Using GPUs in O2

David Rohr
drohr@cern.ch, CERN

Offline Week

5.12.2018

Status and Goals

- Synchronous reconstruction is dominated by TPC tracking.
  - TPC tracking will run on GPUs.
  - O2 farm must be dimensioned to meet TPC tracking requirements.

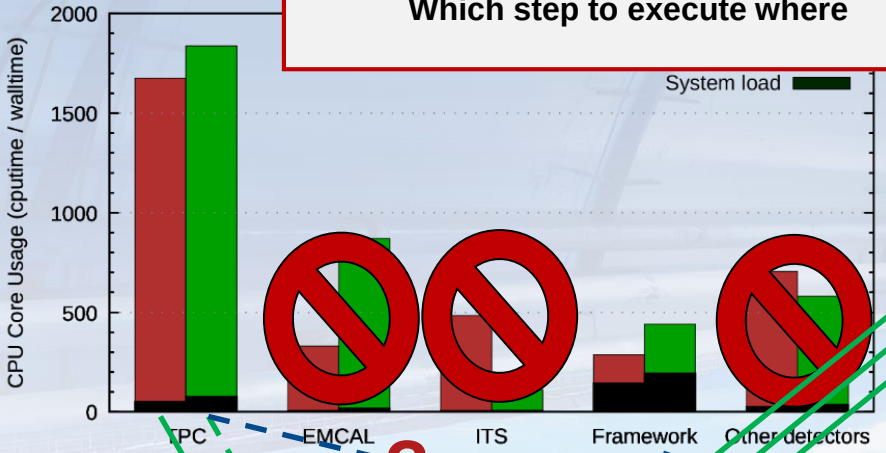
- GPUs are available during asynchronous reconstruction.
  - TPC tracking not dominant in this phase.
  - Should use the GPU for as many tasks as possible.
  - A promising target seems the full chain with ITS / TPC / TRD tracking and fit (this will be the main effort).
  - Porting reconstruction of other detectors to GPUs can be studied in parallel.

See talk in O2 plenary for motivation!

(https://indico.cern.ch/event/686145/)

- **ONLINE:**

Which step to execute where



OFFLINE:

Distribution of processing times among relevant steps

TPC Tracking & dE/dx	TPC Common Mode & Ion Tail Correction	ITS Tracking	Other Detectors
33 %	17%	33%	17%
- Track finding - Merging / Fit - dE/dx	- N[illegible] in [illegible] 3. - C[illegible] co[illegible]ted in [illegible]GA.	- Track finding - Track fit	- TRD tracking

TPC ITS Matching

Global Refit ?

V0 finding

?

?

GPU

Global fit not included here (split among detectors), and new matching step:

→ *Can both utilize GPUs*

CPU

TPC Status

Task	Status	Manpower	eta
TPC track finding	Done: running on GPUs efficiently since run 1 (ca 40x speedup). Some tuning possible for modern GPUs.	David	Done
TPC track fit	Done: running on GPUs in current O2 code (ca 35x speed up). • Should be tuned for modern GPUs. • Could use 1D B-field in first rounds. • Needs track reordering and some tricks for better warp utilization.	David	Done
TPC track merging	Work in progress: No external dependencies that prevent running on GPUs. • Most parts already parallelized on the host, suited well for GPUs. • Some serials parts remain, can run serially on GPU right now. Not efficient, but does away with transfer forth and back, needs negligible time. Can be optimized later, also as self-contained project by students. • Merging algorithm to be updated (see later).	David + small projects	May 2019
TPC dE/dx	Nearly done / Under study: Fast dE/dx component currently running in the HLT can be brought to GPUs easily. • Compatible to Run 2 offline dE/dx without calibration. • Calibration must be added and computation adapted accordingly. Need to investigate how to run on GPUs efficiently.	David / TPC experts (Jens / Marian)	End of 2018 / Q2 2019
TPC low Pt tracking / merging, flagging of loopers.	Runs on GPU already. Took more time than expected in 2018. Track finding works well, most merging cases implemented but need plenty of tuning. • Main problem is insufficient rejection ratio (independent from GPU).	David	Done / tuning in 2019
Compression, cluster removal, track model.	Straight forward task: Prototype for compression with track model exists, can be ported easily.	David	June 2019
Entropy coding.	Under study: • If it works out on GPU, can be used for other detectors as well. • No problem if not, fast CPU 3rd party libraries exist.	Michael, Matthias, David	Q3 2019
TPC cluster transformation	Nearly done / work in progress: Adapted new FastTransform class from Sergey in Tracking, will run on GPU soon. More work needed to go from a-priori transformation to just-in-time transformation for time frames.	David, Sergey	February 2019 / Q3 2019

