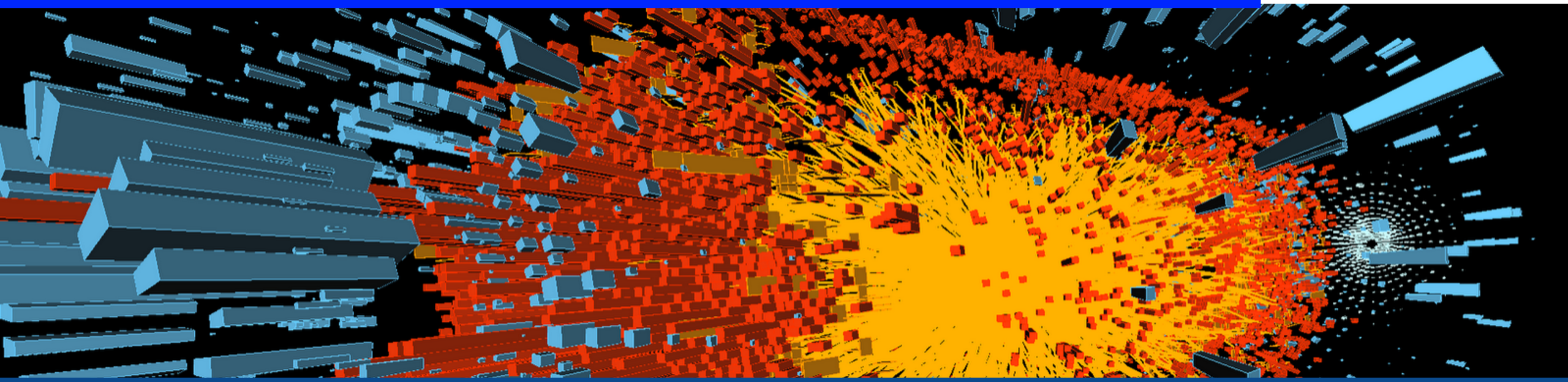
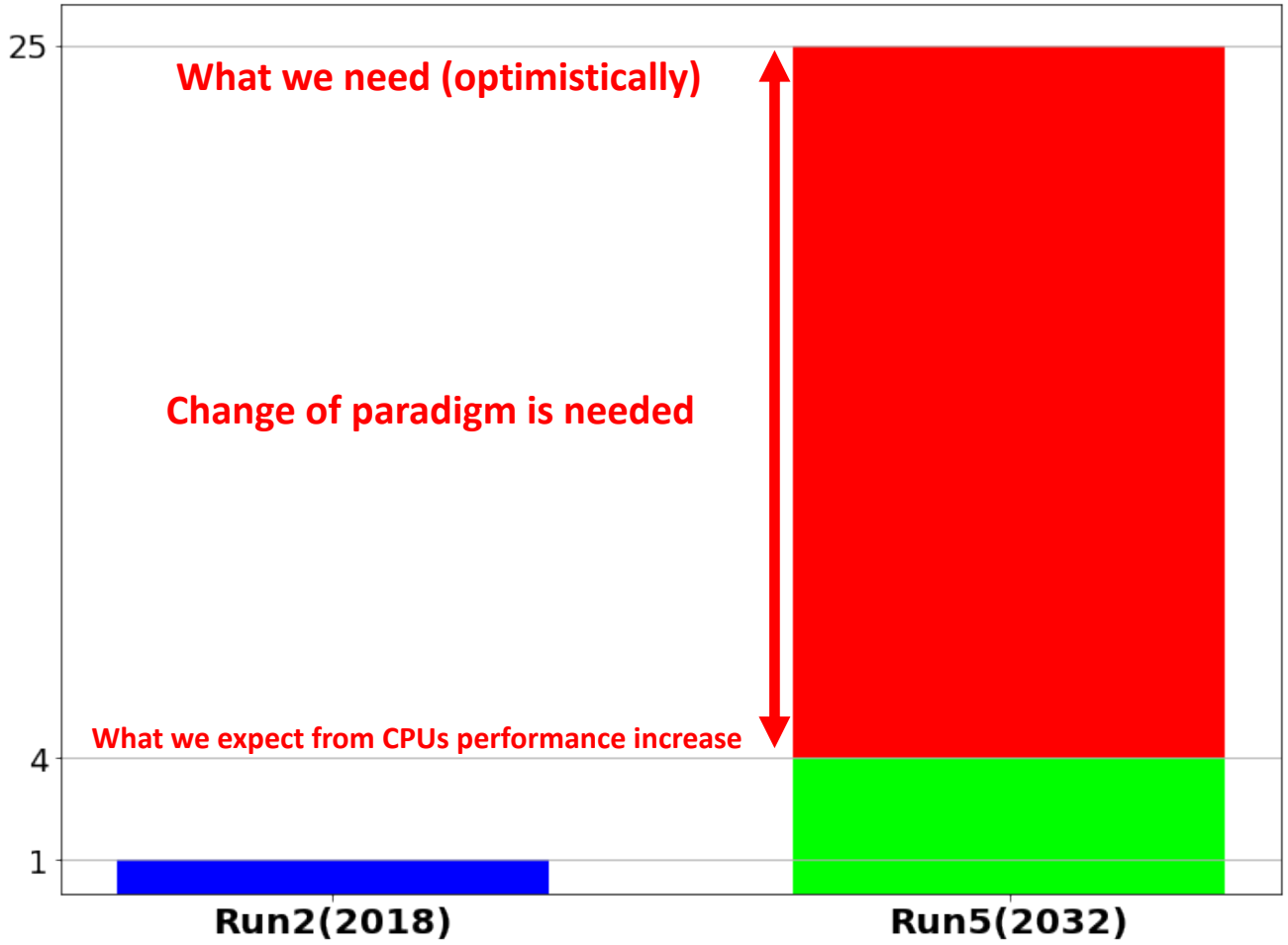
Patatrack : an Heterogenous Validation



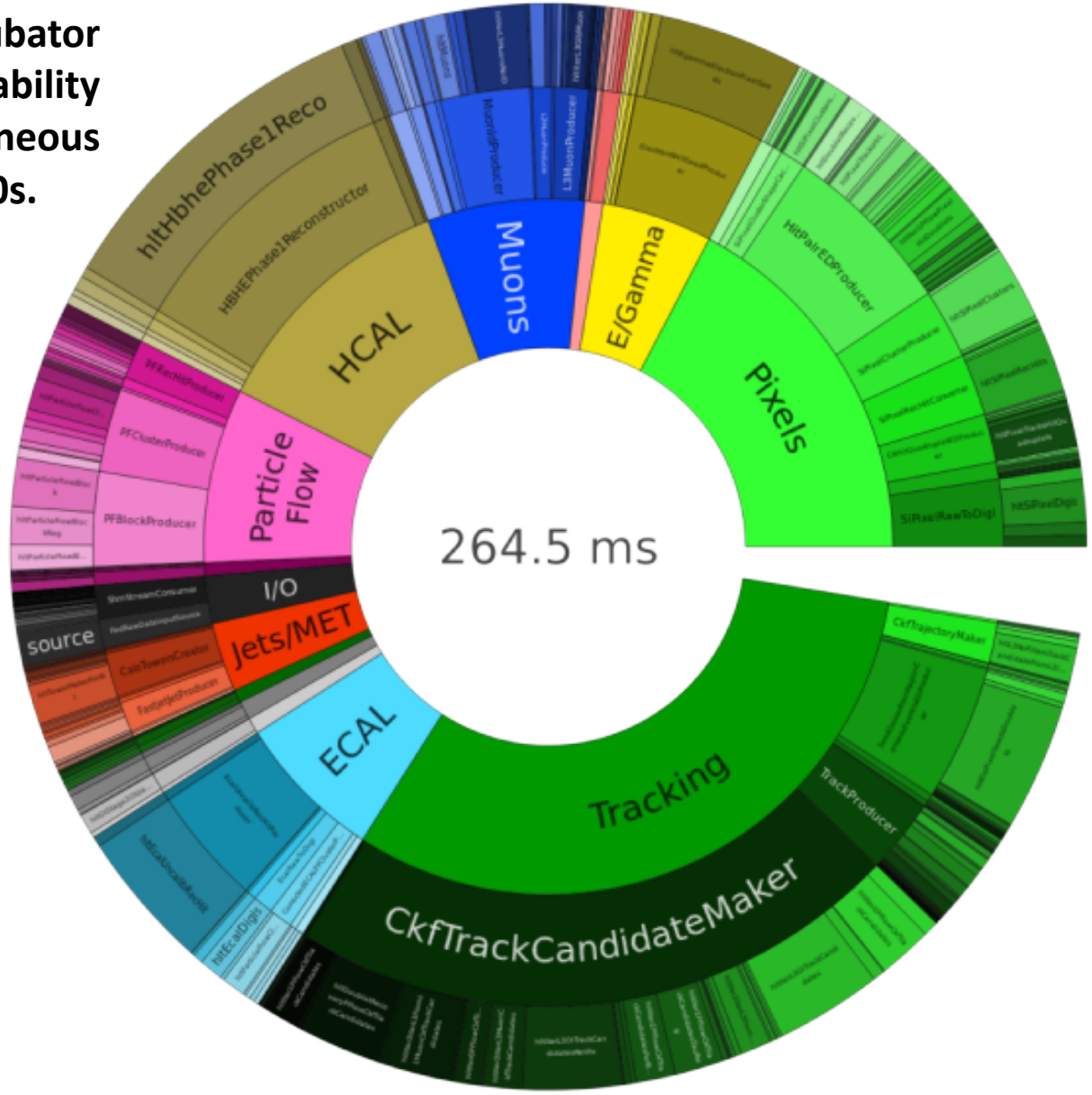
Patatrack – Why?

With High Luminosity LHC, combining the increased instantaneous luminosity, increased Level 1 Trigger output (x7.5) and the increased detector complexity, CMS High Level Trigger would need x25 new resources while, optimistically x4 is what we will get.



Patatrack – What?

Patatrack is a software R&D incubator born in **2016** to study the feasability of the online/offline heterogeneous reconstruction starting from 2020s.



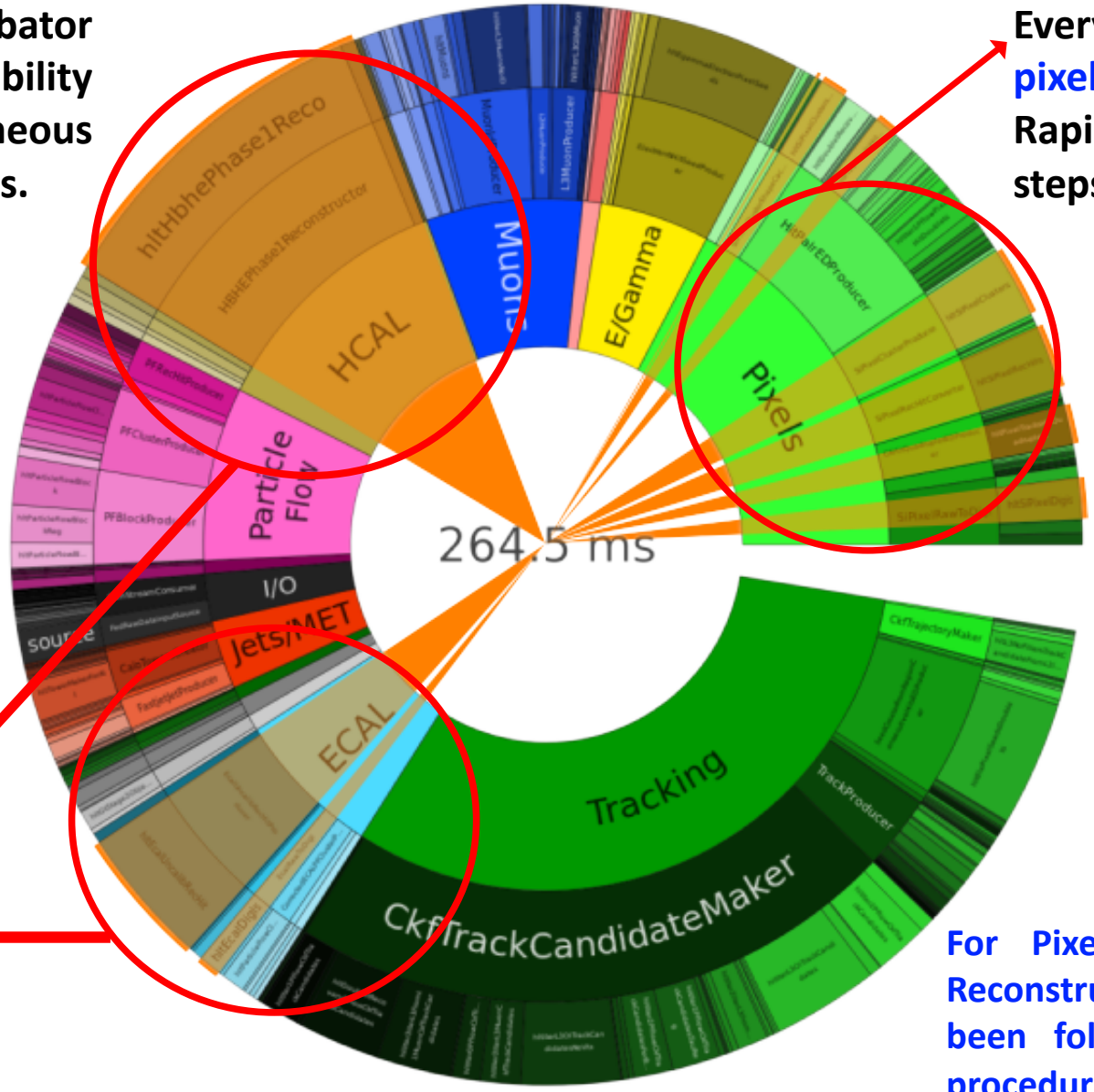
Circles

2018 Data

- **<PU> = 50**
- **2018 L1T & HLT**

Patatrack – What?

Patatrack is a software R&D incubator born in **2016** to study the feasability of the online/offline heterogeneous reconstruction starting from 2020s.



Everything started with global **pixel global track reconstruction.** Rapidly **expanding** to further steps and detectors.

