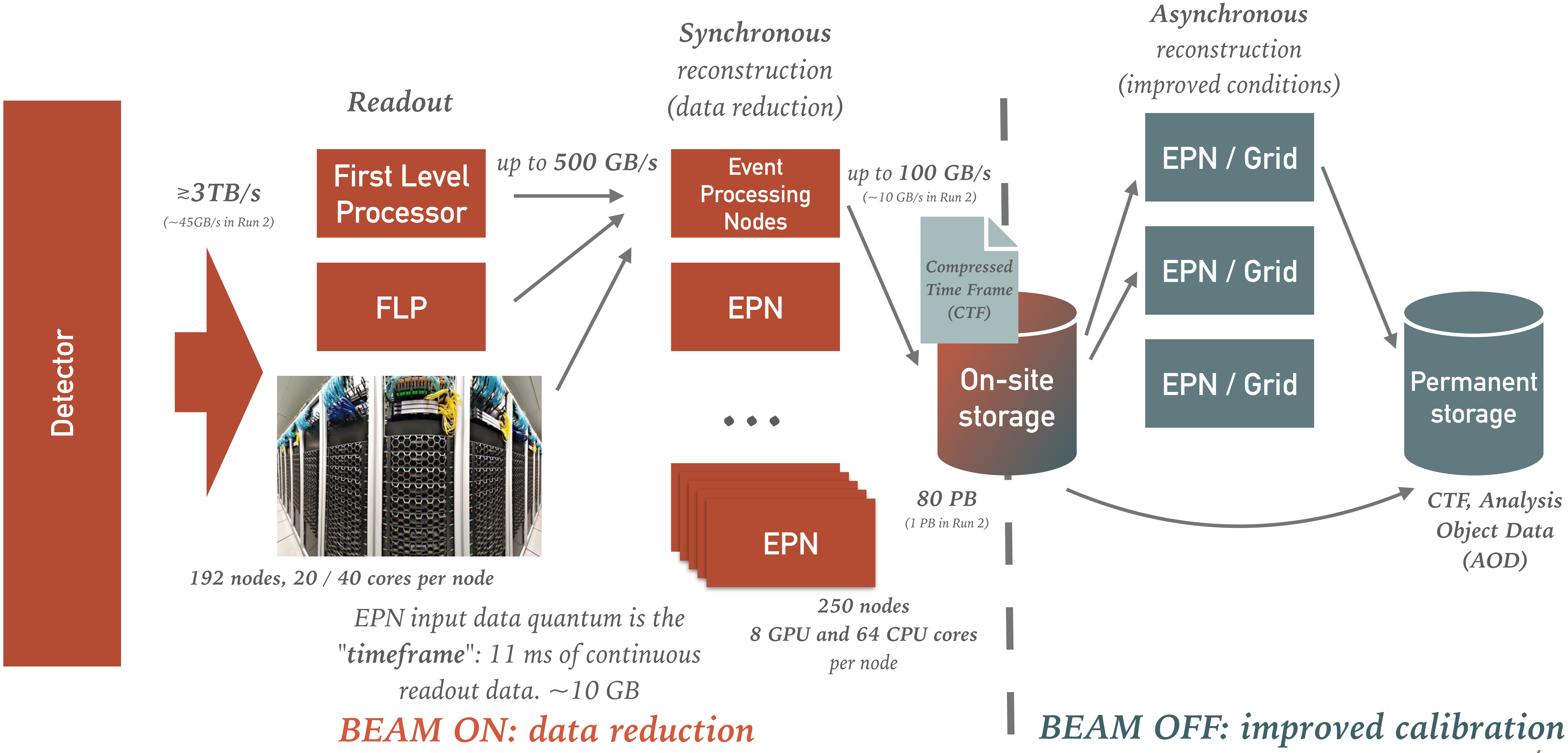
Overlapping events in TPC with realistic bunch structure @ 50 kHz PbPb
Timeframe of 2 ms shown (will be 11 ms in production)
Tracks of different collisions shown in different colour

A NEW COMPUTING ARCHITECTURE FOR ALICE IN RUN 3: ALICE O^2

ALICE can cope with the challenges of Run 3 only by a radical redesign of its software and computing architecture.

- *New architecture based on the experience accumulated in the ALICE HLT during Run 1 / Run 2.*
- *Focus on online data compression, only analysis objects readily available, **trading computational cost for storage**.*
- ***Simplified Data Model** to improve I/O performance.*
- ***Appropriately chosen algorithms** tuned for vectorisation and GPUs.*
- ***Close collaboration with the physics community** to organise analysis efforts.*
- ***Close collaboration with GSI and FAIR** on a common software stack.*

ALICE in Run 3: Point 2



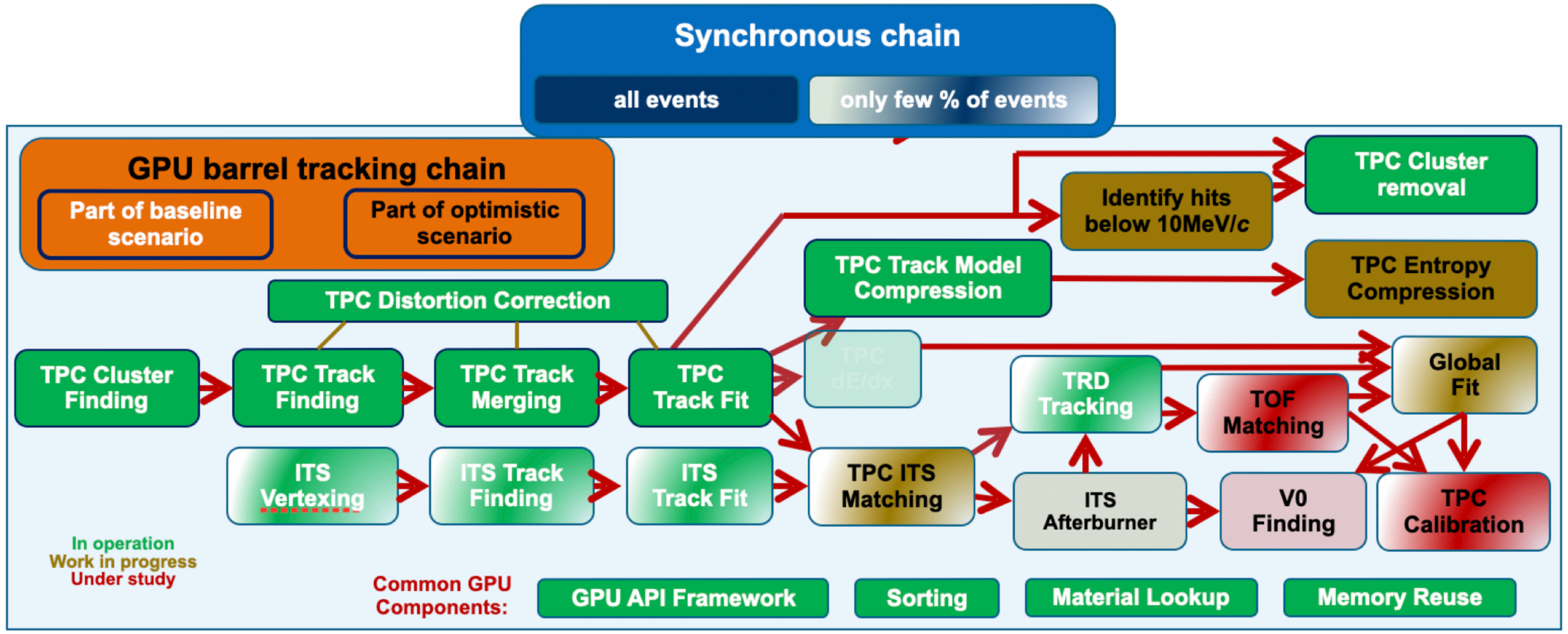
Synchronous Reconstruction: GPUs as First Class Citizens