TRD Status

Task	Status	Manpower	eta
TPC to TRD prolongation tracking.	Nearly done: running in the HLT with Run 2 data, prototype running standalone / in O2 with AliRoot data (MC and raw) and O2 TRD geometry.	Ole, David	End of 2018
Flat O2 TRD geometry for GPUs	Nearly done: flat geometry implemented, currently testing with TRD tracking. Had a problem with FPE so far, which is solved now. Need to verify that tracking works correctly.	David, Ole, Sandro	End of 2018
Regular TRD tracking in O2	Work in progress: Depends on other O2 TRD work, mostly on the simulation. Tracking interface class will be ready soon.	TRD	Some time 2019
Flat material lookup class	Work in progress: Currently, tracking does not take into account material. Need flat material lookup class suited for GPUs to do so. Discussed with Ruben how to implement it, but will take some time until available.	Ruben, David	Q1 2019
Prolongation / Track fit with material	Work in progress: Needs the material look-up class, then more or less straight forward.	Sergey, David	Q2 2019

ITS Status

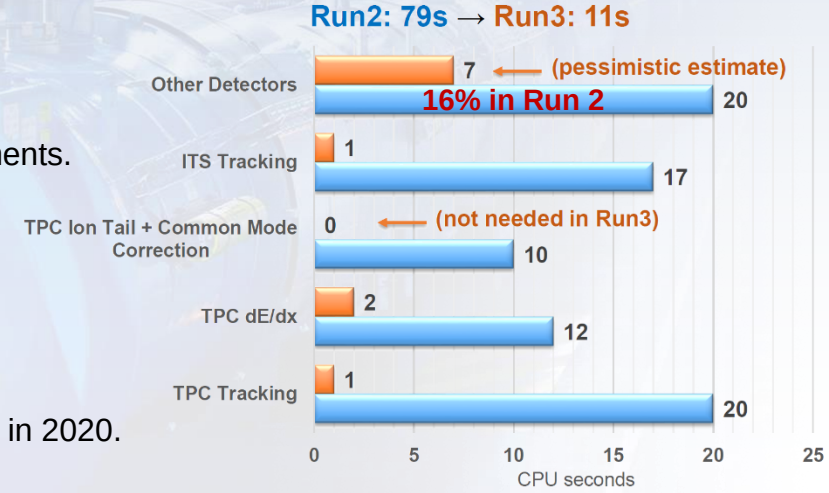
Task	Status	Manpower	eta
Track finding (simple)	Done	Max	Done
Vertexing	Work in progress: Is needed as input for the tracking	Matteo	Q2 2019
Track finding (advanced)	Under study: • Secondary particles • Missing layers • Low momentum • Hit rejection (can be done in the fit)	Max, Matteo, David	2019
GPU Optimizations	Ongoing: Current speedup factor 7x v.s. fast CPU core. • Already >10x for HyperThreaded core / usual server core. • Implemented one bottleneck where step runs twice to estimate memory needed, currently being improved. • More optimization needed	Max, Matteo, David	Ongoing in 2019
Track Fit	Work in progress: This can basically take what is done for the TRD (including the material look-up and propagation), so it is delayed until that is ready.	David, Ruben, Sergey, Max, Matteo	Q2 2019