2018 Data

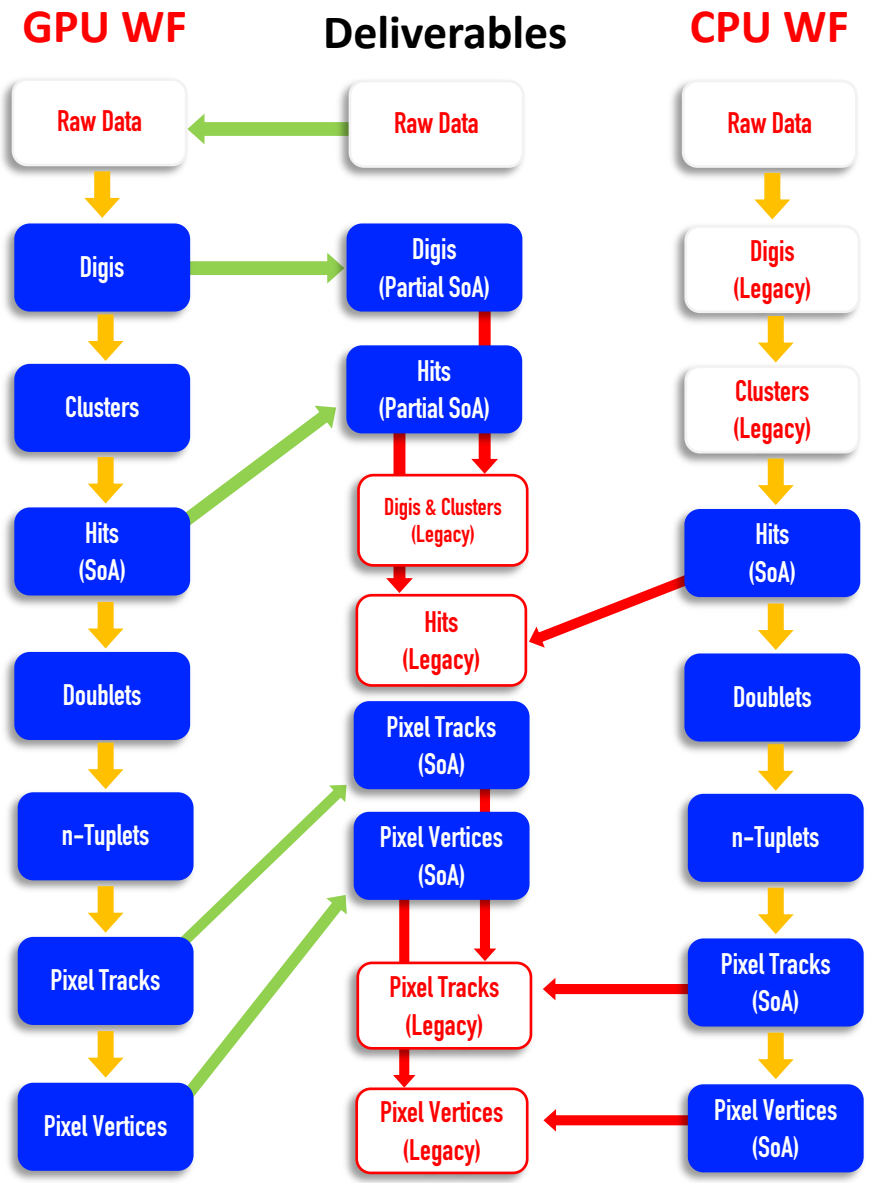
- **<PU> = 50**
- **2018 L1T & HLT**

ECAL & HCAL: local reconstruction & calibration (based on similar algorithms)

Circles

For Pixel Tracks & Calorimeter Local Reconstruction different approaches have been followed and different validation procedures have been developed

Patatrack Pixel Tracks – The Workflow

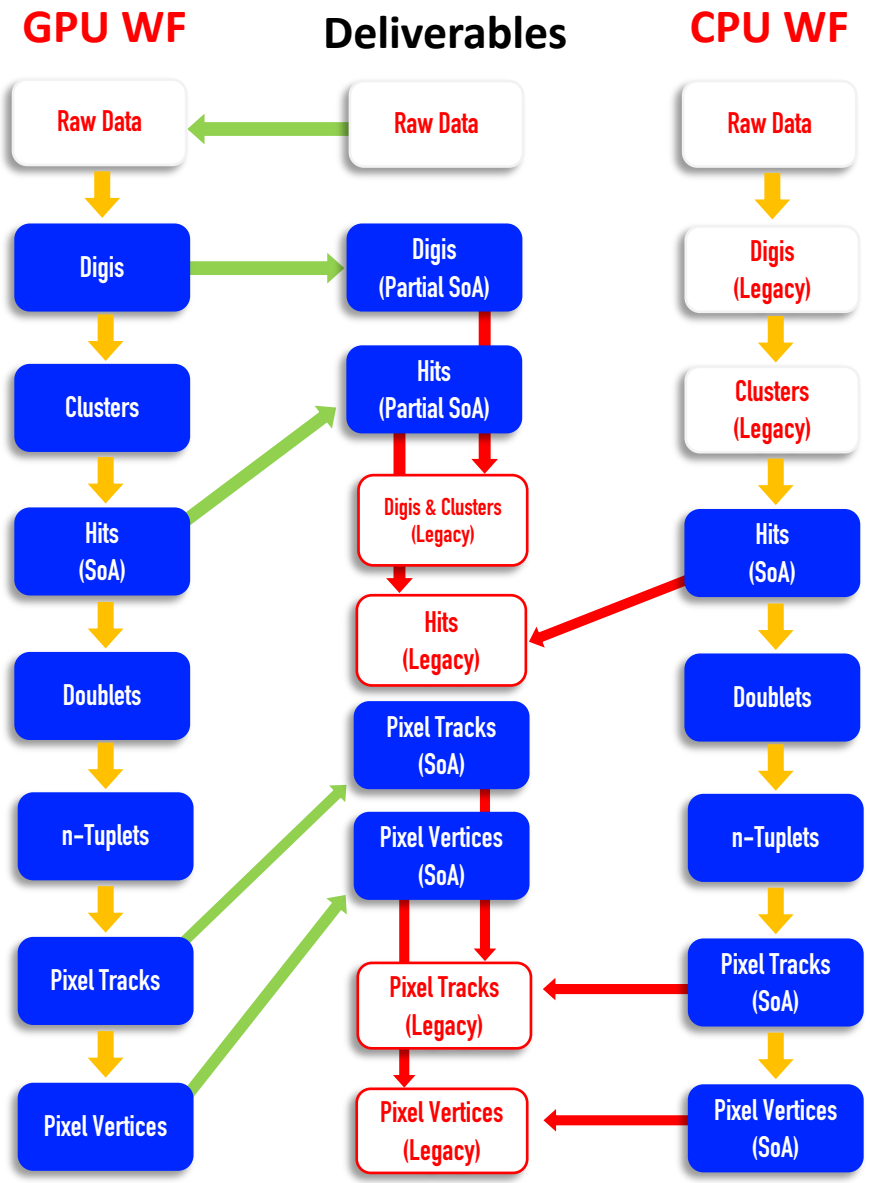


The **goal:** reconstruct pixel-based tracks and vertices on the GPU minimising data transfer.

The **workflow**,running both on **CPU** and **GPU:**

- **copy** the raw data to the GPU
- run multiple kernels to perform the **various steps**
  - decode the raw data
  - cluster the pixel hits
  - form hit doublets
  - form hit quadruplets (or ntuplets)
  - clean up duplicates
  - fitting & vertexing
- **Copy back the final results to the host, optimised SoA format, and eventually convert to legacy event data**

Patatrack Pixel Track – The Validation

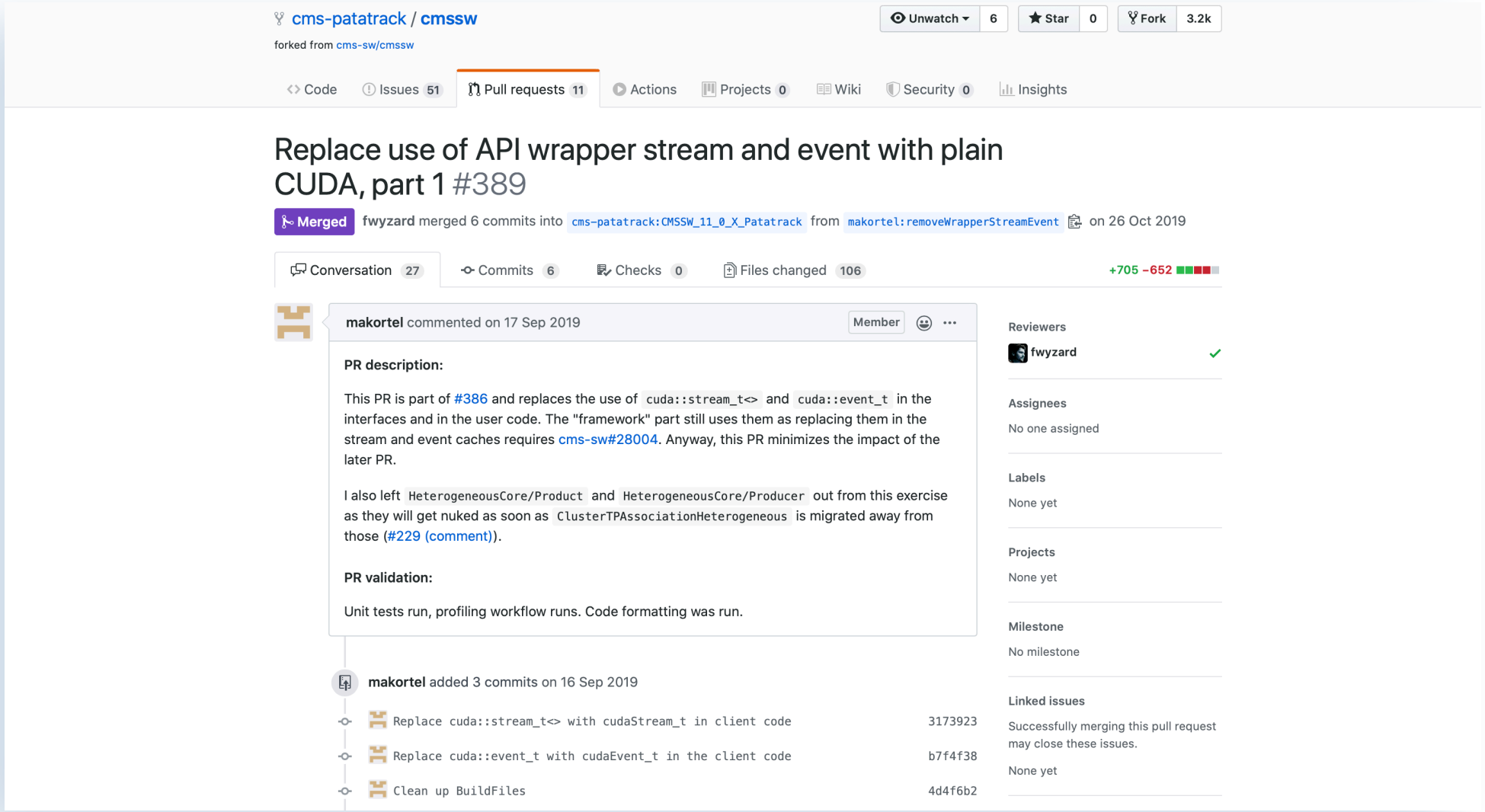


The **validation:**

- compares results at **dataset** level: physics performance metrics of the delivered collections
- is done in **three stages:**
  - For **each** Pull Request
  - CMSSW Legacy **reference vs** Patatrack **reference**
  - **CPU** and **GPU** Patatrack Workflows
- is **based** on the already existing tracking validation there in the CMS reconstruction software that has been **adapted to monitor the new collections**. It relies on the DQM tools, ROOT formats.
- is **fully automatised** and **structured** (see here)
- includes computational performance plots (**throughput**)

Patatrack Pixel Track – The Validation – PRs

Let's take an example **PR**



cms-patatrack / cmssw

Unwatch 6 | Star 0 | Fork 3.2k

forked from cms-sw/cmssw

Code | Issues 51 | Pull requests 11 | Actions | Projects 0 | Wiki | Security 0 | Insights

Replace use of API wrapper stream and event with plain CUDA, part 1 #389

Merged fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:removeWrapperStreamEvent` on 26 Oct 2019

Conversation 27 | Commits 6 | Checks 0 | Files changed 106 +705 −652

makortel commented on 17 Sep 2019 — Member

PR description:

This PR is part of #386 and replaces the use of `cuda::stream_t<>` and `cuda::event_t` in the interfaces and in the user code. The "framework" part still uses them as replacing them in the stream and event caches requires cms-sw#28004. Anyway, this PR minimizes the impact of the later PR.

I also left `HeterogeneousCore/Product` and `HeterogeneousCore/Producer` out from this exercise as they will get nuked as soon as `ClusterTPAssociationHeterogeneous` is migrated away from those (#229 (comment)).

PR validation:

Unit tests run, profiling workflow runs. Code formatting was run.

makortel added 3 commits on 16 Sep 2019

- Replace cuda::stream_t<> with cudaStream_t in client code 3173923
- Replace cuda::event_t with cudaEvent_t in the client code b7f4f38
- Clean up BuildFiles 4d4f6b2

Reviewers: fwyzard ✓

Assignees: No one assigned

Labels: None yet

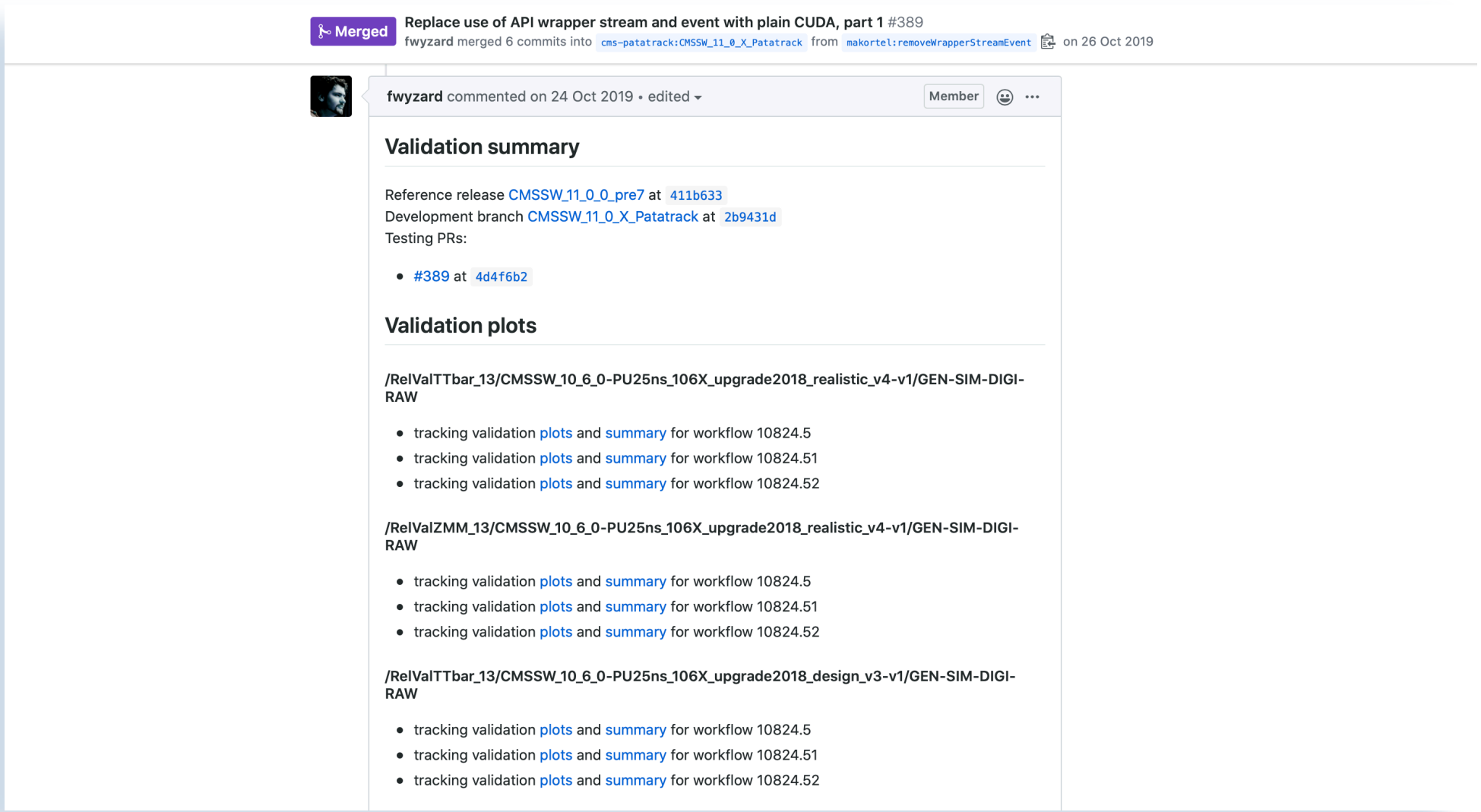
Projects: None yet

Milestone: No milestone

Linked issues: Successfully merging this pull request may close these issues. None yet

Patatrack Pixel Track – The Validation - PRs

All the procedure happens through a **single (long) validate** script (by A.Bocci™) allowing to compare (multiple) **PRs** or **local** developments



Merged **Replace use of API wrapper stream and event with plain CUDA, part 1** #389
fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:removeWrapperStreamEvent` on 26 Oct 2019

fwyzard commented on 24 Oct 2019 • edited — Member

Validation summary

Reference release CMSSW_11_0_0_pre7 at `411b633`
Development branch CMSSW_11_0_X_Patatrack at `2b9431d`
Testing PRs:

- #389 at `4d4f6b2`

Validation plots

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

/RelValZMM_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

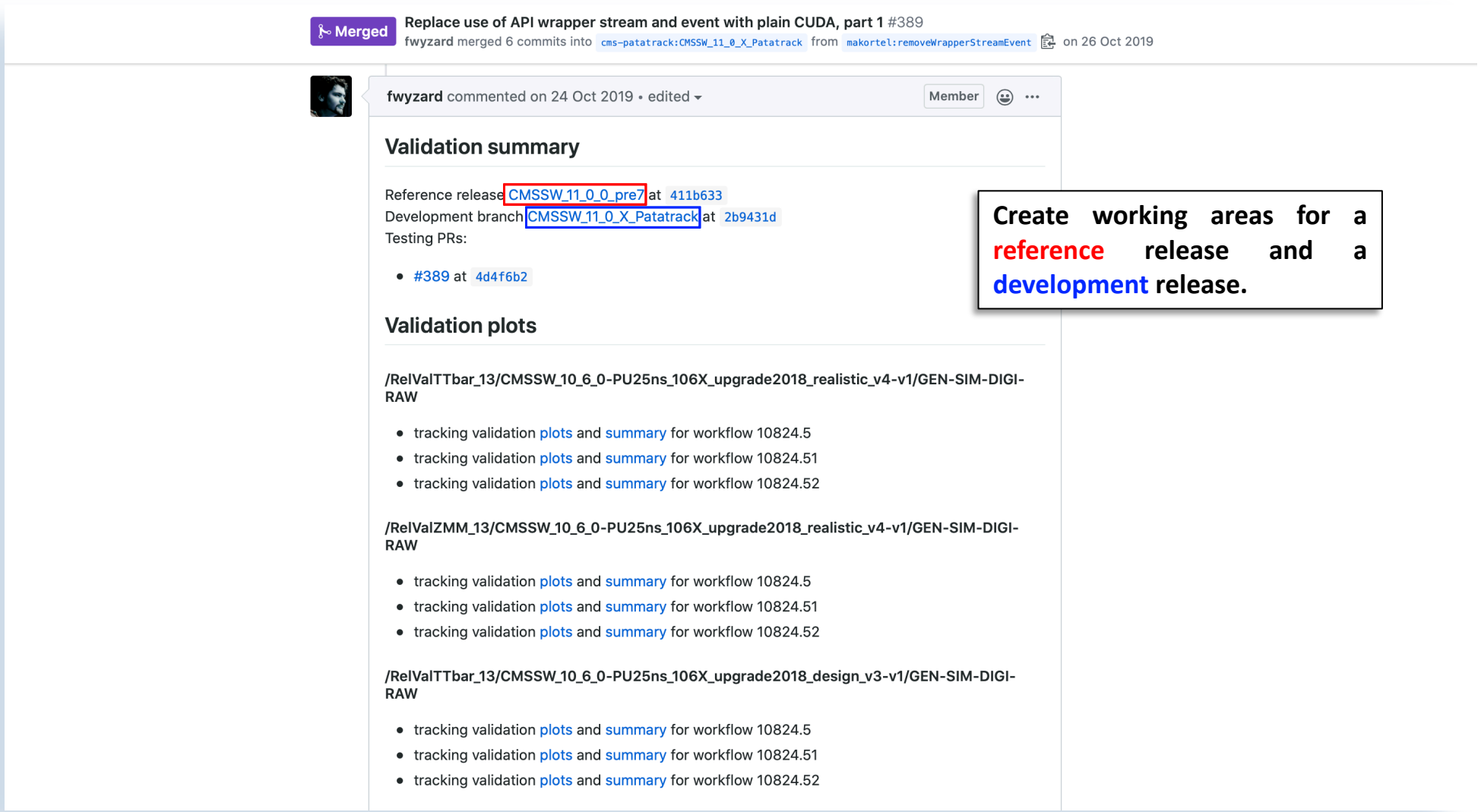
- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_design_v3-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

Patatrack Pixel Track – The Validation - PRs

All the procedure happens through a **single (long) validate** script (by A.Bocci™) allowing to compare (multiple) **PRs** or **local** developments



Merged **Replace use of API wrapper stream and event with plain CUDA, part 1** #389
fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:removeWrapperStreamEvent` on 26 Oct 2019

fwyzard commented on 24 Oct 2019 • edited — Member

Validation summary

Reference release CMSSW_11_0_0_pre7 at `411b633`
Development branch CMSSW_11_0_X_Patatrack at `2b9431d`
Testing PRs:

- #389 at `4d4f6b2`

Validation plots

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

/RelValZMM_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

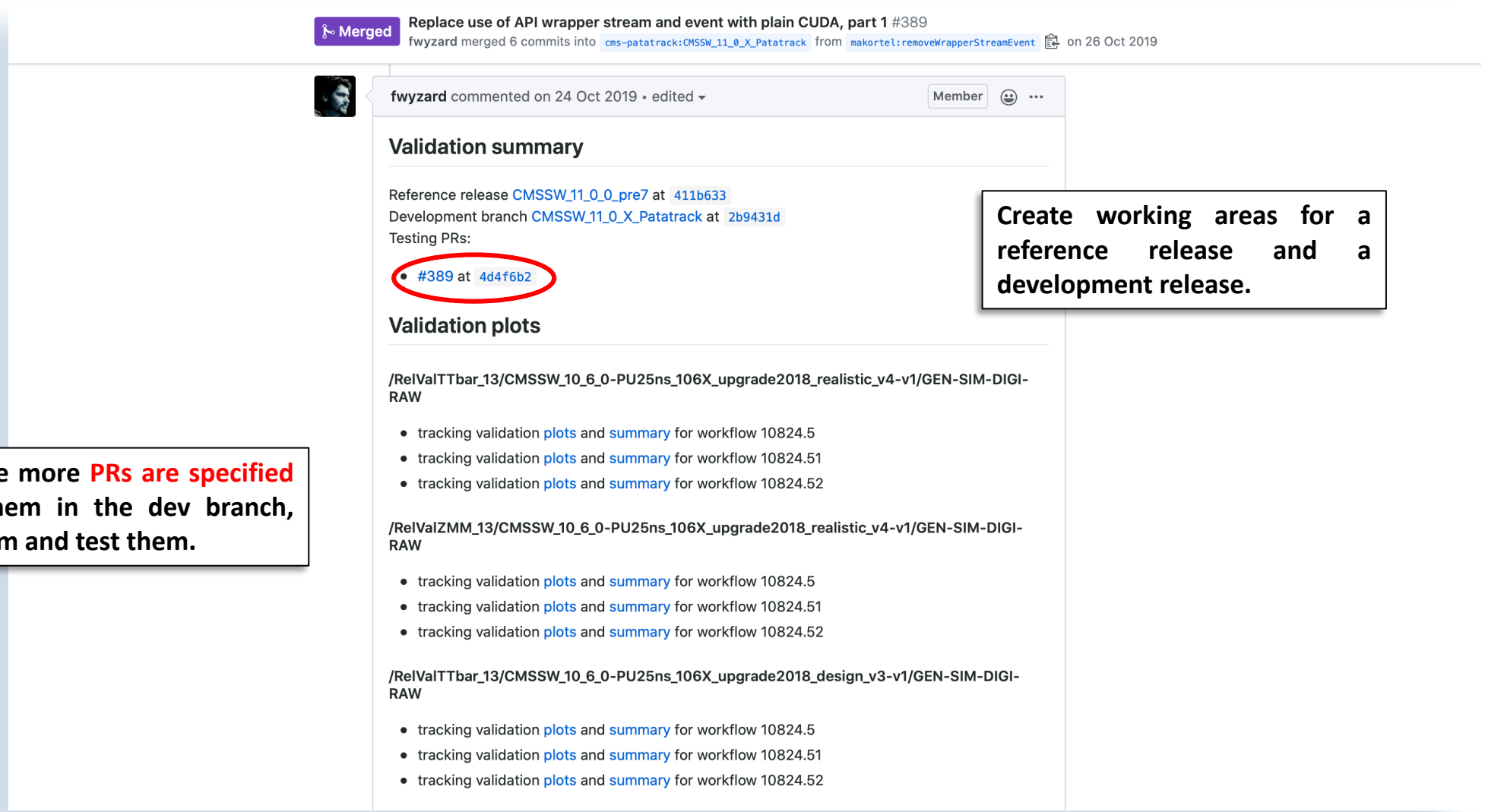
/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_design_v3-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

Create working areas for a reference release and a development release.

Patatrack Pixel Track – The Validation - PRs

All the procedure happens through a **single (long) validate** script (by A.Bocci™) allowing to compare (multiple) **PRs** or **local** developments



Merged **Replace use of API wrapper stream and event with plain CUDA, part 1** #389

fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:removeWrapperStreamEvent` on 26 Oct 2019

fwyzard commented on 24 Oct 2019 • edited — Member

Validation summary

Reference release CMSSW_11_0_0_pre7 at `411b633`
Development branch CMSSW_11_0_X_Patatrack at `2b9431d`
Testing PRs:

- #389 at `4d4f6b2`

Validation plots

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

/RelValZMM_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_design_v3-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

Create working areas for a reference release and a development release.

If one ore more PRs are specified merge them in the dev branch, build them and test them.

Patatrack Pixel Track – The Validation - PRs

All the procedure happens through a **single (long) validate** script (by A.Bocci™) allowing to compare (multiple) **PRs** or **local** developments

Merged **Replace use of API wrapper stream and event with plain CUDA, part 1** #389
fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:removeWrapperStreamEvent` on 26 Oct 2019

fwyzard commented on 24 Oct 2019 • edited — Member

Validation summary

Reference release CMSSW_11_0_0_pre7 at `411b633`
Development branch CMSSW_11_0_X_Patatrack at `2b9431d`
Testing PRs:

- #389 at `4d4f6b2`

Validation plots

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5 → Legacy pixel-tracks reconstruction
- tracking validation plots and summary for workflow 10824.51 → Pixel-tracks (with Riemann fit) on the GPU
- tracking validation plots and summary for workflow 10824.52 → Pixel-tracks (with Riemann fit) on the CPU

/RelValZMM_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_design_v3-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

Create working areas for a reference release and a development release.

If one ore more PRs are specified merge them in the dev branch, build them and test them.

Run various workflows (legacy, GPU, CPU) on RelVal samples producing physics performance plots and summary tables uploading them to a comparisons are uploaded to the user web area making them easily browsable.

Patatrack Pixel Track – The Validation - PRs

All the procedure happens through a **single (long) validate** script (by A.Bocci™) allowing to compare (multiple) **PRs** or **local** developments

Merged **Replace use of API wrapper stream and event with plain CUDA, part 1** #389
fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:removeWrapperStreamEvent` on 26 Oct 2019

fwyzard commented on 24 Oct 2019 • edited — Member

Validation summary

Reference release CMSSW_11_0_0_pre7 at `411b633`
Development branch CMSSW_11_0_X_Patatrack at `2b9431d`
Testing PRs:

- #389 at `4d4f6b2`

Validation plots

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5 → **Legacy pixel-tracks reconstruction**
- tracking validation plots and summary for workflow 10824.51 → **Pixel-tracks (with Riemann fit) on the GPU**
- tracking validation plots and summary for workflow 10824.52 → **Pixel-tracks (with Riemann fit) on the CPU**

/RelValZMM_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_design_v3-v1/GEN-SIM-DIGI-RAW

- tracking validation plots and summary for workflow 10824.5
- tracking validation plots and summary for workflow 10824.51
- tracking validation plots and summary for workflow 10824.52

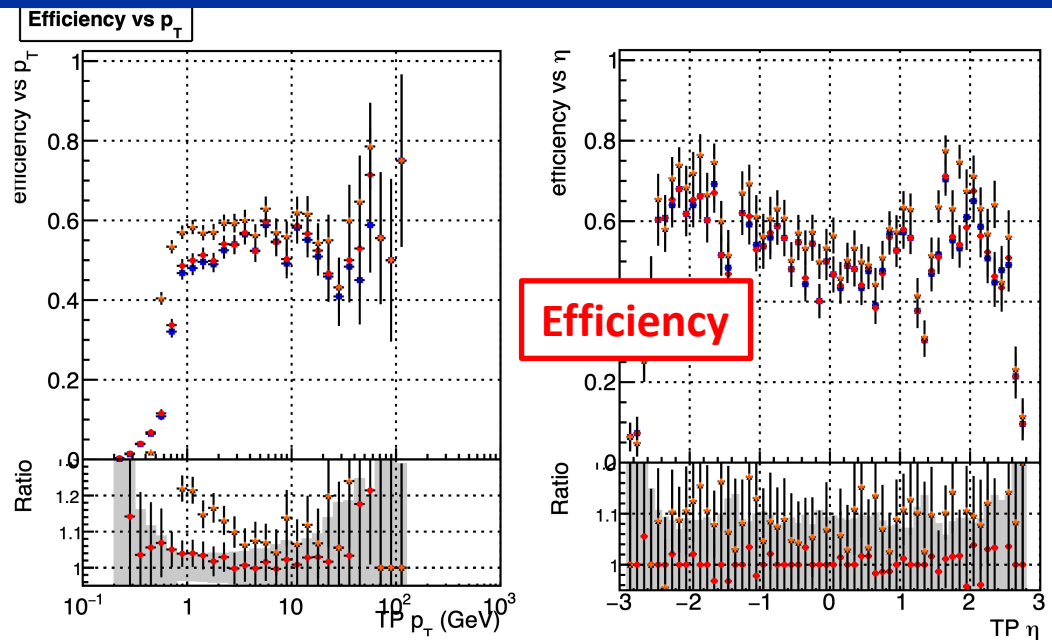
Create working areas for a reference release and a development release.

If one ore more PRs are specified merge them in the dev branch, build them and test them.

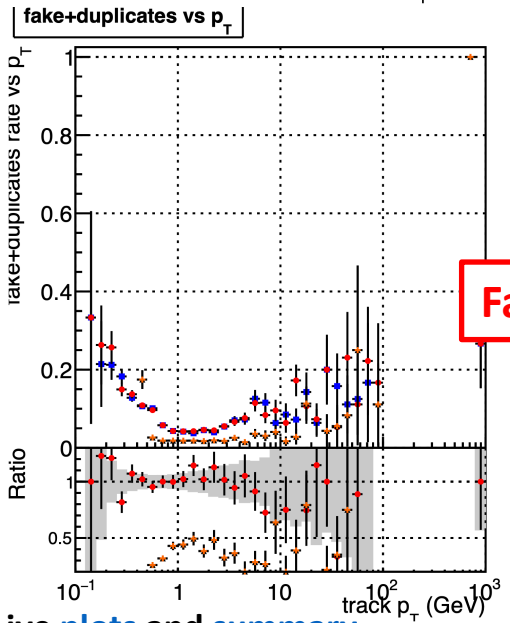
Testing on different event topologies and detector conditions to avoid biases (not Patatrack specifics – common in CMS)

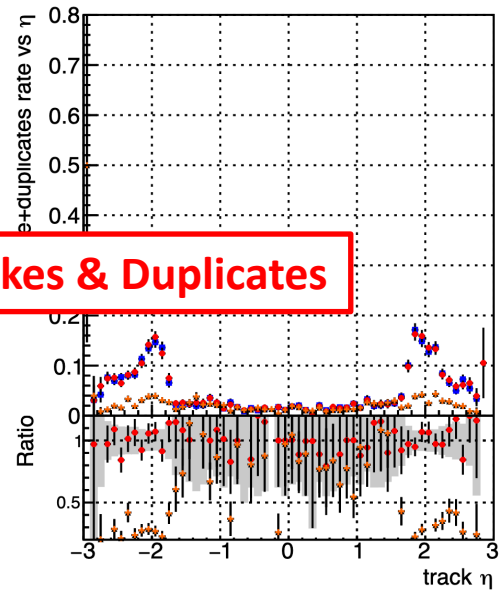
Run various workflows (legacy, CPU, GPU) on RelVal samples producing physics performance plots and summary tables uploading them to a comparisons are uploaded to the user web area making them easily browsable.