Synchronous processing requires GPU utilisation for TPC tracking. One modern GPU replaces 40 CPU cores. Changing the algorithm gives an additional 20x - 25x speedup. ***GPUs provide a 4x total benefit in terms of cost.***

ALICE will use ~250 dual AMD Rome for a total of 64 cores, each equipped with 8 AMD MI50 32 GB GPUs. 1500 GPUs needed to process @ 50 kHz, 30% margin.

Besides TPC tracking, baseline foresees running most of ITS tracking on the GPU. ***99% of the computing in synchronous phase already running on the GPU.***

*Same source code can targeted to support different GPU middlewares (**AMD HIP**, nVIDIA CUDA, OpenCL) or CPU (mostly for debugging and validation).*



Asynchronous Reconstruction

Follows the PbPb data taking, interleaved with pp. **Two processing cycles** per data taking year.

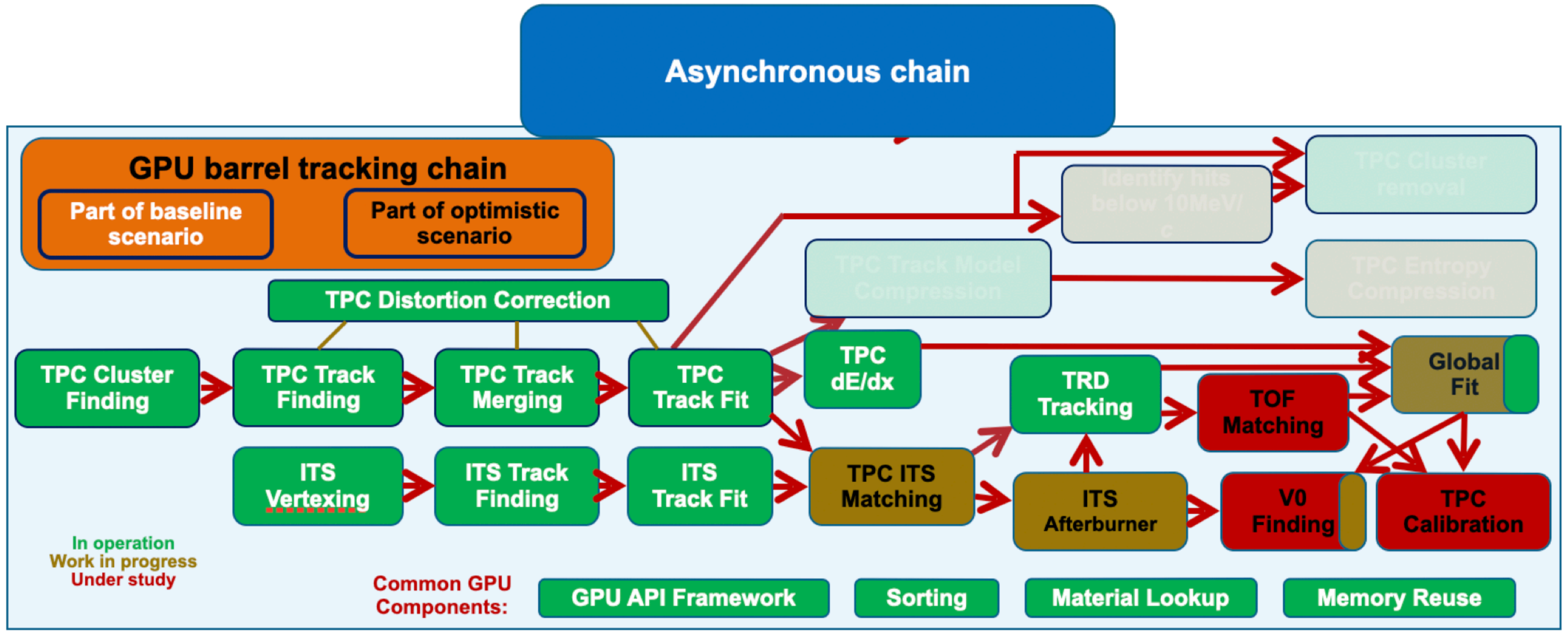
Processing on EPN farm (2/3 CTF volume) and the Grid (1/3).

Currently over 80% of the CPU - equivalent computing time running on GPUs. GPU usage is crucial to effectively use EPN farm when not taking data.

After 2nd cycle CTF will remain only on tape. Any subsequent cycle will have to wait until LHC LS.