Global Tracking Status

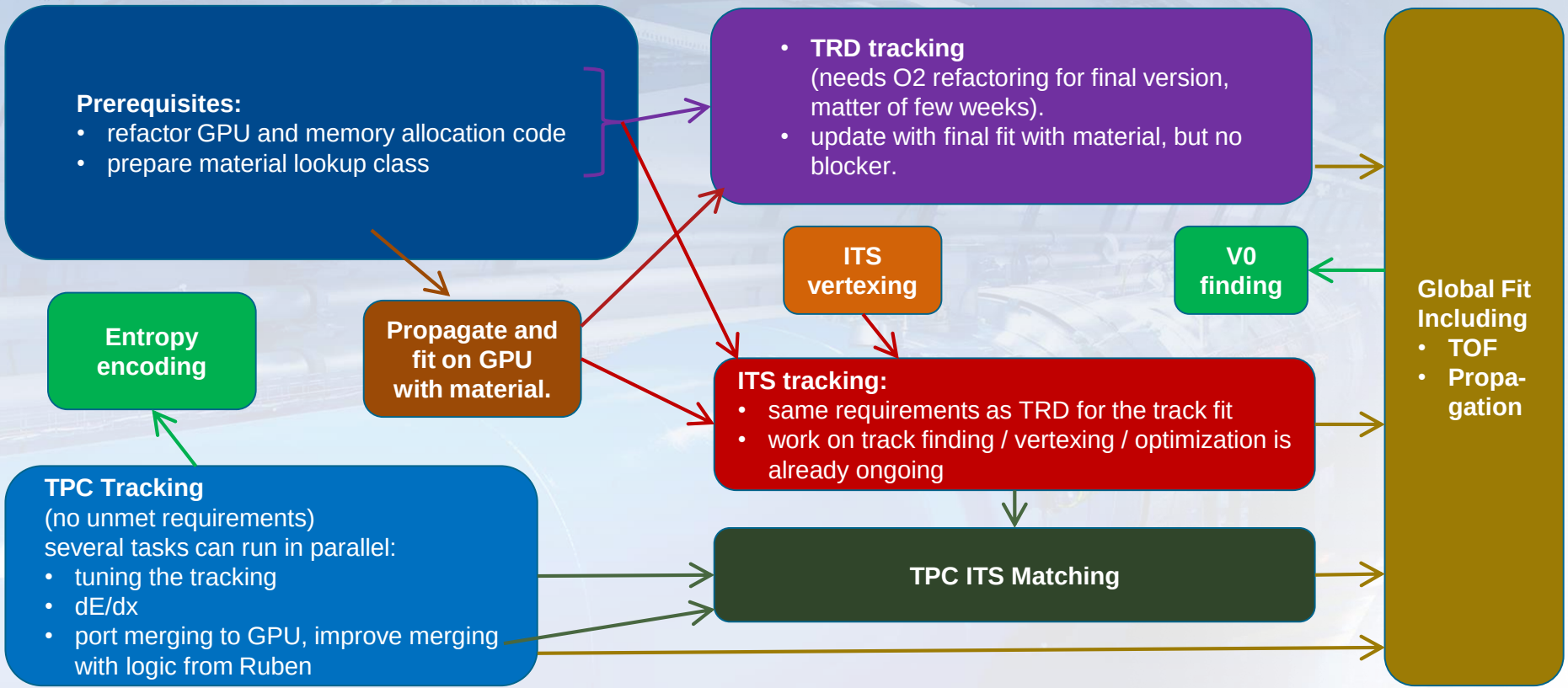
Task	Status	Manpower	eta
TPC to ITS matching	Work in progress: Available on CPU by Ruben. • Superior logic compared to TPC track merging. Will adapt the better logic for TPC merging and GPU,. • Use the same code for TPC-TPC and TPC-ITS merging. Not high priority, as CPU version is running OK.	David, Ruben	September 2019
Cluster rejection by interpolation from two-way fit.	Work in progress: Needed for ITS tracking, could improve TPC tracking / search for high-incl. segment of loopers.	David	September 2019
Prolongation to TOF	Not yet started: CPU code exists, to be studied how to perform in on GPU. • Not so CPU intense, but parallel propagation well suited for GPU, and could be used for global refit. • In principle, many things are there: can reuse most developments (Propagation + Material from TRD prolongation).	Chiara, David, Francesco	Unclear
Global Refit	Straight forward task: Kalman filter will run efficiently on GPU, basically everything is there already. • Can discuss whether to use the HLT GPU track fit (linearizing a model of the track) or the O2 fit (linearizing the current track parameters). Probably attempt both and compare. • Not high priority, only feasible after TPC to ITS matching.	David	End of 2019
V0 finder	Not yet started: Promising candidate for GPU processing (parallel by design). • Needs global refit as requirement.		End of 2019
Kink finder	Not yet started: Should run in the TPC tracking, ideally with the better cluster rejection code.	David	End of 2019
Propagation to outer layers	Not yet started / straight forward: Even though other detectors are not processed on the GPU, we can use all the tracking machinery for free for all necessary propagation. (Has the same requirements as the propagation through TRD with material.)	David	End of 2019