Patatrack Pixel Track Validation – Physics Peformance



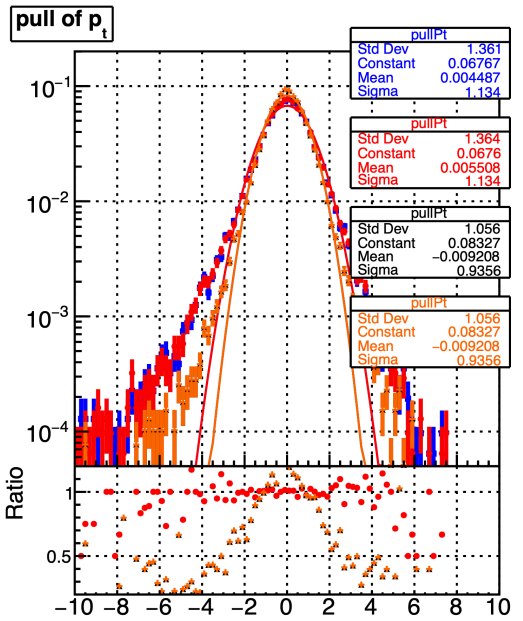
Efficiency

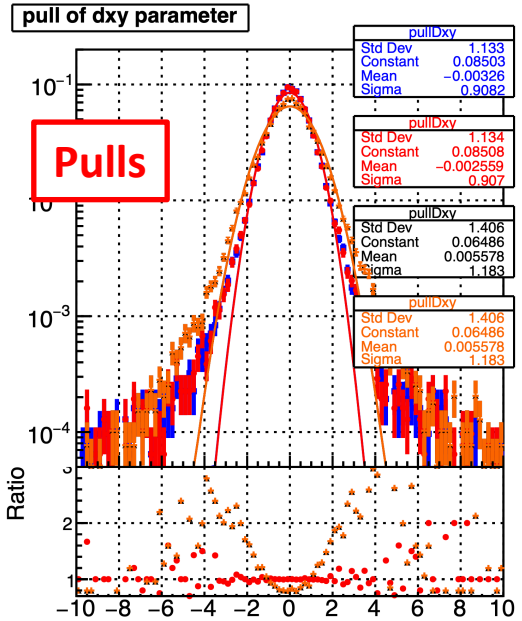




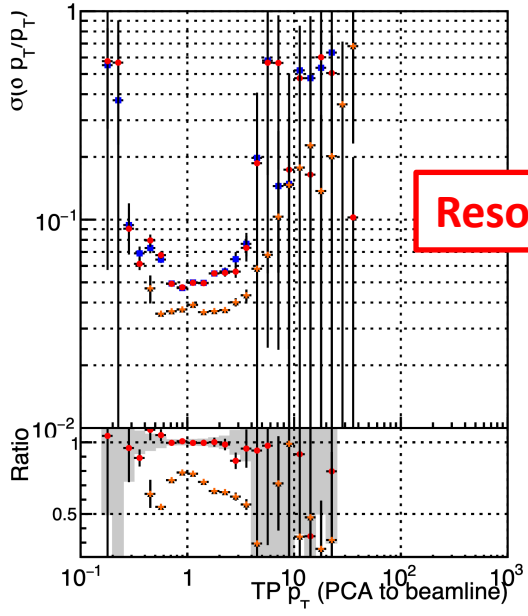
Fakes & Duplicates

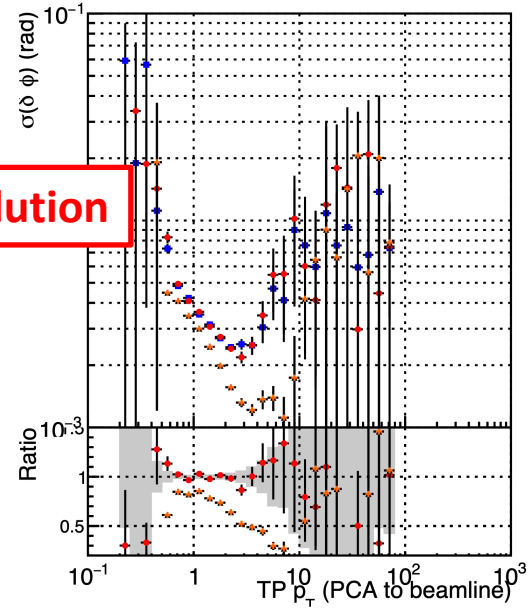
- Legacy
- Patatrack Ref
- Patatrack Dev
- PR Testing





Pulls

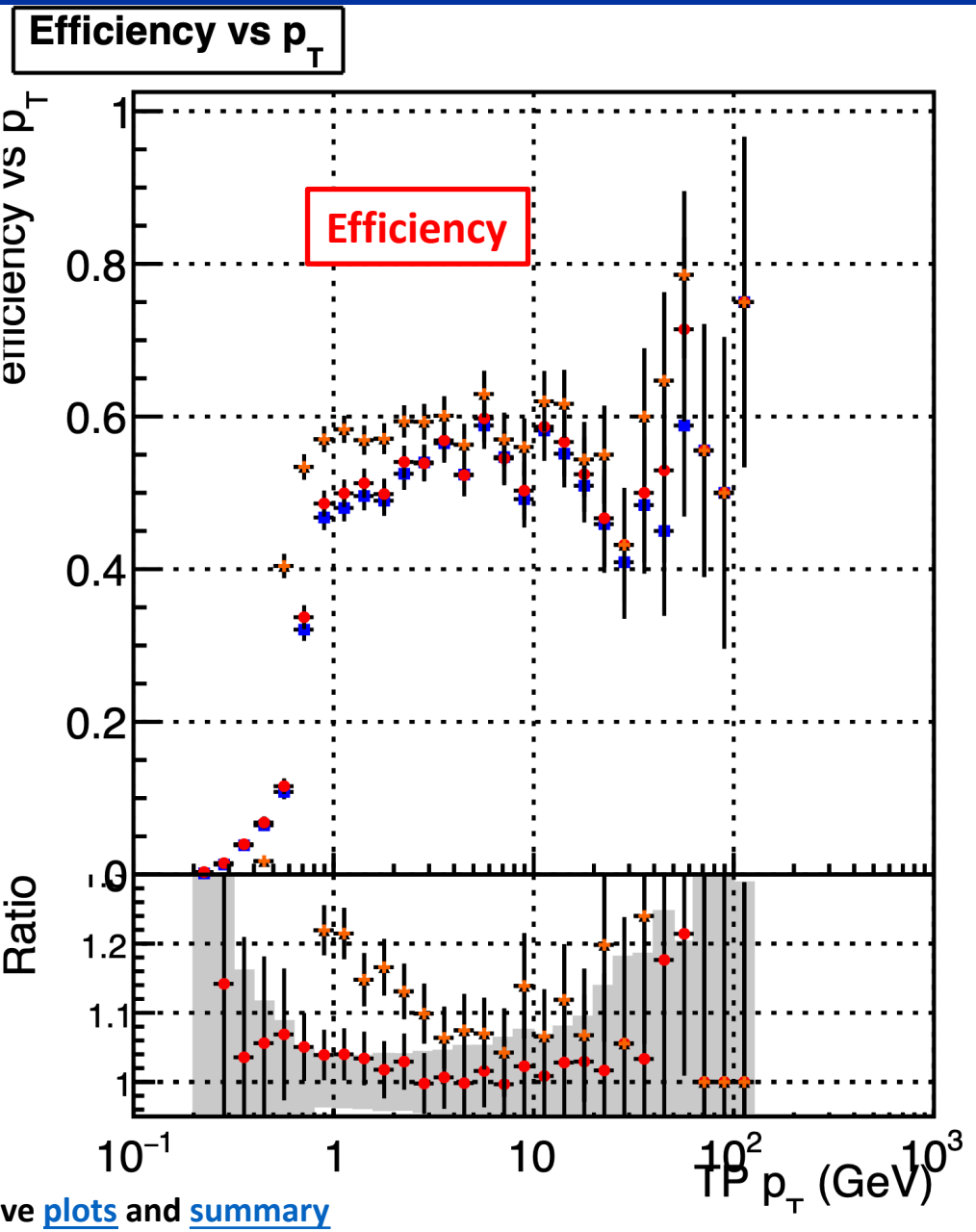




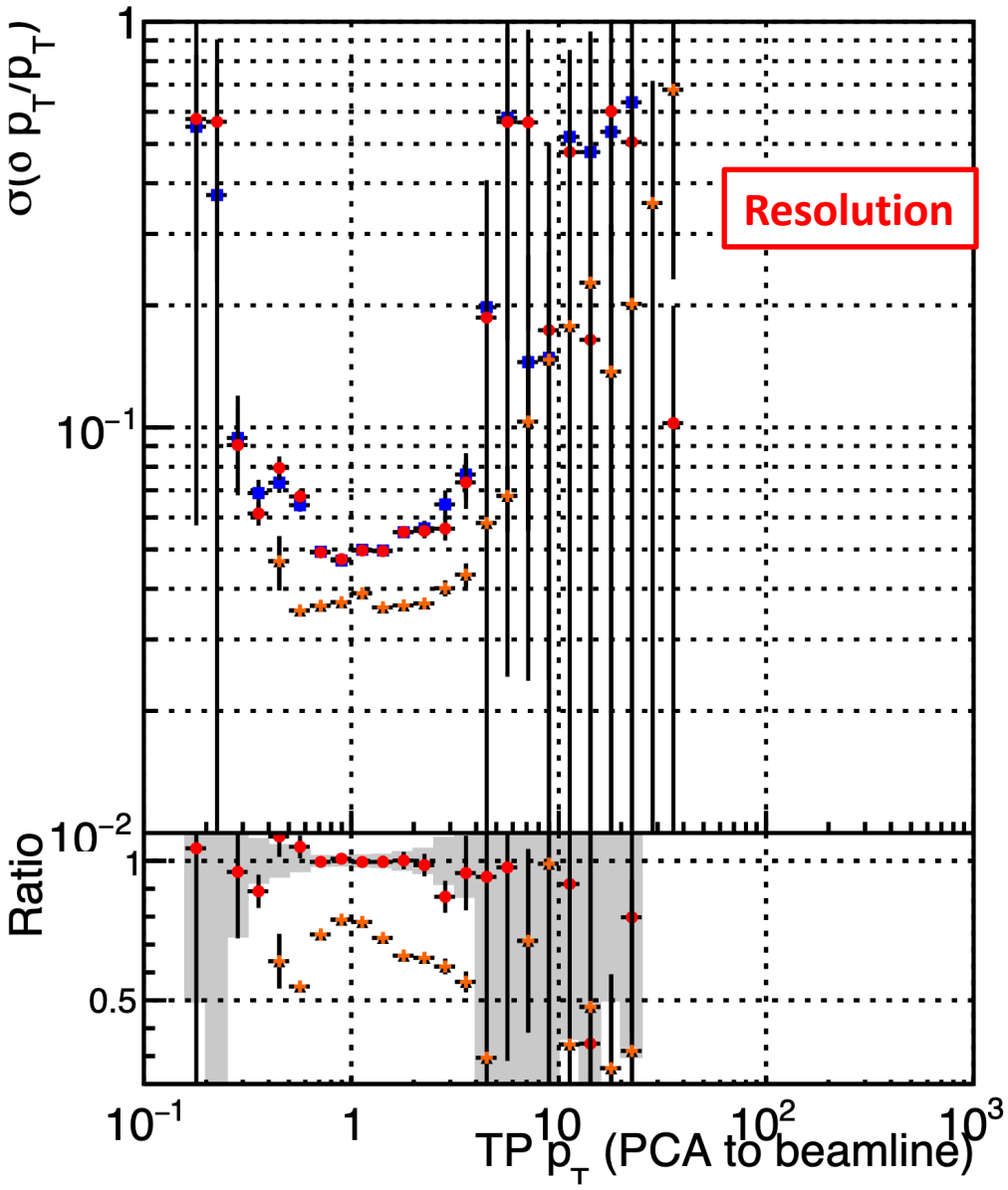
Resolution

Live plots and summary

Patatrack Pixel Track Validation – Physics Peformance

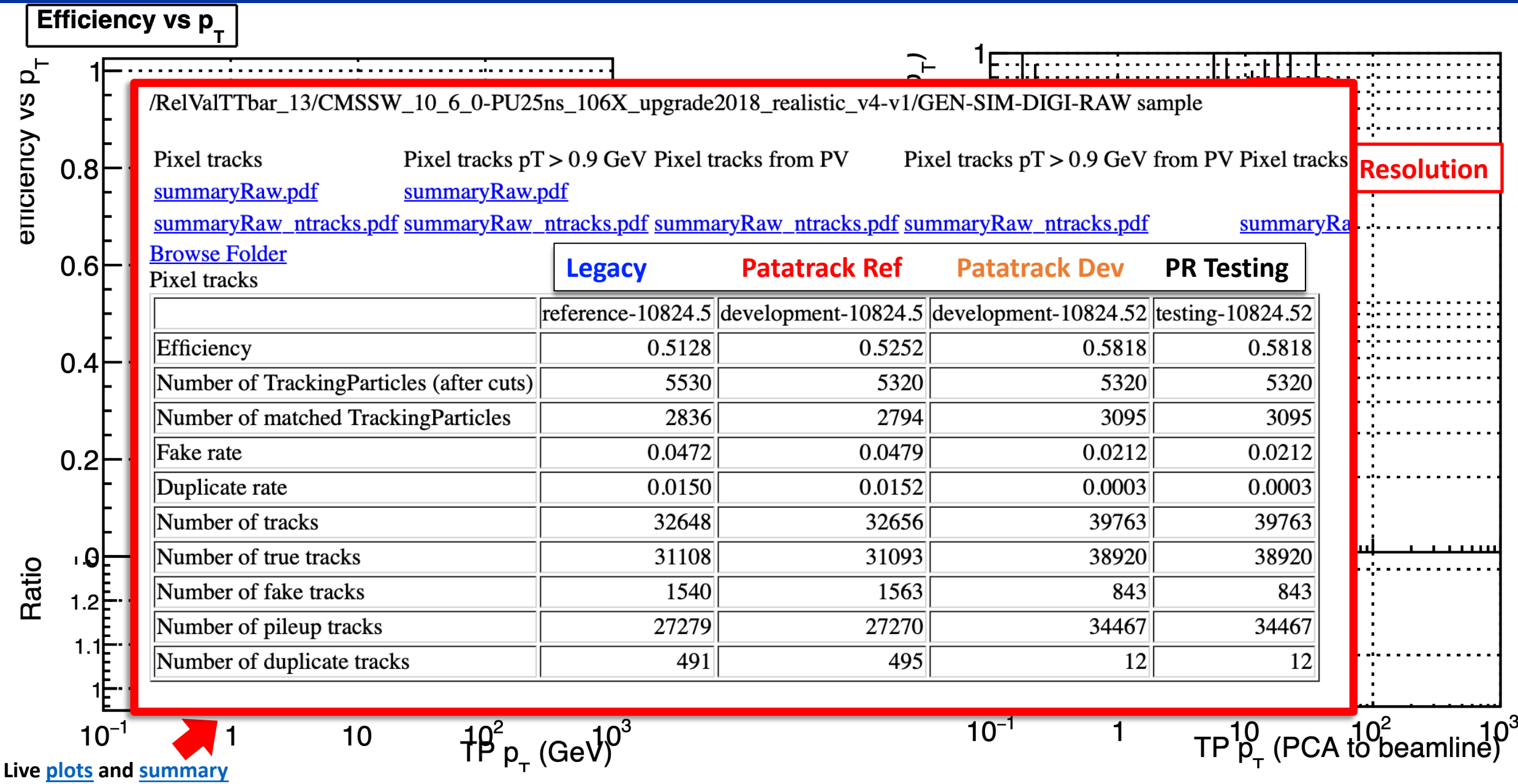


- Legacy
- Patatrack Ref
- Patatrack Dev
- PR Testing



Live plots and summary

Patatrack Pixel Track Validation – Physics Peformance



/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-v1/GEN-SIM-DIGI-RAW sample

Pixel tracks	Pixel tracks pT > 0.9 GeV	Pixel tracks from PV	Pixel tracks pT > 0.9 GeV from PV	Pixel tracks
summaryRaw.pdf	summaryRaw.pdf			
summaryRaw_ntracks.pdf	summaryRaw_ntracks.pdf	summaryRaw_ntracks.pdf	summaryRaw_ntracks.pdf	summaryRa

Browse Folder

Pixel tracks

Legacy Patatrack Ref Patatrack Dev PR Testing

	reference-10824.5	development-10824.5	development-10824.52	testing-10824.52
Efficiency	0.5128	0.5252	0.5818	0.5818
Number of TrackingParticles (after cuts)	5530	5320	5320	5320
Number of matched TrackingParticles	2836	2794	3095	3095
Fake rate	0.0472	0.0479	0.0212	0.0212
Duplicate rate	0.0150	0.0152	0.0003	0.0003
Number of tracks	32648	32656	39763	39763
Number of true tracks	31108	31093	38920	38920
Number of fake tracks	1540	1563	843	843
Number of pileup tracks	27279	27270	34467	34467
Number of duplicate tracks	491	495	12	12

Live plots and summary

Patatrack Pixel Track Validation – Checks

For the CUDA-enabled workflows the pixel reconstruction job is also run **multiple times** under **cuda-memcheck:**

- `cuda-memcheck --tool initcheck`
- `cuda-memcheck --tool memcheck --leak-check full --report-api-errors all`
- `cuda-memcheck --tool synccheck`

Logs & reports are produced and uploaded.