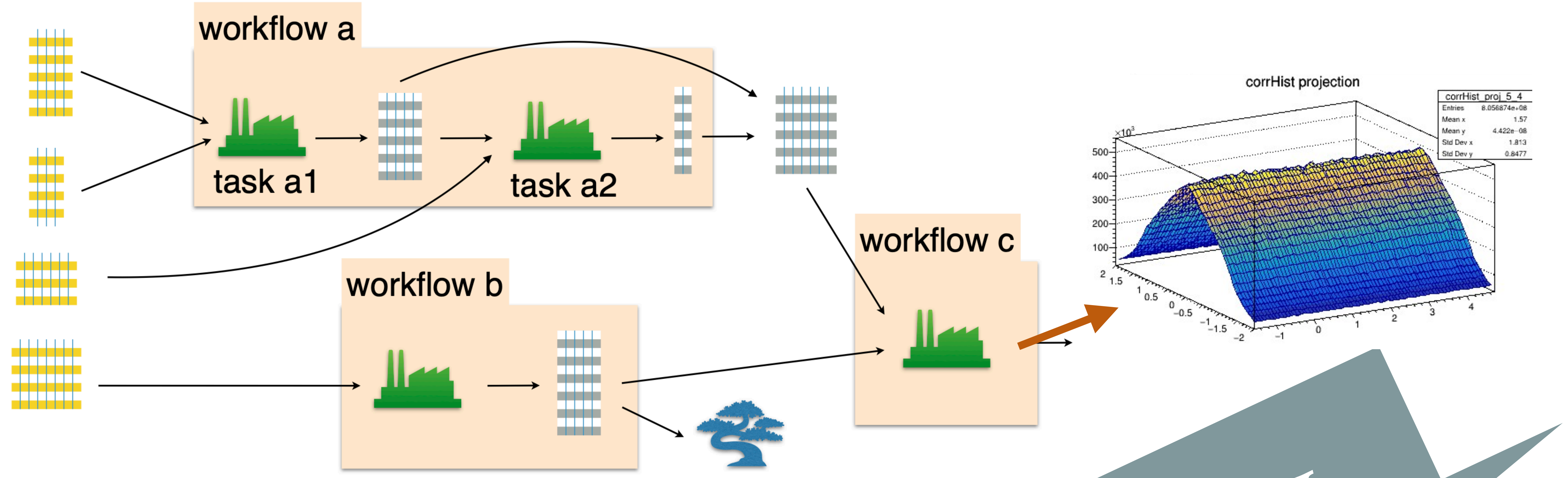
Single persistent analysis object output - Analysis Object Data. All the analysis will have to be performed on such data and the associated derived objects.

20 PB of EOS disk cache already benchmarked and ready for commissioning.



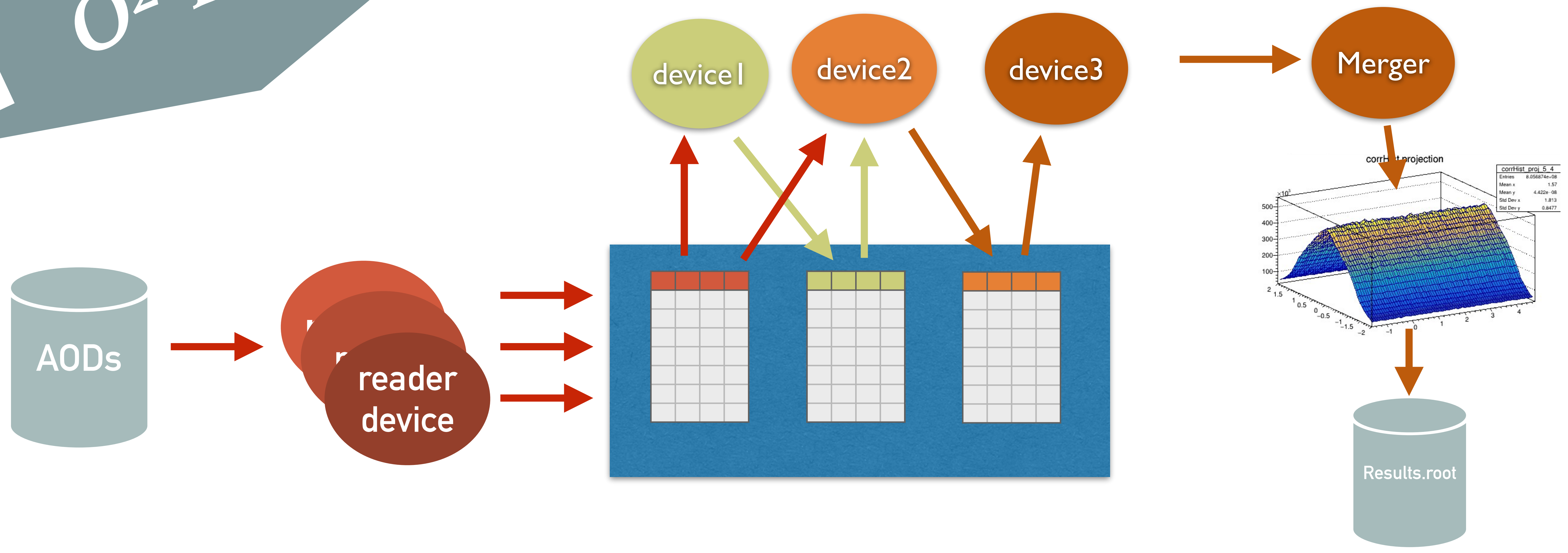
A picture of a running system is worth more than any number of words!

O² Data Processing Layer



User provides a description in terms of tasks and physics quantities.

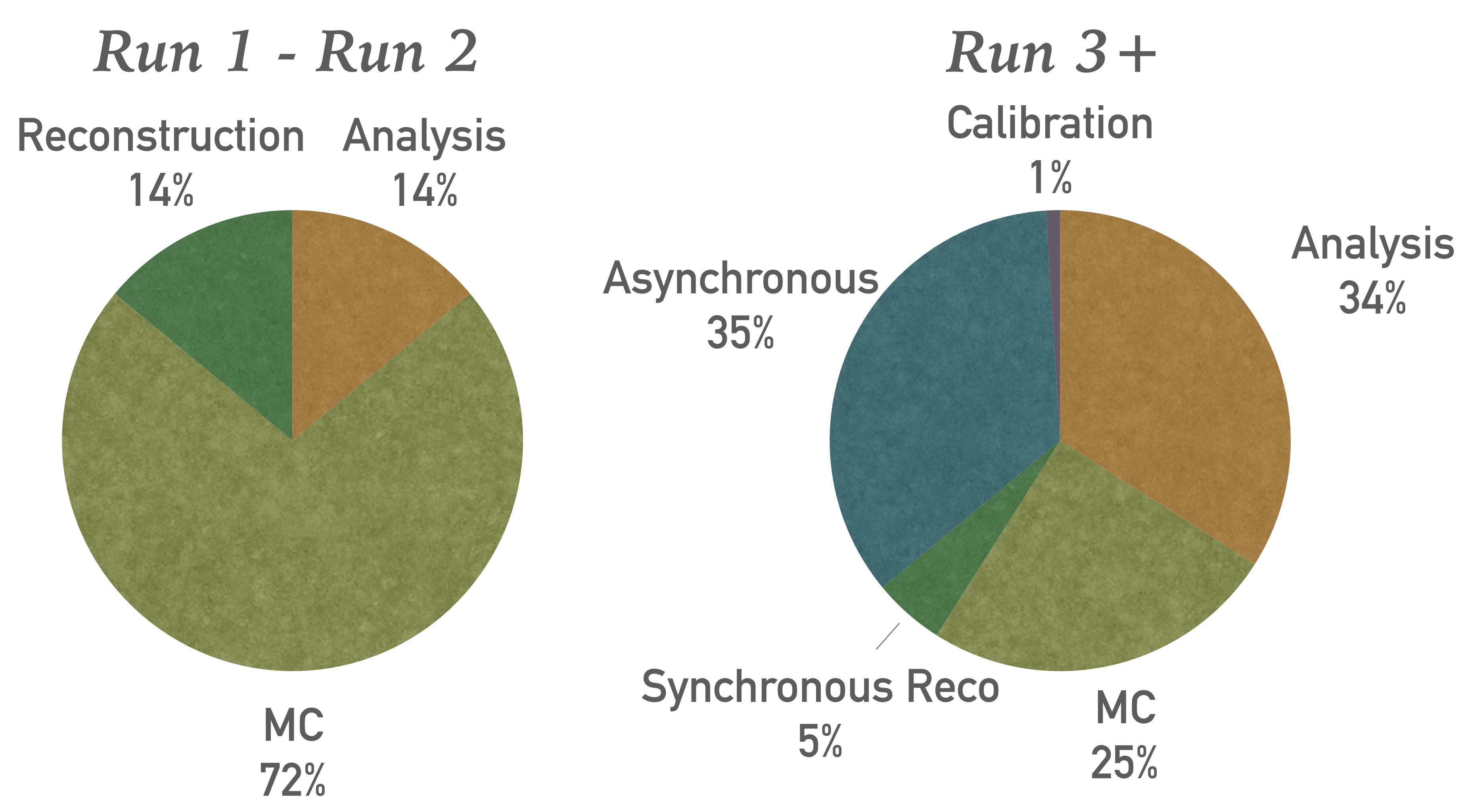
O² Data Processing Layer (DPL) translates the implicit workflow(s) defined by physicists to an actual FairMQ topology of devices, injecting readers and merger devices, completing the topology and taking care of parallelism / rate limiting.