Other detectors

- Other detectors are mostly independent (from reconstruction side).
  - The tracking ITS + TPC + TRD (+ TOF) is kind of one entity with many dependencies.
  - Other detectors need the tracks, but there are no other dependencies: no need to share data besides tracks.
  - Could principally run on the GPU after the tracking, but does not affect the tracking.
- Other detectors represent only 16% of total CPU load (estimated from Run 2 – see last O2 plenary talk).
  - We should focus on the tracking on GPU.
  - Good fraction of the 16% might already be handled by GPU tracking (TRD, V0, propagation to other detectors).
  - CPUs should be capable of handling the other detectors (<10%).
  - Development for other detectors can happen in parallel.
    - EMCAL is investigating this.
    - PHOS lacks manpower, ould profit from EMCAL GPU developments.
    - MUON tracking good candidate, lacks manpower as well.
  - In my opinion:
    - No sense in moving manpower from barrel tracking to others
    - Use whatever GPU algorithm developed in parallel.
    - Otherwise we use the CPU version.
    - If the barrel tracking is working well, we can dedicate manpower in 2020.



Roadmap


Prerequisites:
• refactor GPU and memory allocation code
• prepare material lookup class
• TRD tracking
(needs O2 refactoring for final version, matter of few weeks).
• update with final fit with material, but no blocker.
Global Fit Including
• TOF
• Propa-gation
ITS vertexing
V0 finding
Entropy encoding
Propagate and fit on GPU with material.
ITS tracking:
• same requirements as TRD for the track fit
• work on track finding / vertexing / optimization is already ongoing
TPC Tracking
(no unmet requirements)
several tasks can run in parallel:
• tuning the tracking
• dE/dx
• port merging to GPU, improve merging with logic from Ruben
TPC ITS Matching

Unclear / critical steps

- **Entropy encoding:**
  - Not clear if feasible on GPU.
  - Implementation not yet started.
  - At the end of compression chain, no blocker for any other step.
- ITS tracking:
  - In principle on track, but we do not yet have the full tracking for secondaries / tracks with missing layers, etc.
  - Would be important to evaluate the full physics performance soon.
  - However, this problem is not related to the GPU or CPU version.
  - General performance OK, GPU speedup should be improved.
- V0 finding:
  - Promising candidate for GPU offload, but not investigated yet. More or less independent of the rest.
- TPC:
  - Cluster removal does not yet achieve the desired compression rate (unrelated to GPU / CPU version).
  - Need to test tracking of timeframes (and track merging) with distortions.