Merged Replace use of API wrapper stream and event with plain CUDA, part 1 #

fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:`

logs and `nvprof` / `nvvp` profiles

/RelValTTbar_13/CMSSW_10_6_0-PU25ns_106X_upgrade2018_realistic_v4-
RAW

- reference release, workflow 10824.5
- development release, workflow 10824.5
- development release, workflow 10824.51
- development release, workflow 10824.52
  - ✓ `cuda-memcheck --tool initcheck` (report, log) did not find any errors
  - ✓ `cuda-memcheck --tool memcheck --leak-check full --report-api-errors all` (report, log) did not find any errors
  - ✓ `cuda-memcheck --tool synccheck` (report, log) did not find any errors
- development release, workflow 136.86452
- testing release, workflow 10824.5
- testing release, workflow 10824.51
- testing release, workflow 10824.52
  - ✓ `cuda-memcheck --tool initcheck` (report
  - ✓ `tool memcheck --leak-c`
  - any errors
  - `tool synccheck` (report
  - 6.86452

/RelValZMM_13/CMSSW_10_6_0-PU25ns_106X_upgrad
RAW

```
========= CUDA-MEMCHECK
========= LEAK SUMMARY: 0 bytes leaked in 0 allocations
========= ERROR SUMMARY: 0 errors
```

```
%MSG-i CUDAService:  (NoModuleName) 24-Oct-2019 19:59:44 CEST pre-events
CUDA runtime successfully initialised, found 1 compute devices.

CUDA device 0: Tesla T4
  compute capability:          7.5 (sm_75)
  streaming multiprocessors:             40
  CUDA cores:                          2560
  single to double performance:        32:1
  compute mode:           default (shared)
  memory:  14969 MB free /  15079 MB total
  constant memory:                    64 kB
  L2 cache size:                    4096 kB
  L1 cache mode:   local and global memory
```

Patatrack Pixel Track Validation – Throughput

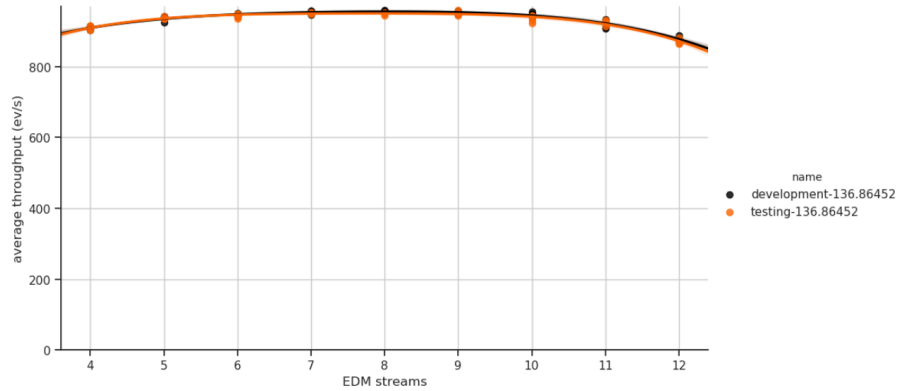
Throughput on DATA samples is measured for the testing PR and Patatrack reference. Plots are produced (scanning # EDM streams) and uploaded. The effect of each PR is **traced** and **recorded**.

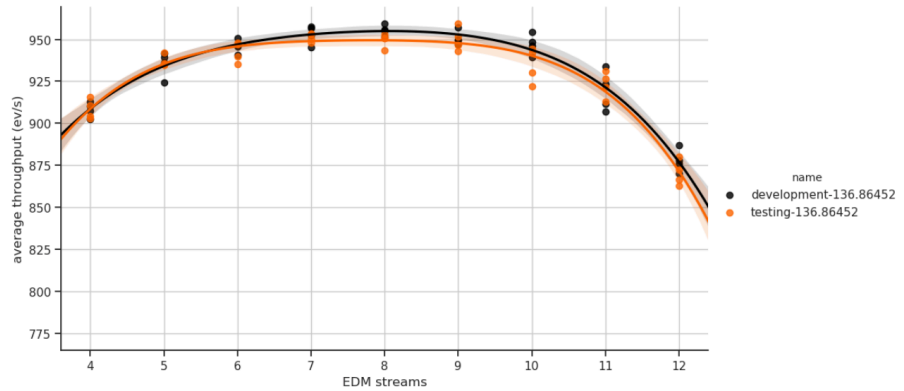
Merged **Replace use of API wrapper stream and event with plain CUDA, part 1** #389

fwyzard merged 6 commits into `cms-patatrack:CMSSW_11_0_X_Patatrack` from `makortel:removeWrapp`

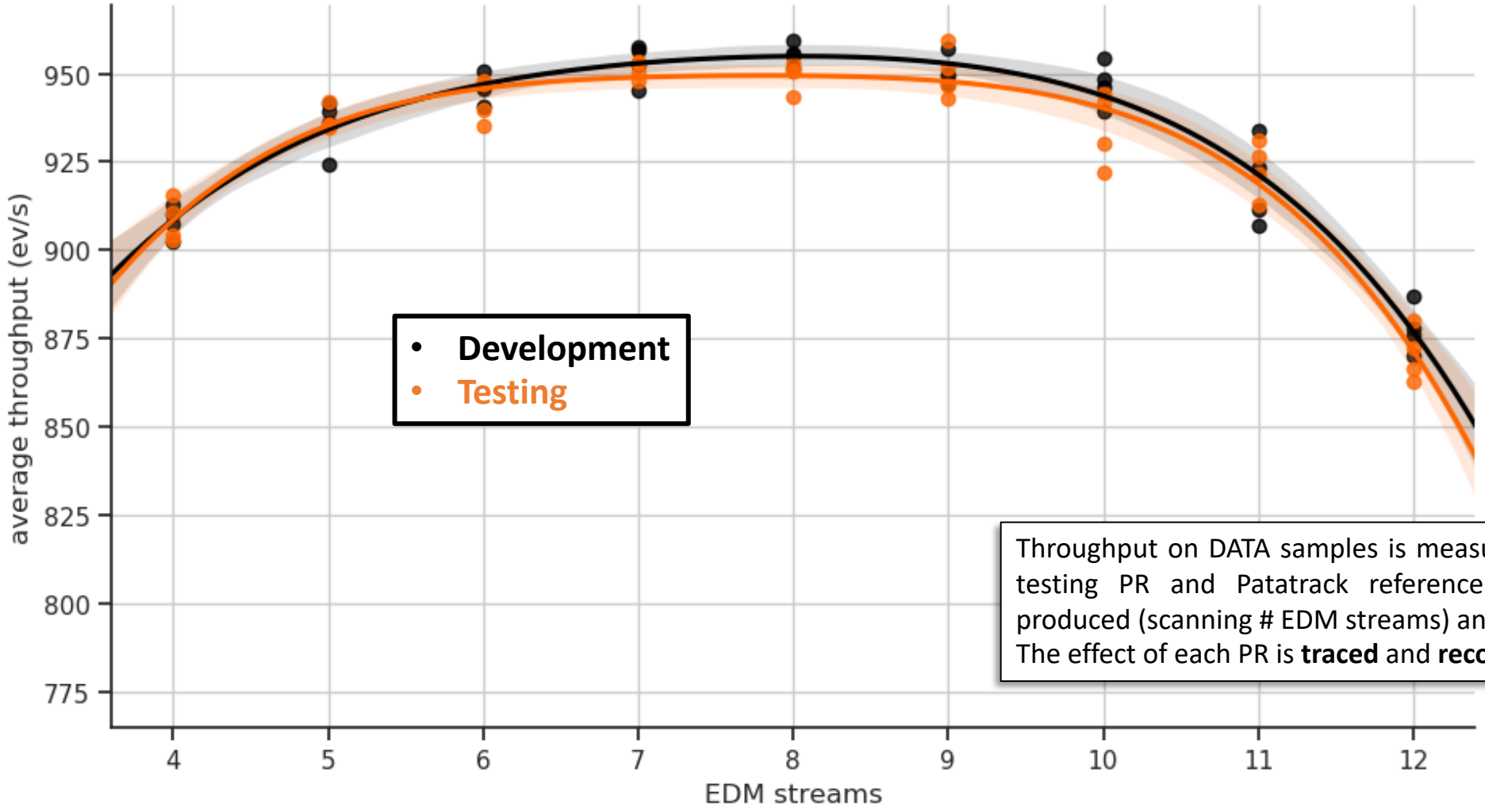
Throughput plots

/EphemeralHLTPhysics1/Run2018D-v1/RAW run=323775 lumi=53





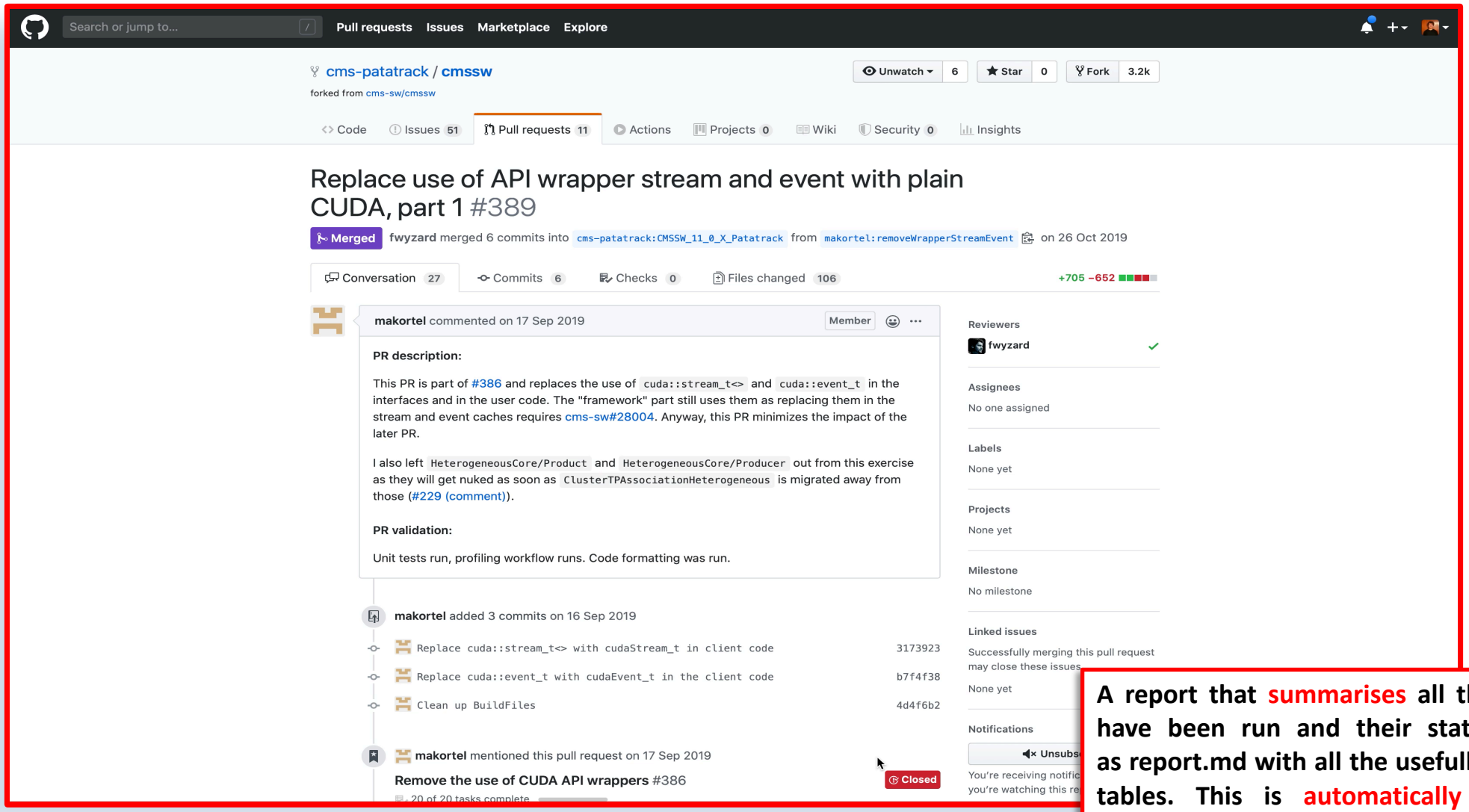
Patatrack Pixel Track Validation – Throughput



Throughput on DATA samples is measured for the testing PR and Patatrack reference. Plots are produced (scanning # EDM streams) and uploaded. The effect of each PR is **traced** and **recorded**.

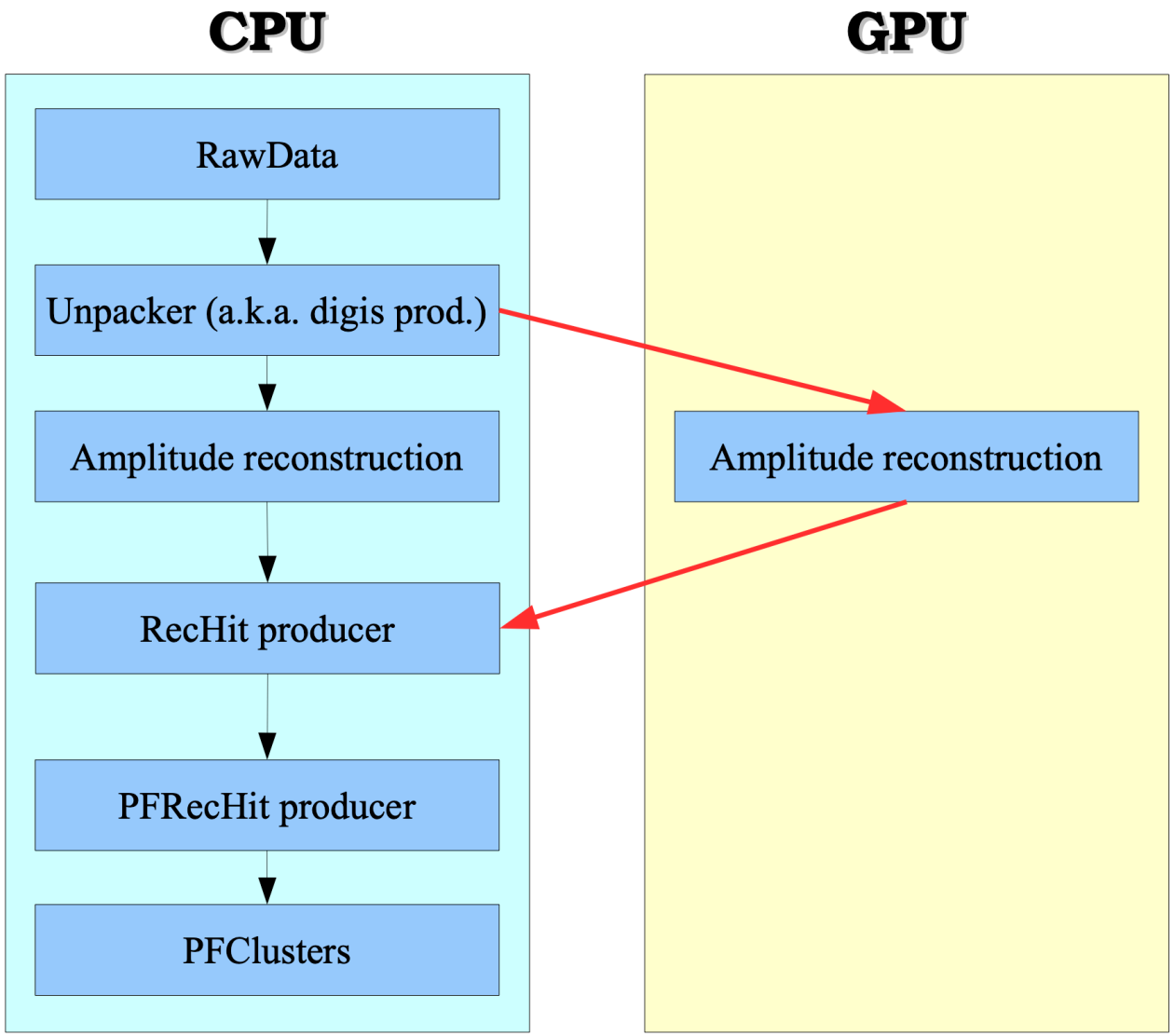
Patatrack Pixel Track Validation – PR Summary

All the procedure happens thorugh a **single (long) validate** script allowing to compare (multiple) **PRs** or **local** developments



A report that summarises all the jobs that have been run and their status is saved as report.md with all the usefull links, plots, tables. This is automatically posted on GitHub as a comment to the PR.