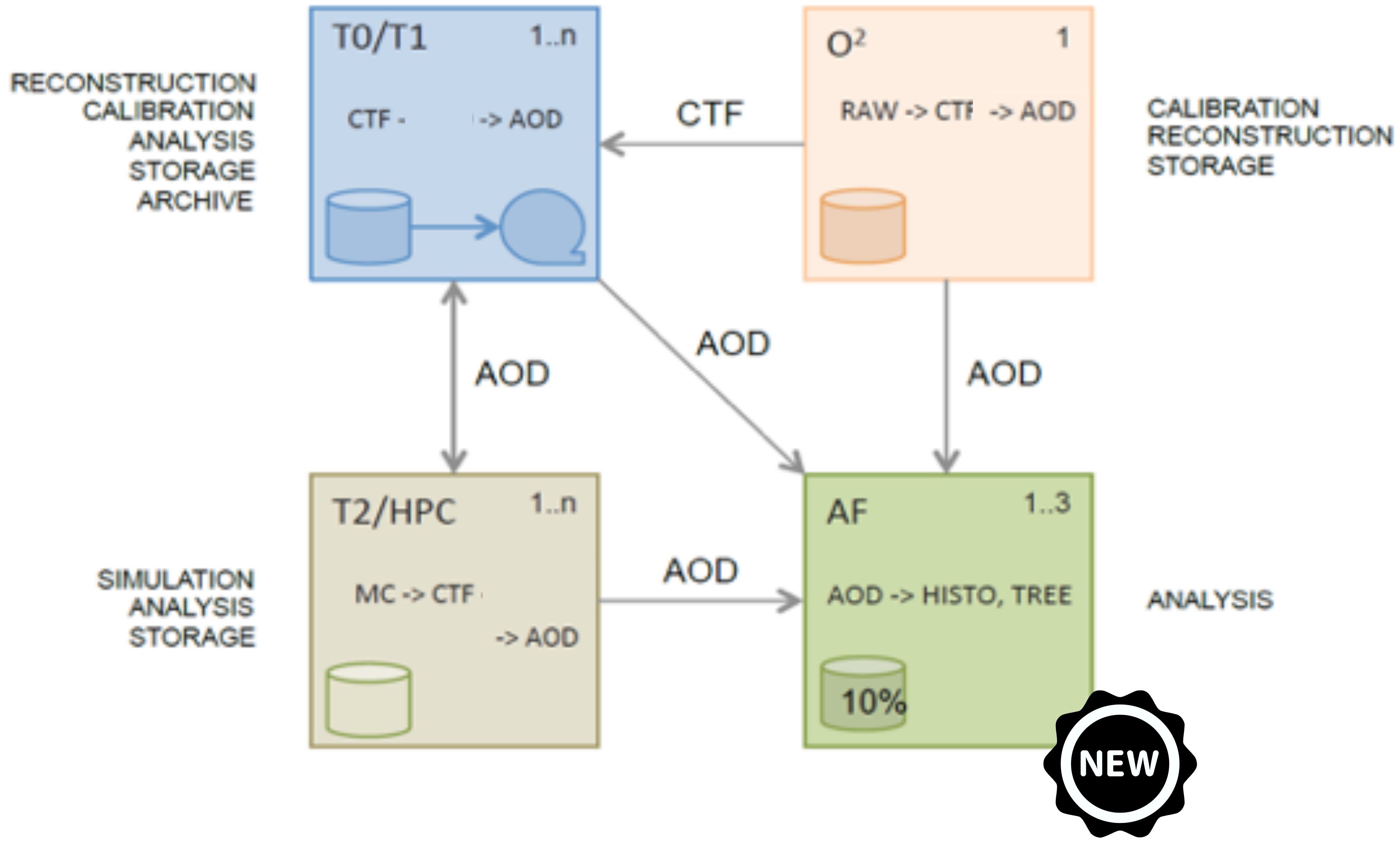


Analysis Model in Run 3

Solid foundations: *the idea of organised analysis (trains) will stay. Improve on the implementation.*

- ***x100*** *more collisions compared to present setup,* ***AOD only.***
- *Initial analysis of 10% of the data at fewer* ***Analysis Facilities,*** *highly performant in terms of data access.*
- ***Streamline data model****, trade generality for speed, flatten data structures.*
- ***Recompute*** *quantities on the fly rather than storing them. CPU cycles are cheap.*
- ***Produce highly targeted ntuples*** *to reduce turnaround for some key analysis.*
- ***Goal from TDR is to have each Analysis Facility go through the equivalent of 5PB of AODs every 12 hours (~100 GB/s).***