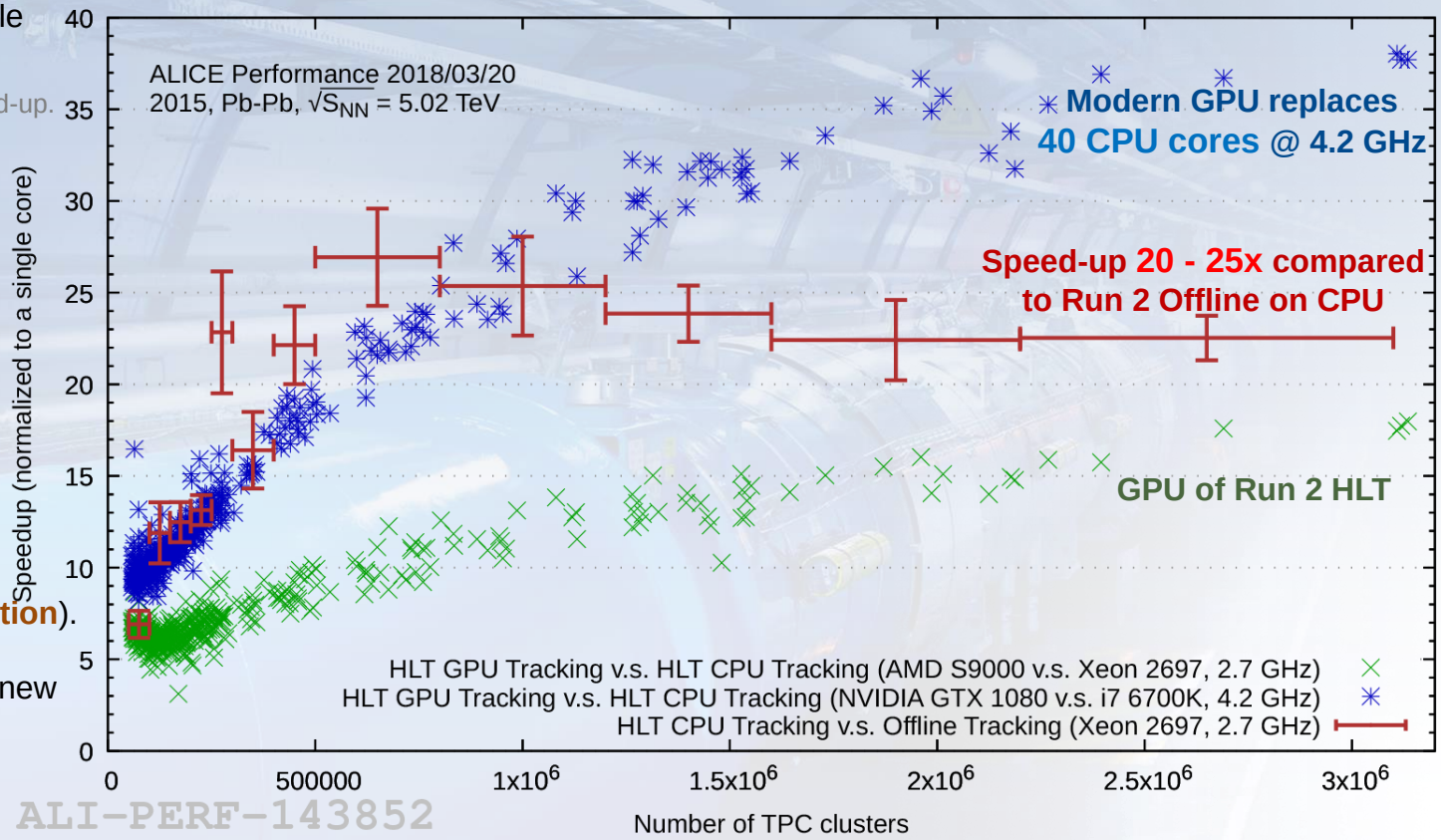
TPC Tracking performance

- 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 TPC tracking, one GPU replaces **> 800 CPU cores** (**blue * red**). (**at same efficiency / resolution**).

- We see ~30% speedup with new GPU generation (RTX 2080 v.s. GTX 1080)