Calorimeter Local Reconstruction – Workflow

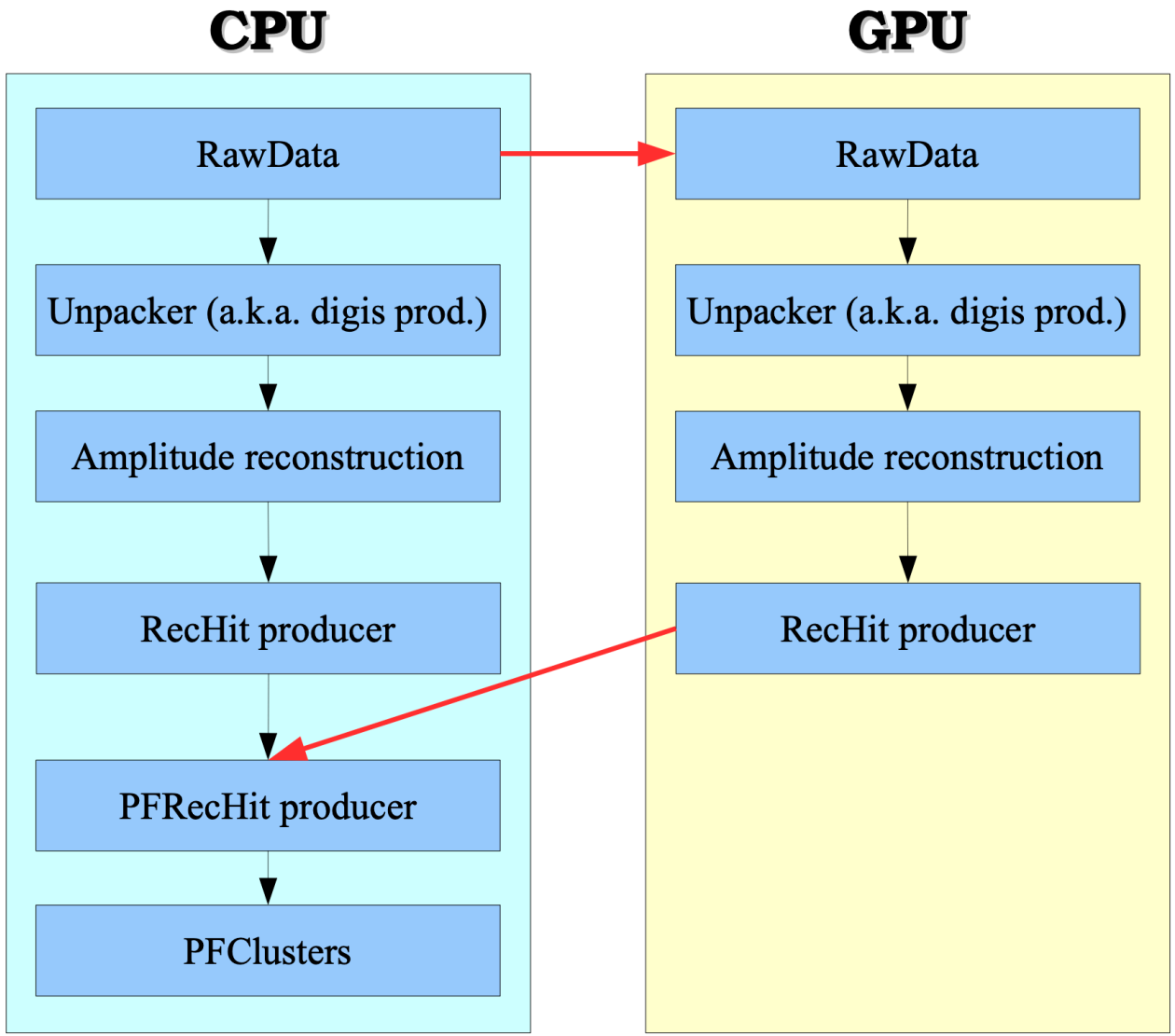


The **goal:** port calorimeter local reconstruct on the GPU.

The **workflow:**

- pulse shape analysis reconstruction (a.k.a. **multifit** for ECAL, **MAHI** for HCAL) is a similar algorithm for both detectors. **Parallelization** per crystal and **restructuring** of amplitude algorithm to **expose parallelization.**
- **unpacker** ported on GPU
- RecHit producer **ported** to GPU:
  - convert the **amplitudes** in ADC counts
  - transform into **energy**: APD/PN, alpha, flags, IC,
  - conditions ported/transferred to GPU as well

Calorimeter Local Reconstruction – Workflow

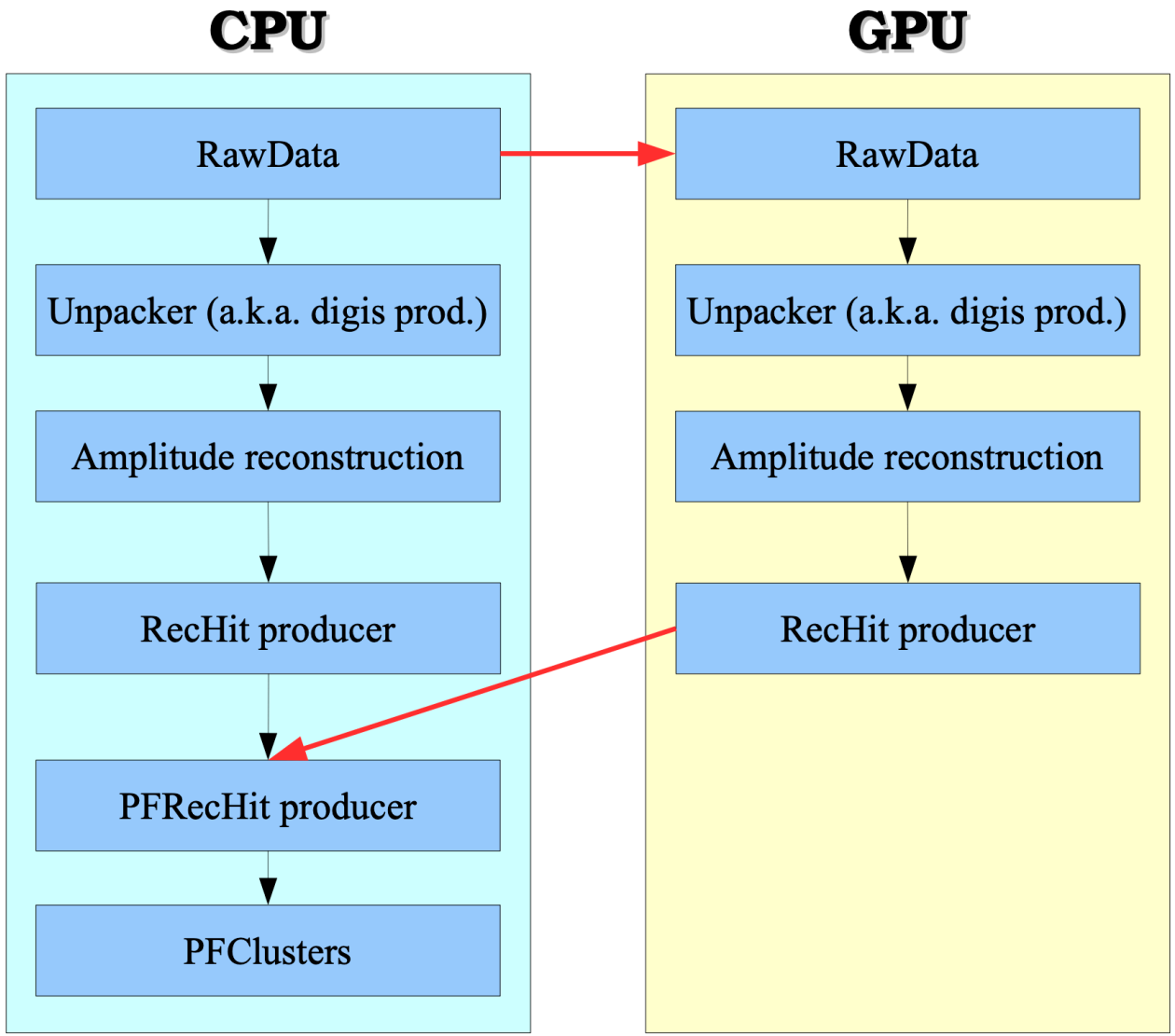


The **goal:** port calorimeter local reconstruct on the GPU.

The **workflow:**

- pulse shape analysis reconstruction (a.k.a. **multifit** for ECAL, **MAHI** for HCAL) is a similar algorithm for both detectors. **Parallelization** per crystal and **restructuring** of amplitude algorithm to **expose parallelization.**
- **unpacker** ported on GPU
- RecHit producer **ported** to GPU:
  - convert the **amplitudes** in ADC counts
  - transform into **energy**: APD/PN, alpha, flags, IC,
  - conditions ported/transferred to GPU as well

Calorimeter Local Reconstruction – Validation

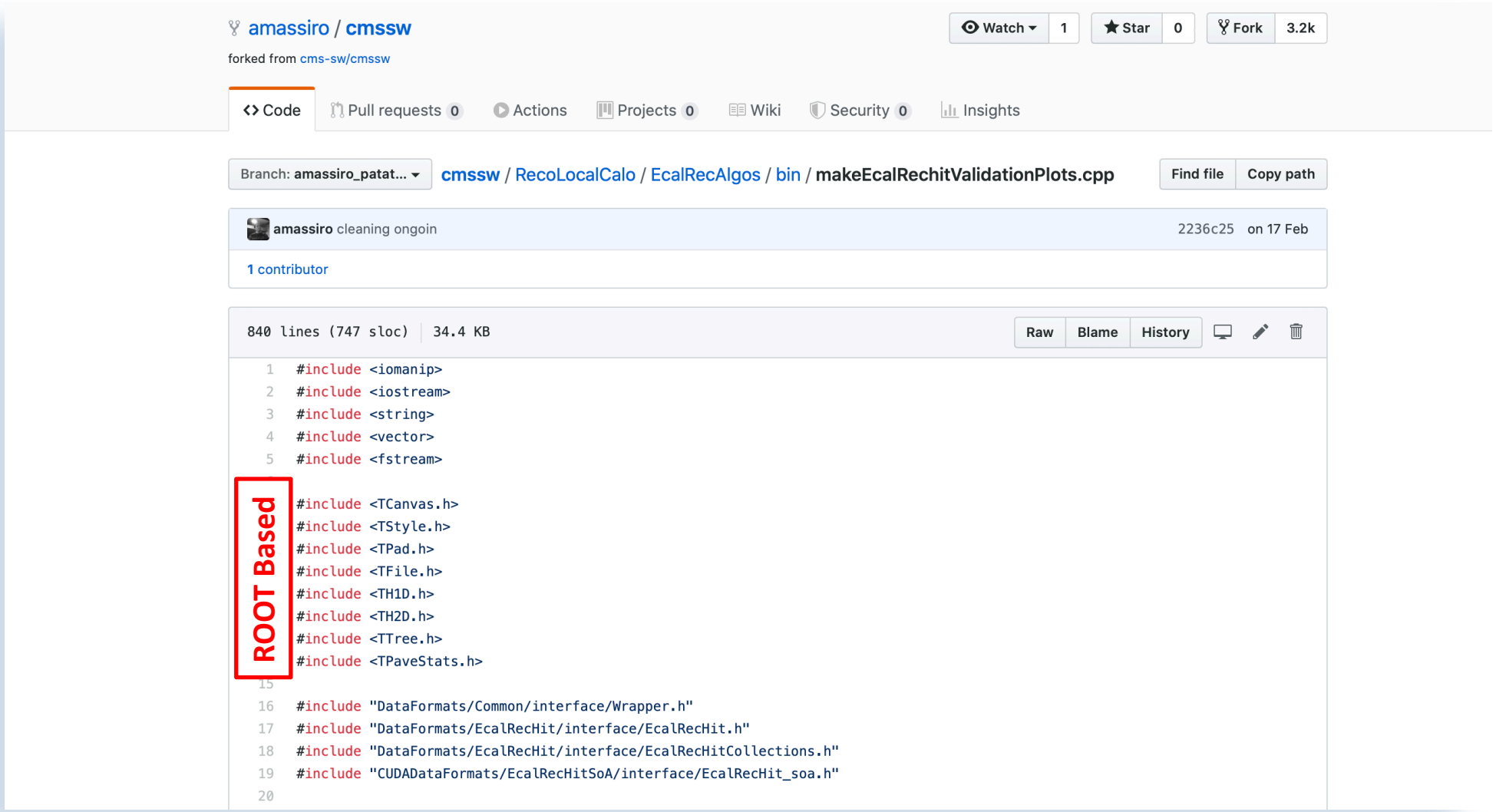


The **validation (ECAL, similar for HCAL):**

- is performed event by event for **each channel (crystal)**
- compares MultiFit **amplitueds**, RecHits **#**,**energies**, χ^2,flags
- introduces **new** plots (to be integrated within the CMSSW **DQM** context).
- is "**lower level**" than the Pixel Tracks validation. **Rapidly** evolving. Based on A.Massironi scripts.

ECAL - Validation

A different approach with respect to Pixel Tracks. **Legacy CPU** & **Patatrack GPU** workflows run on the same events and both collections of deliverables are written in the Event.



ECAL - Validation

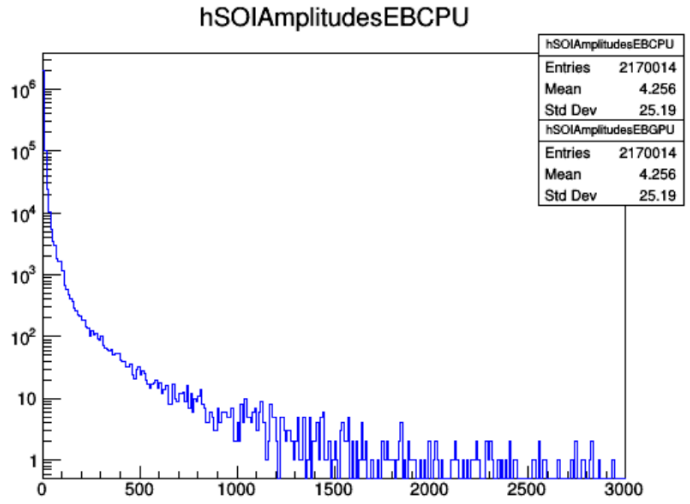
A different approach with respect to Pixel Tracks. **Legacy CPU** & **Patatrack GPU** workflows run on the same events and both collections of deliverables are written in the Event.