Analysis Framework

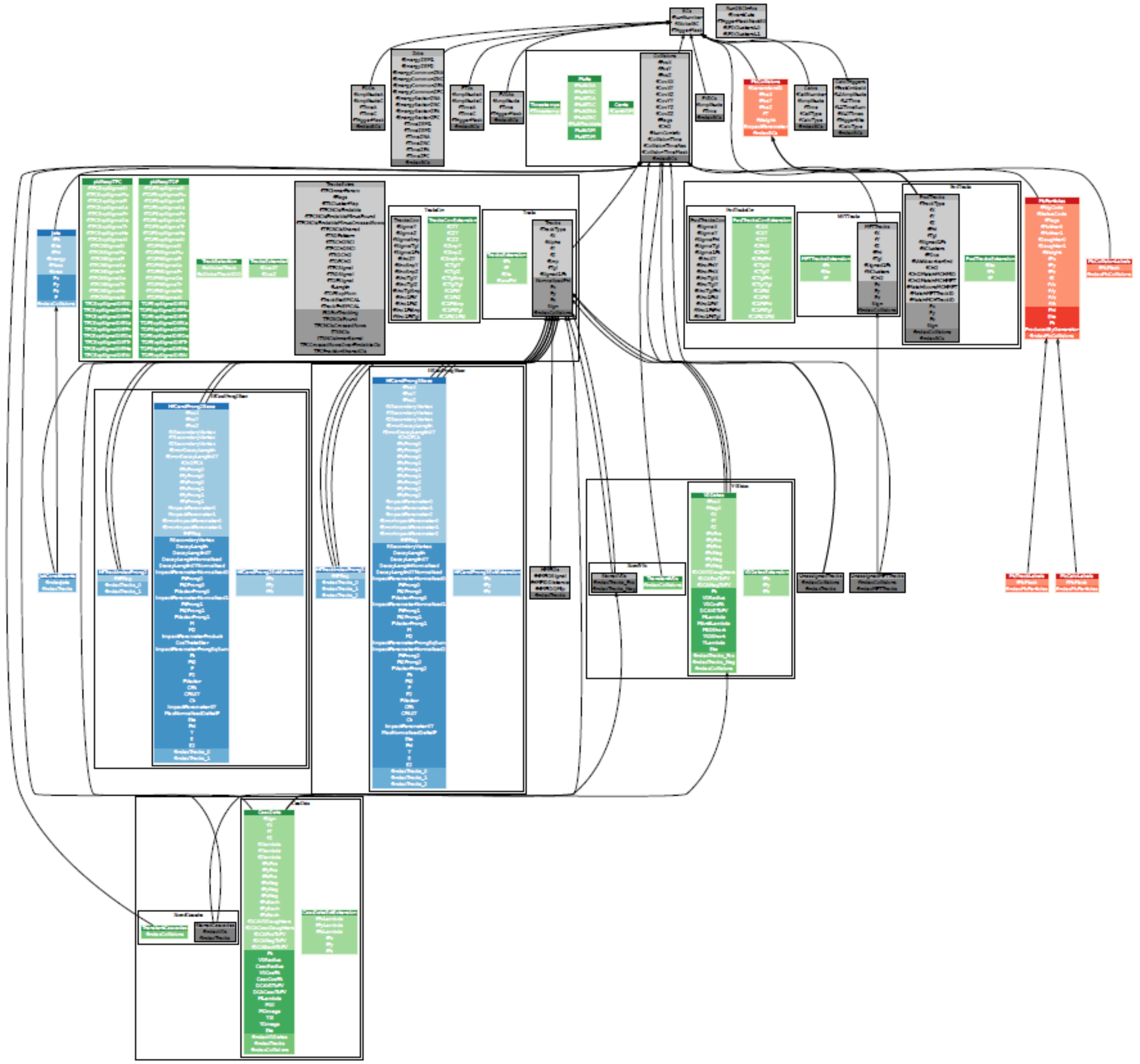
We have completely rewritten our analysis software to be able to run on top of the same software stack, the O² DPL, simplifying the data model while doing so.

Each Analysis Task is now a DPL device, taking advantage of its innate parallelism.

Cross indexed flat tables rather than hierarchy of objects.

Objected Relation Mapping (ORM) API provided to hide backing store and use `track.pt()`.

A declarative API, to easily define filters, joins and expression, providing efficient bulk manipulations.



Analysis Framework

We have completely rewritten our analysis software to be able to run on top of the same software stack, the O² DPL, simplifying the data model while doing so.

Each Analysis Task is now a DPL device, taking advantage of its innate parallelism.

Cross indexed flat tables rather than hierarchy of objects.

Objected Relation Mapping (ORM) API provided to hide backing store and use `track.pt()`.

A declarative API, to easily define filters, joins and expression, providing efficient bulk manipulations.

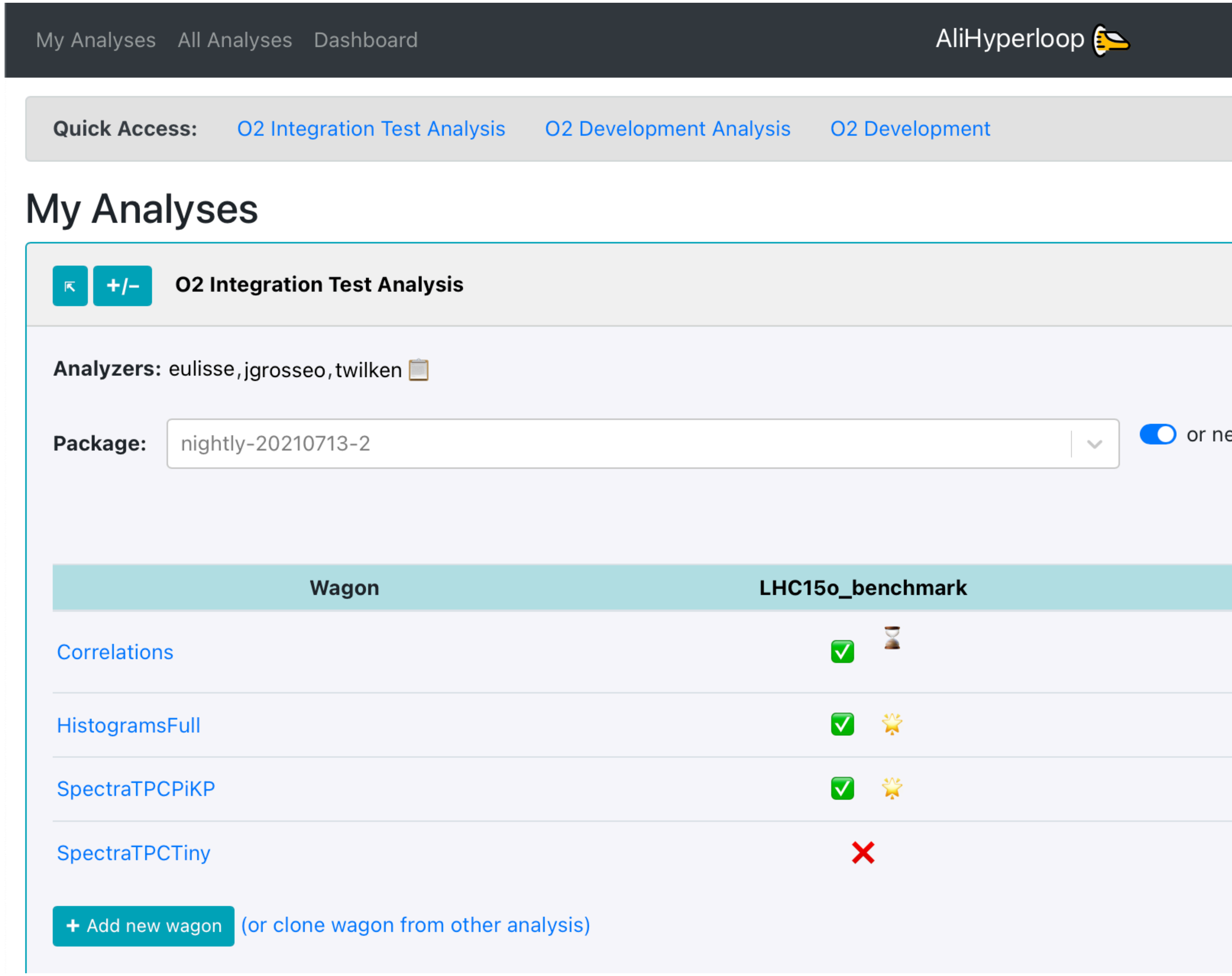
Declarative filters. Can be precomputed and vectorised by the framework

```
Filter vertexFilter = nabs(collision::posZ) < 7;
Filter ptFilter = track::pt > 0.5f;

void process(Collision const& collision,
             Tracks const& tracks)
{
  // some complex event selection
  // which does not work declarative

  for (auto& track : tracks)
    hist.Fill(track.pt());
}
```