TPC Tracking performance

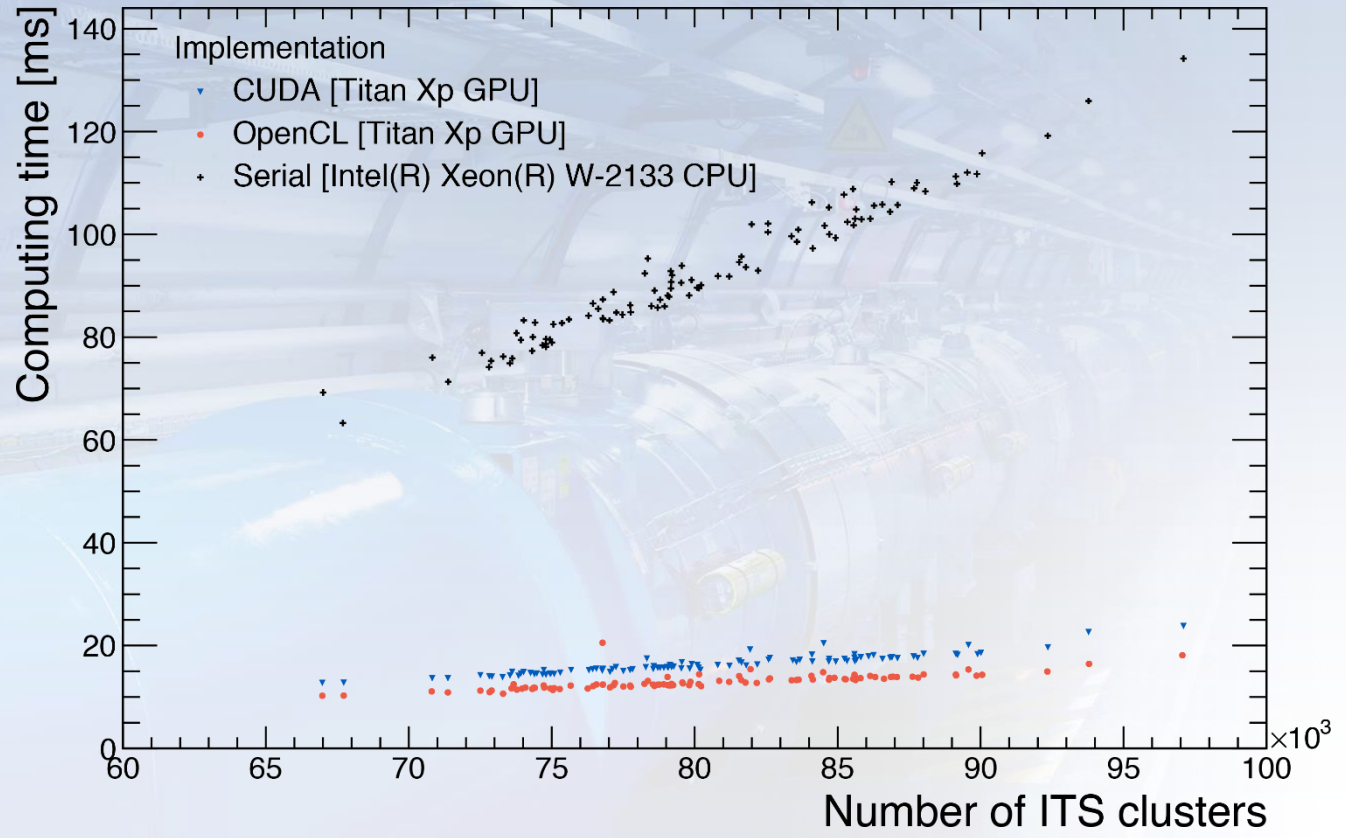
- Overlapping events in TPC with realistic bunch structure @ 50 kHz Pb-Pb.
- Timeframe of 2 ms shown (will be 10 – 20 ms in production).
- Tracks of different collisions shown in different color.

Processing of **23 ms** time frame needs **~20 seconds** on one GPU.

- Does not yet include all steps (merging, compression).
- In the end we will need around 1500 GPUs (currently state of the art) for TPC tracking at synchronous stage.

- Other detectors will add only a small contribution to GPU load during synchronous stage.

ITS Tracking performance

- GPU performance versus a single CPU core.
- Yet, GPU speedup of only factor 7.
- Currently implementing some improvements to GPU version.
- Total processing time in the order of 10 ms / central Pb-Pb.
- Full tracking with secondaries and missing layers will be slower.
- Does not yet include track fit.

→ Need to improve GPU performance, but total time in same order of magnitude as TPC tracking.