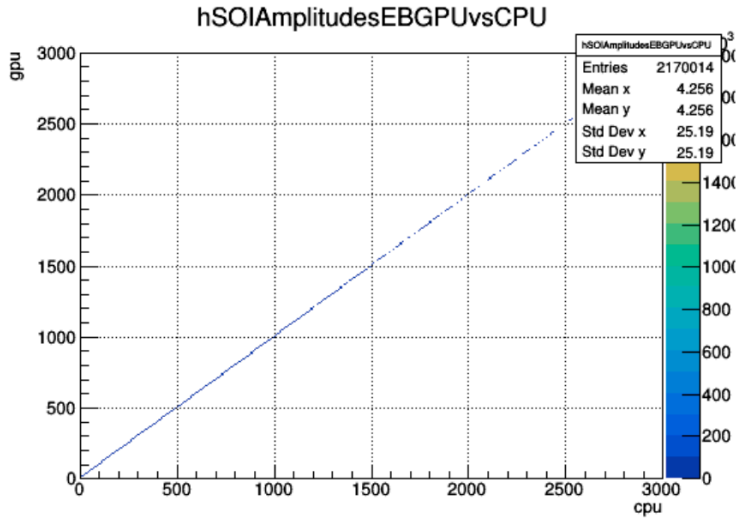
Metric histograms filled event by event

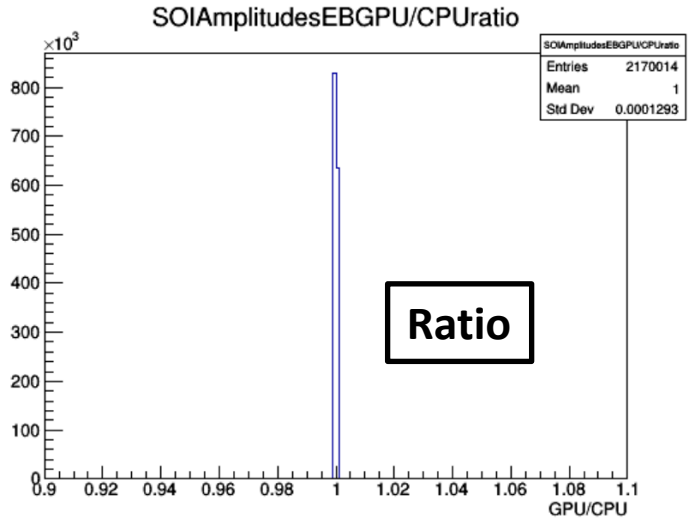
```
144   std::cout << "validating file " << fileName << std::endl;
145   TFile rf{fileName.c_str()};
146   TTree *rt = (TTree*)rf.Get("Events");
147
148   // Allocating the appropriate data to their respective pointers
149   rt->SetBranchAddress("ecalTagsoaecalRecHit_ecalCPURecHitProducer_EcalRecHitsEB_RECO."  &wgpuEB);
150   rt->SetBranchAddress("ecalTagsoaecalRecHit_ecalCPURecHitProducer_EcalRecHitsEE_RECO."  &wgpuEE);
151   rt->SetBranchAddress("EcalRecHitsSorted_ecalRecHit_EcalRecHitsEB_RECO."  &wcpuEB);
152   rt->SetBranchAddress("EcalRecHitsSorted_ecalRecHit_EcalRecHitsEE_RECO."  &wcpuEE);
153
154   constexpr float eps_diff = 1e-3;
155
156   // accumulate sizes for events and sizes of each event on both GPU and CPU
157 //   auto const nentries = rt->GetEntries();
158   int nentries = rt->GetEntries();
159
160   //---- AM: tests
161   if (nentries > 1000) {
162     nentries = 1000;
163   }
164 //   nentries = 1;
165
166   std::cout << "#events to validate over: " << nentries << std::endl;
167   for (int ie=0; ie<nentries; ++ie) {
168     rt->GetEntry(ie);
169
170 //     const char* ordinal[] = { "th", "st", "nd", "rd", "th", "th", "th", "th", "th", "th" };
171     auto cpu_eb_size = wcpuEB->bareProduct().size();
172     auto cpu_ee_size = wcpuEE->bareProduct().size();
173     auto gpu_eb_size = wgpuEB->bareProduct().energy.size();
174     auto gpu_ee_size = wgpuEE->bareProduct().energy.size();
175     float eb_ratio = (float) gpu_eb_size/cpu_eb_size;
176     float ee_ratio = (float) gpu_ee_size/cpu_ee_size;
177
178     // Filling up the histograms on events sizes for EB and EE on both GPU and CPU
179     hRechitsEBGPU->Fill(gpu_eb_size);
180     hRechitsEBCPU->Fill(cpu_eb_size);
181     hRechitsEEGPU->Fill(gpu_ee_size);
182     hRechitsEECPU->Fill(cpu_ee_size);
183     hRechitsEBGPUvsCPU->Fill(cpu_eb_size, gpu_eb_size);
184     hRechitsEEGPUvsCPU->Fill(cpu_ee_size, gpu_ee_size);
185     hRechitsEBGPUCPUratio->Fill(eb_ratio);
```

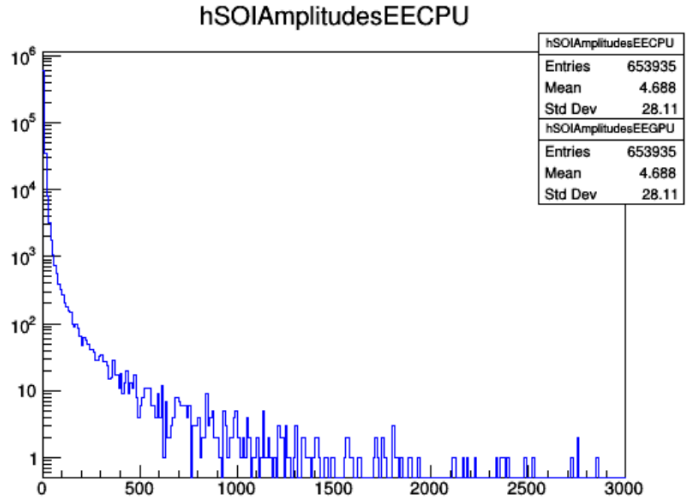
ECAL Validation – Amplitudes

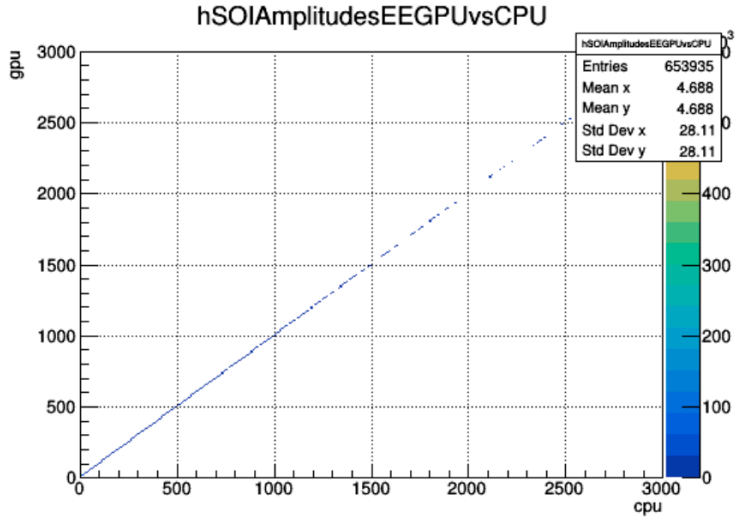
Amplitude reconstruction: **perfect** match

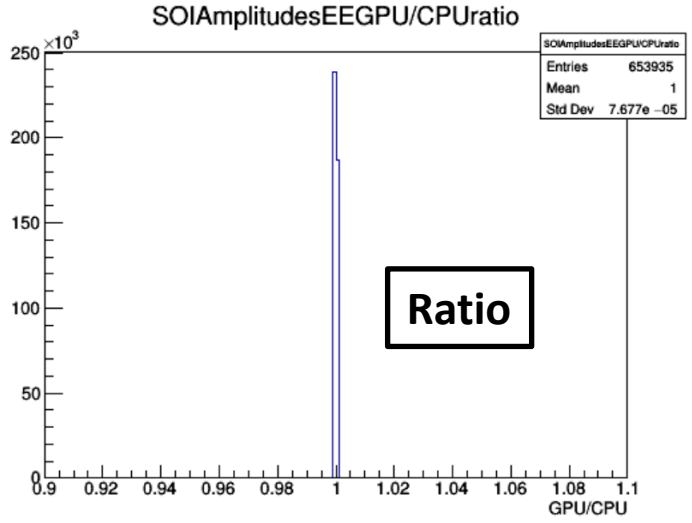






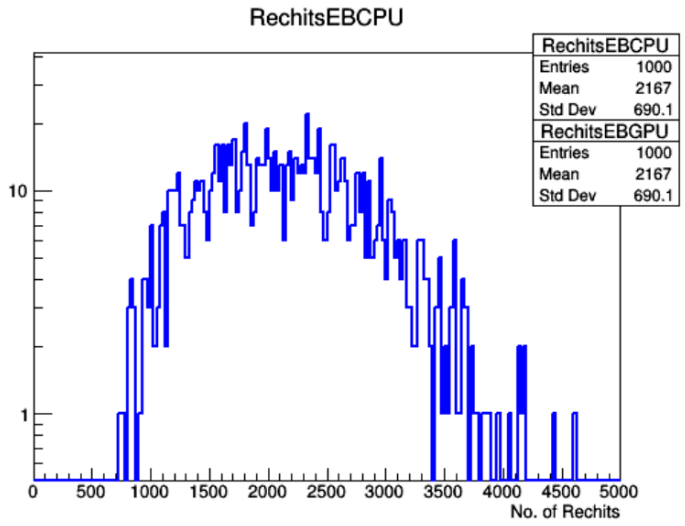


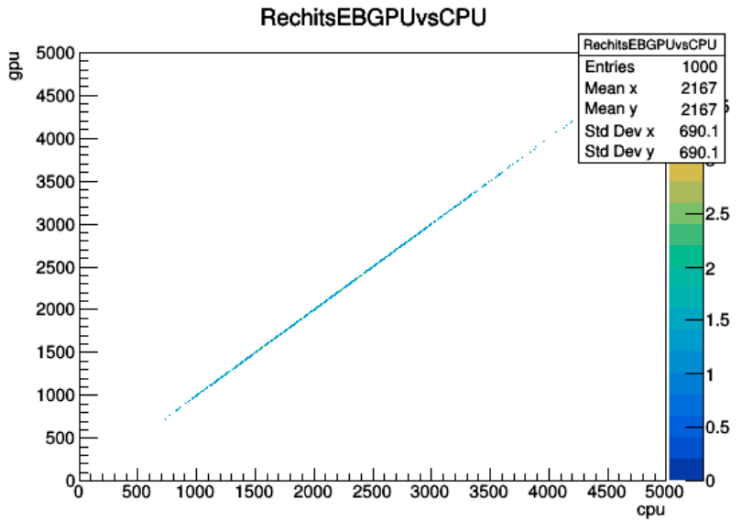


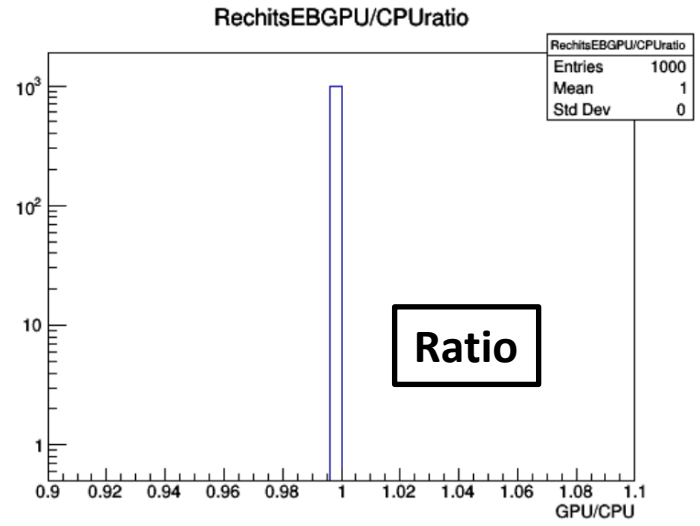


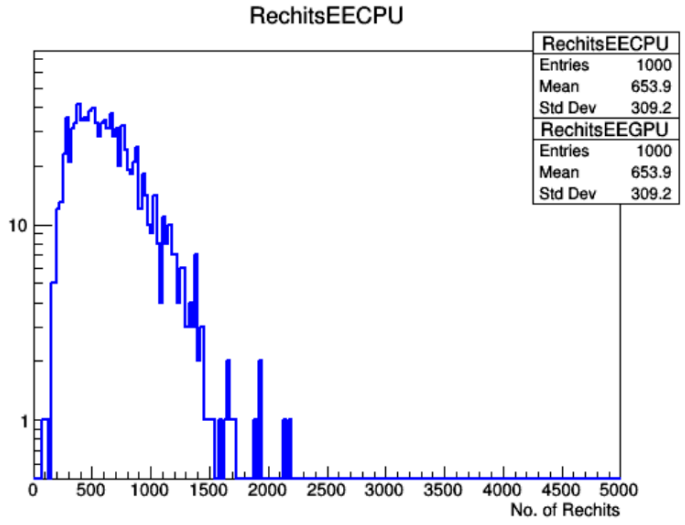
ECAL Validation –RecHits

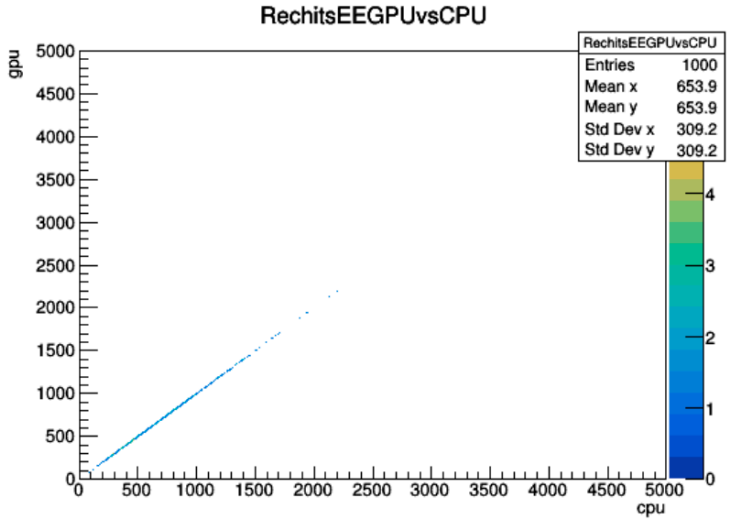
Number of RecHits**: perfect** match

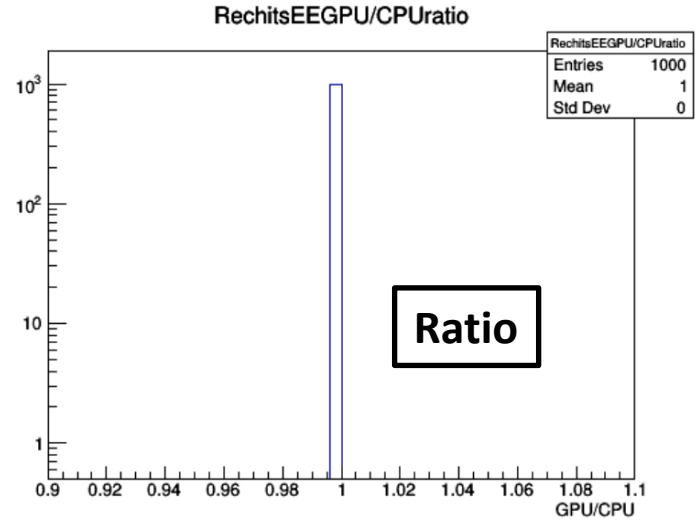












Summary

For Pixel Tracks & Calorimeter Local Reconstruction different **approaches** have been followed and then **different validation procedures** have been developed.