Imperative part: user has almost the same freedom as a classic object oriented framework

Analysis Trains

- ALICE has a tradition of organised analysis (trains), which are scheduled together to run on the Grid, amortising per task access to storage cost.
- It integrates Grid job submission with bookkeeping and shields the users from the mechanics of resubmitting and merge.
- Extremely popular among ALICE users (>90% of Run 1 / Run 2 analysis).
- Revamped web interface with better profiling abilities, the ability to (de)compose trains to optimise throughput / resource utilisation.



Take Away Points

- **x100 more data with only x4 more (Grid) resources in 10 years.**
- New software & computing architecture to cope with it.
- **GPUs are critical for ALICE ability to process data in Run 3.**
- ALICE physics community is busy porting code to the new framework, with a mixed imperative / declarative paradigm being used.
- ALICE "Trains" infrastructure is being upgraded to take advantage of the new framework as well.

BACKUP

WLCG pledged resources

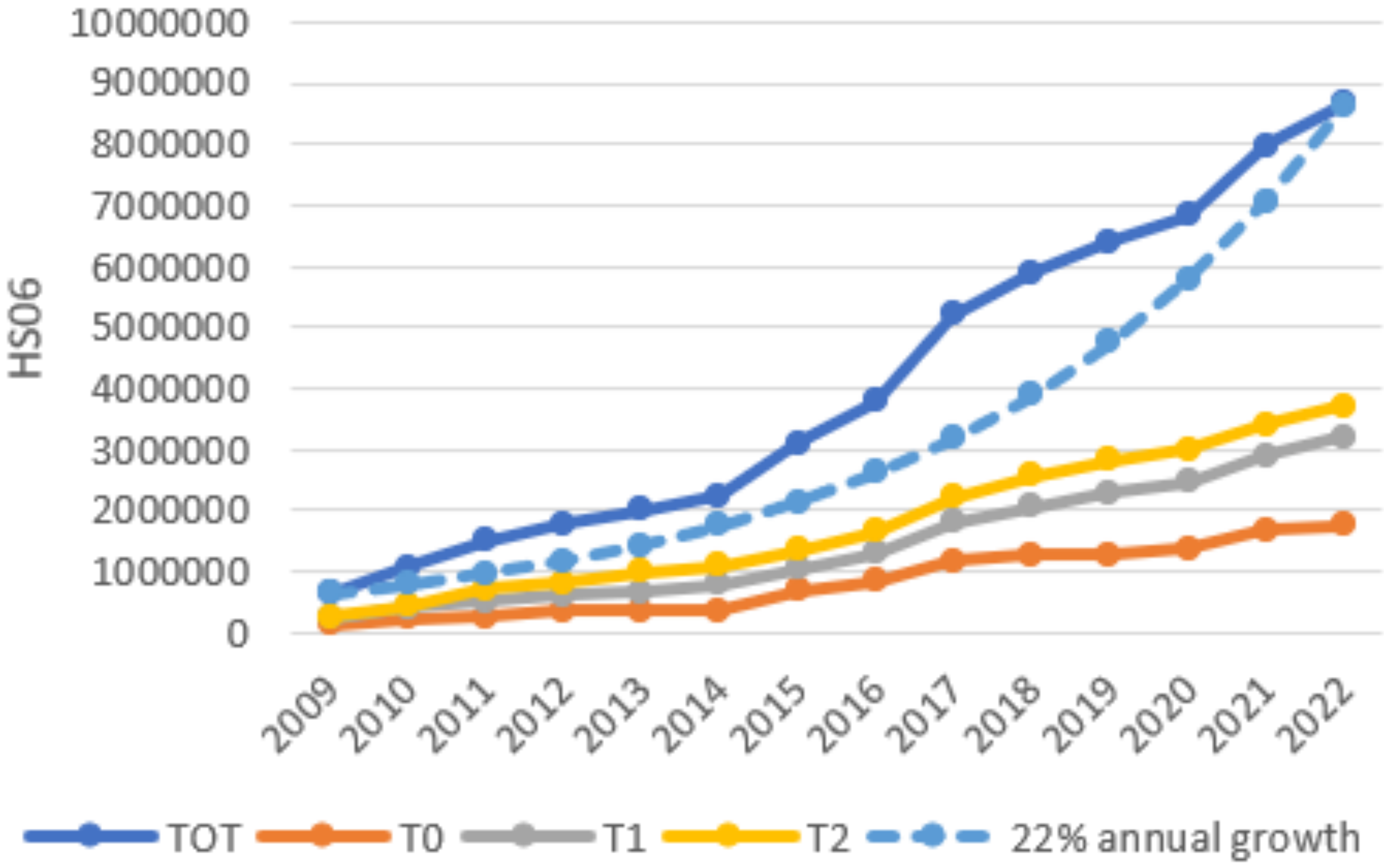
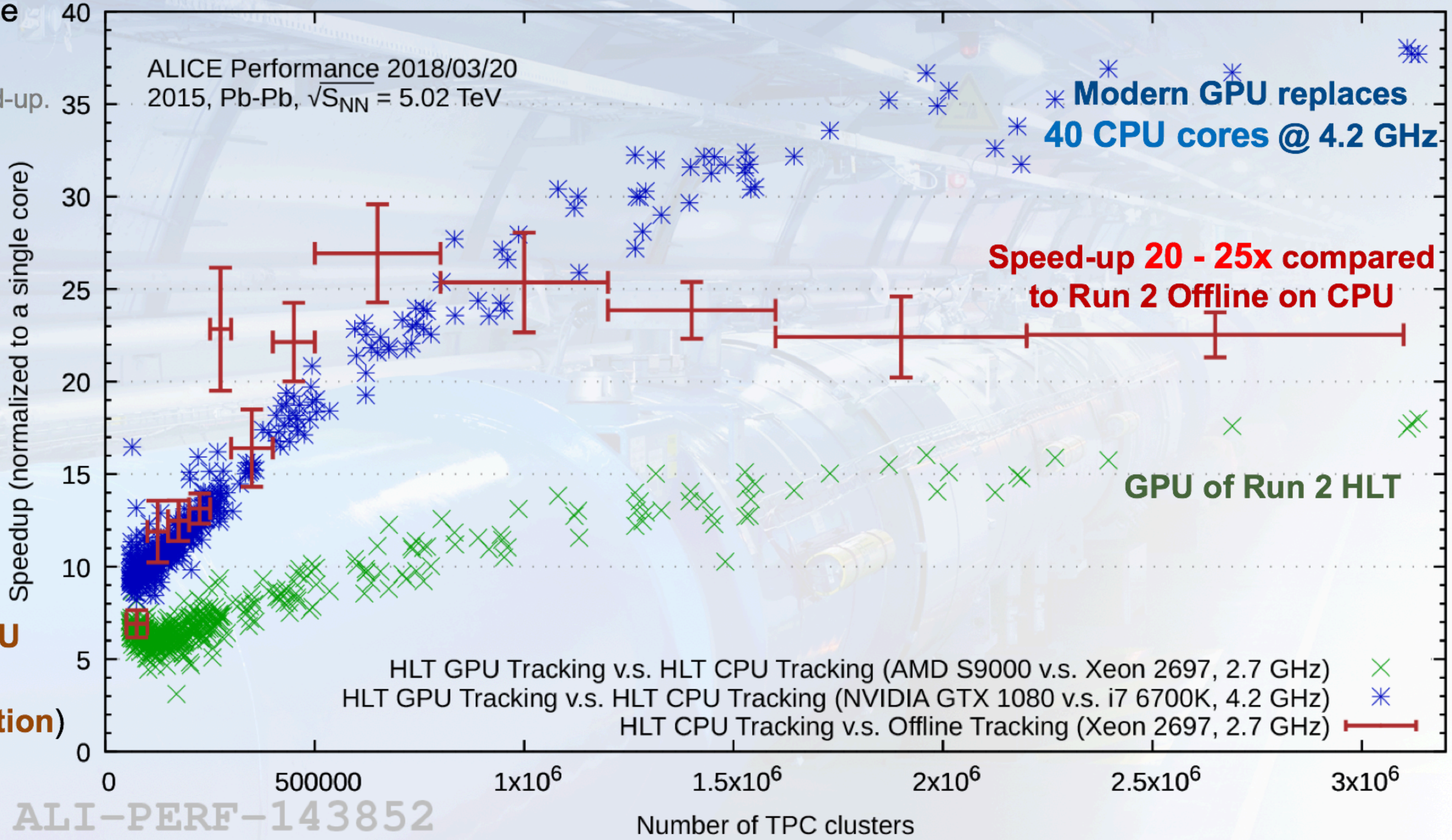
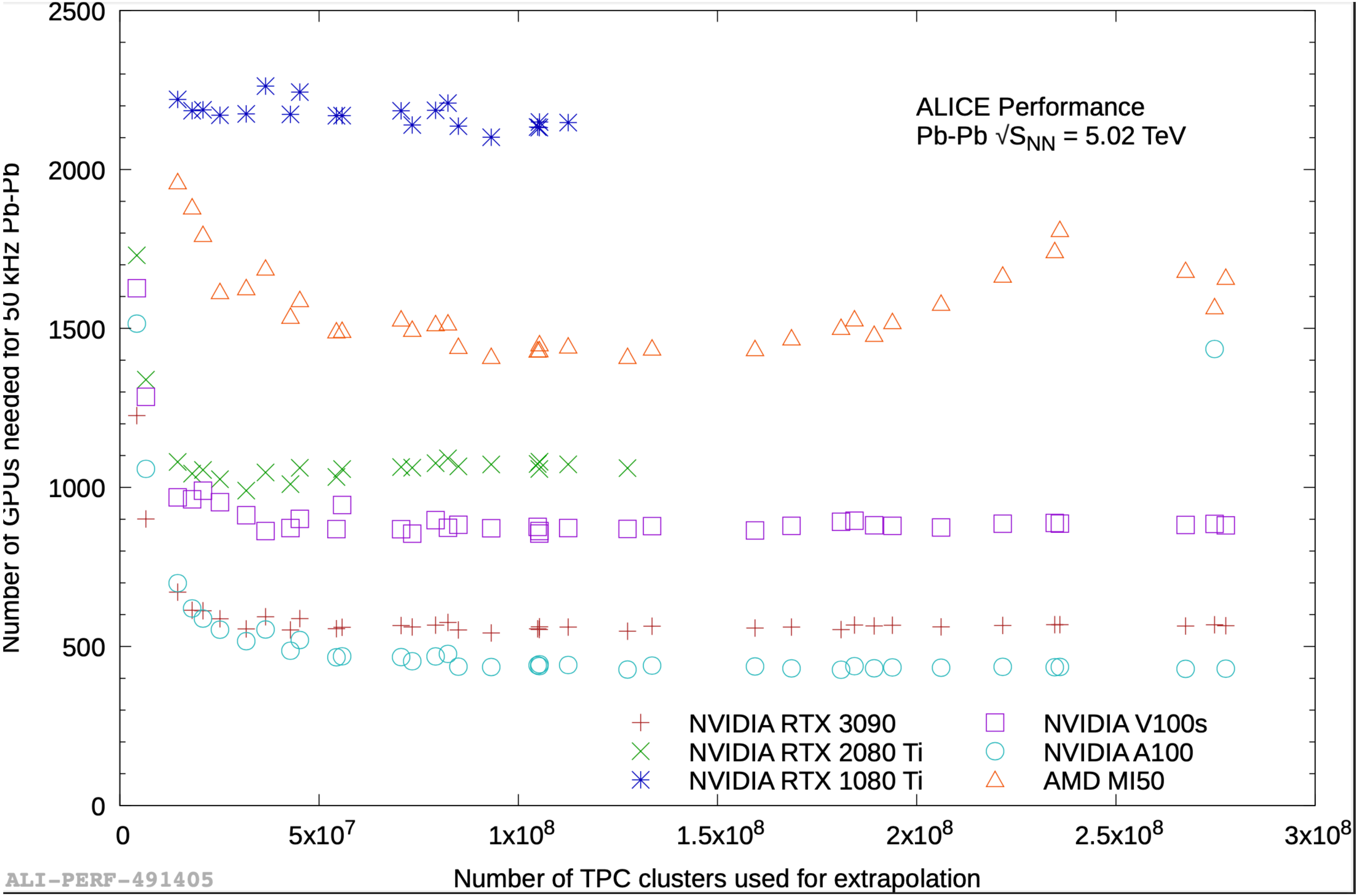


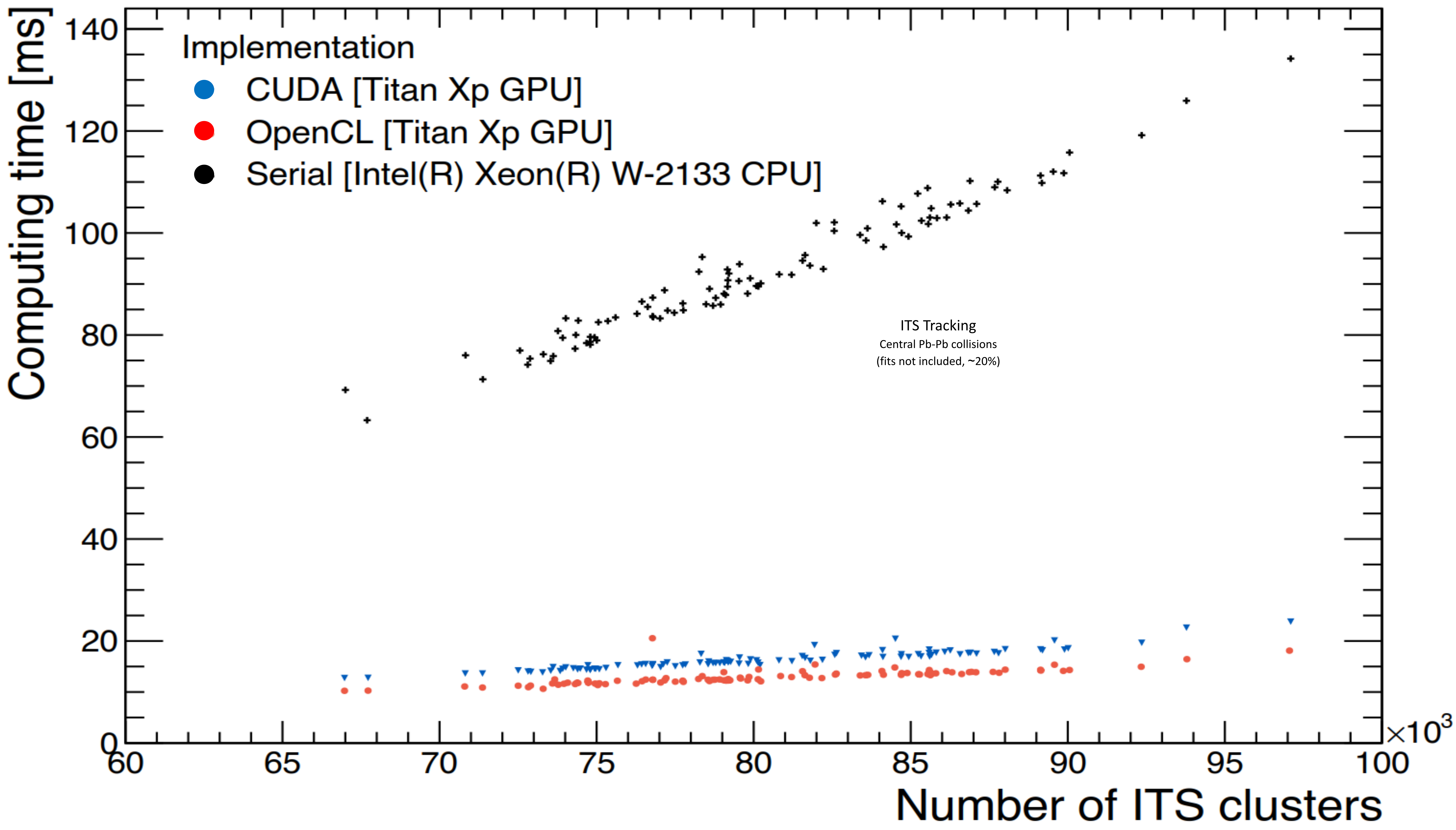
Figure: Stefano Piano, based on WLCG

Tracking time

- Speed-up normalized to single CPU core.
    - Red curve: exactly the speed-up.
    - Other curves: corrected for required CPU resources.
        - How many cores does the GPU replace.
- Significant gain with newer GPU (blue v.s. green).
- Compared to Run 2 offline, One GPU replaces **> 800 CPU cores** (**blue * red**). (**at same efficiency / resolution**)




ALICE Performance
Pb-Pb $\sqrt{s_{NN}}$ = 5.02 TeV
Number of GPUs needed for 50 kHz Pb-Pb
Number of TPC clusters used for extrapolation
0
500
1000
1500
2000
2500
0
5x10^7
1x10^8
1.5x10^8
2x10^8
2.5x10^8
3x10^8
NVIDIA RTX 3090
NVIDIA RTX 2080 Ti
NVIDIA RTX 1080 Ti
NVIDIA V100s
NVIDIA A100
AMD MI50
ALI-PERF-491405


Computing time [ms]
Implementation
CUDA [Titan Xp GPU]
OpenCL [Titan Xp GPU]
Serial [Intel(R) Xeon(R) W-2133 CPU]
ITS Tracking
Central Pb-Pb collisions
(fits not included, ~20%)
Number of ITS clusters
$\times 10^3$

EOS Total IO
60 GB/s
50 GB/s
40 GB/s
30 GB/s
20 GB/s
10 GB/s
0 B/s
06/21 00:00
06/21 08:00
06/21 16:00
06/22 00:00
06/22 08:00
06/22 16:00
06/23 00:00
06/23 08:00
06/23 16:00
06/24 00:00
06/24 08:00
06/24 16:00
06/25 00:00
avg
bytes_read 11.2 GB/s
bytes_written 20.1 GB/s