The **common ground**:

- the comparison among different platforms is mainly based on **physics performance plots** comparison
- features/tools **inherited** from the already existing **CMS validation** infrastructure (DQM,ROOT, Tracking Validation) and adapted to the new collections & requirements

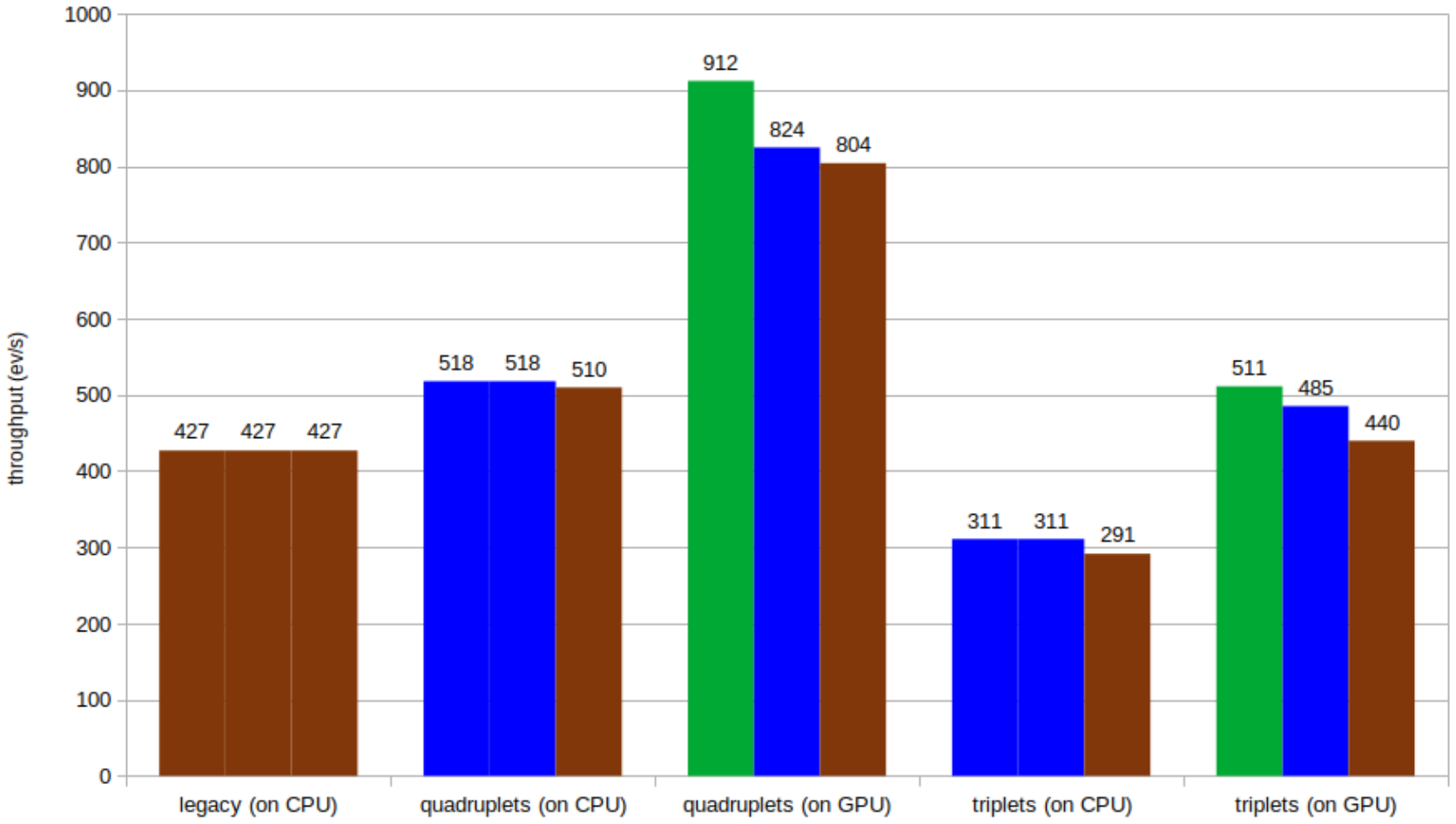
For **ECAL/HCAL** validation: based on the comparison of the GPU/CPU **reconstructed deliverables**. Done **event by event, channel by channel.**

For **Pixel Tracks** in addition**:**

- validation process fully **automatised**
- **PR** validation implemented
- computational performace (**throughput**) and **checks** for the new **development** on **GPU** added

Backup

Pixel Tracks – Computational Performance



CPU

- dual socket Xeon Gold 6130 2 × 16 cores (2 x 32 threads)
- throughput measured on a full node
- 4 jobs with 16 threads

GPU

- single NVIDIA Tesla T4
- 2560 CUDA cores
- single job with 10-16 concurrent events

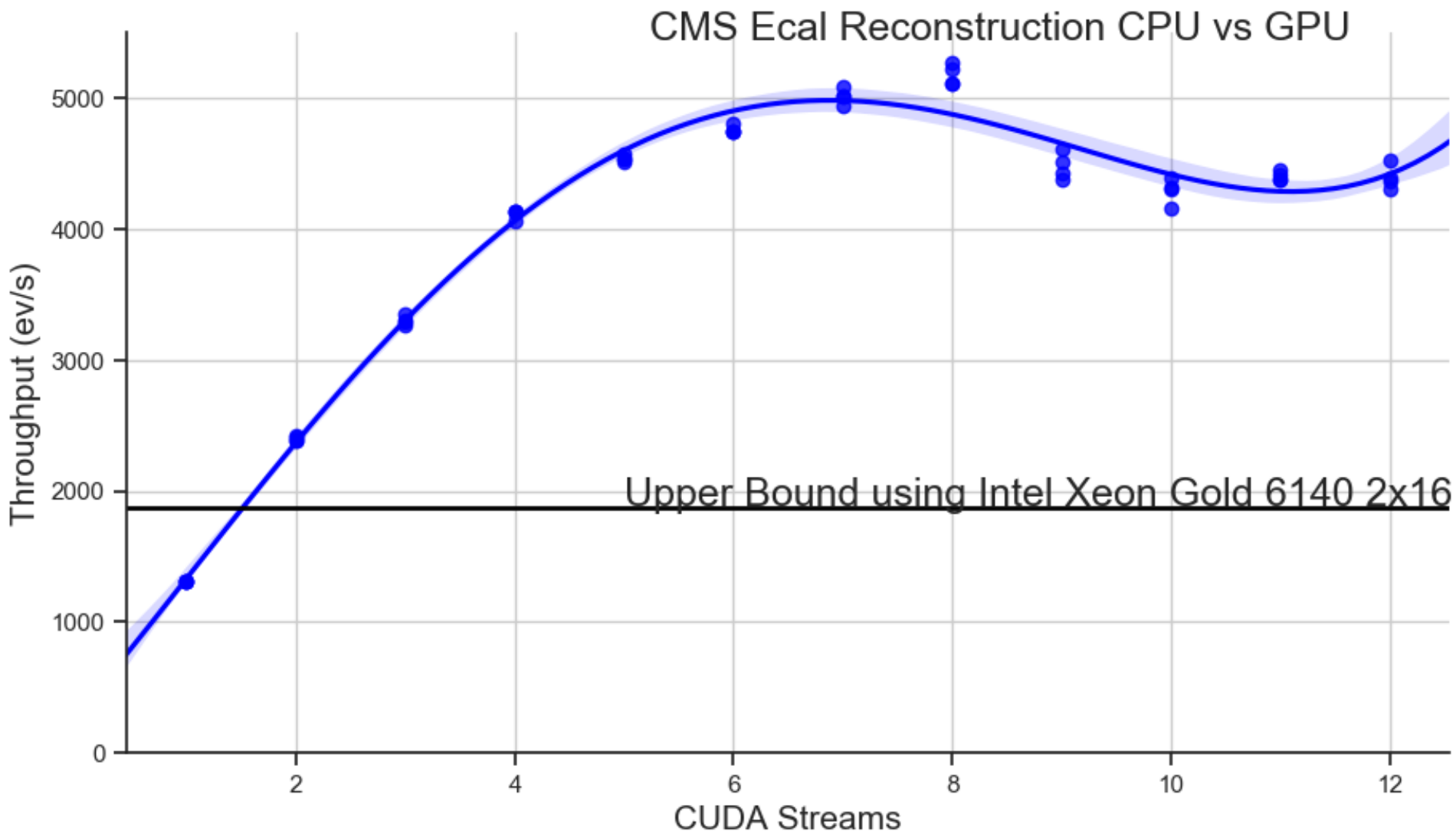
Transfer

- on demand
- small impact on event throughput

Conversion

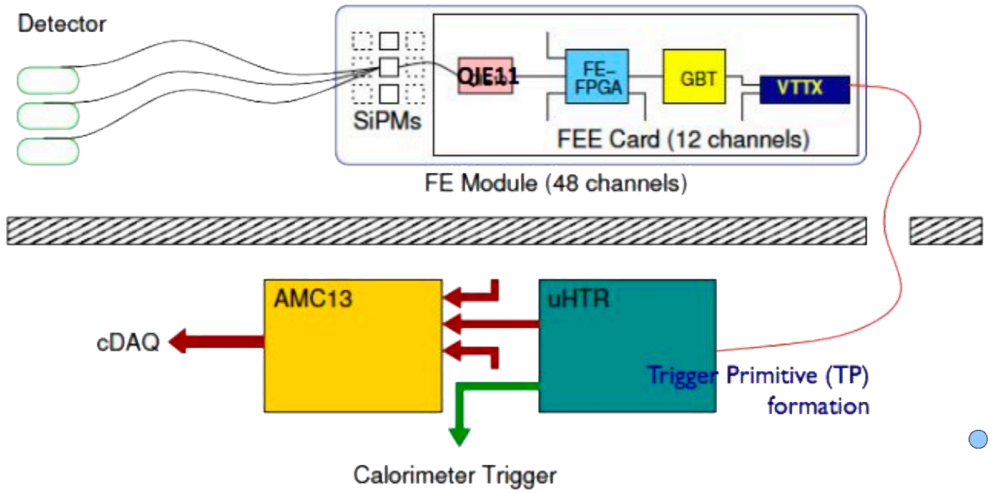
- on demand, to be minimised
- small impact on event throughput
- high cost in CPU usage

ECal Reconstruction – Computational performance



HCAL & ECAL – Readout

Hadronic CALorimeter

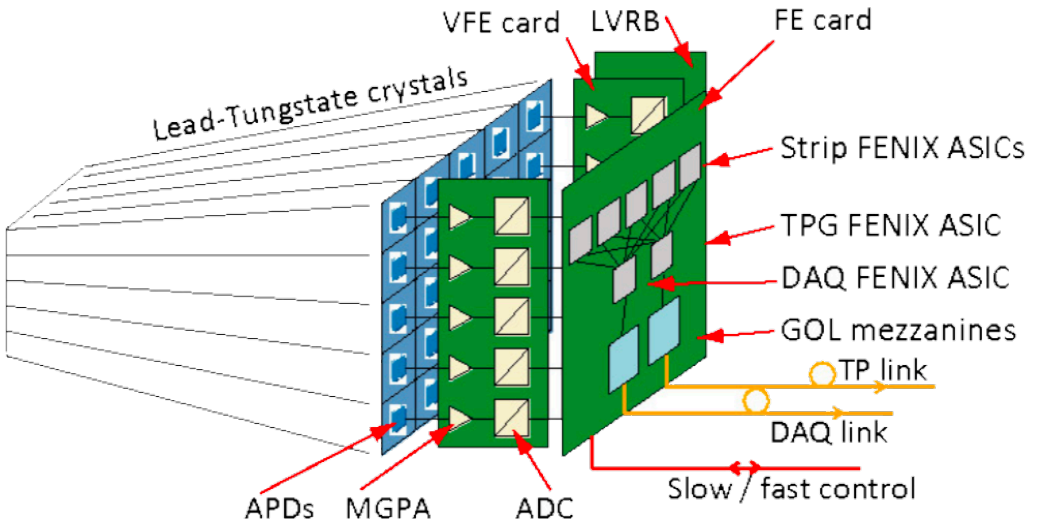


- hadronic showers
- brass and scintillator (HB) + steel and quartz fiber (HE)
- HPD in HB & SiPM in HE
- 2592 (HB) + 3384 x 2 (HE) = 9360 independent channels

  ↓

  9072 for HL-HLC

Electromagnetic CALorimeter (ECAL)

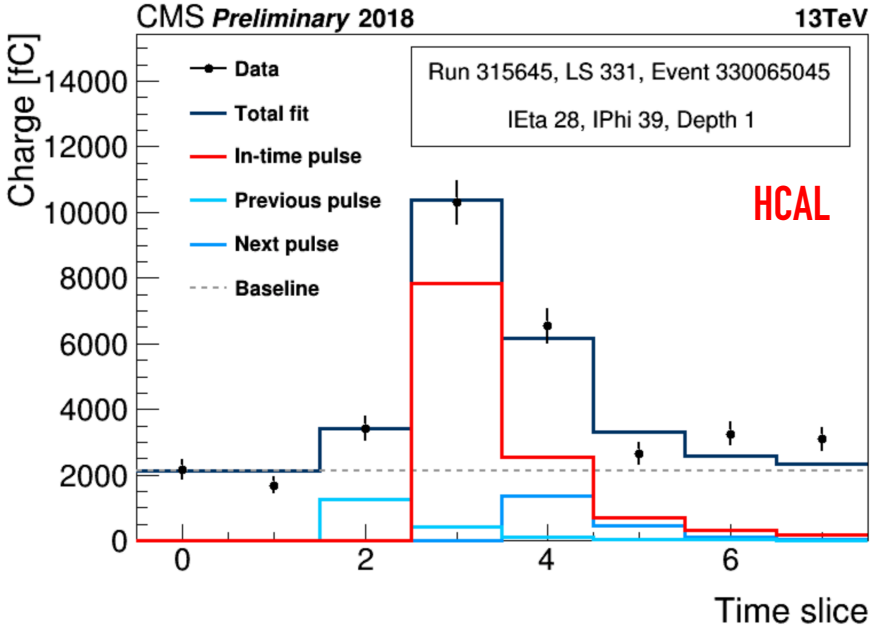


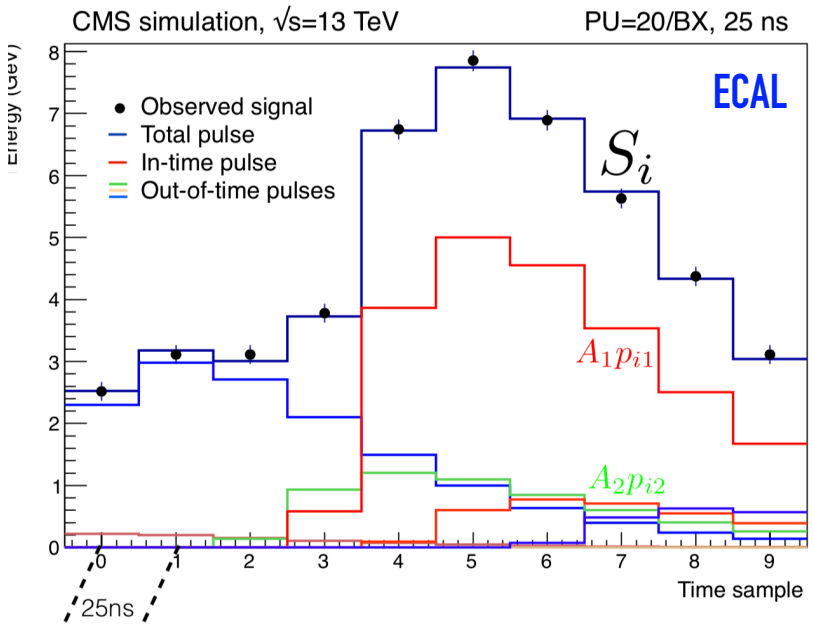
- e/γ
- lead tungstate ($PbWO_4$) as scintillating crystals
- VPT in EE & APD in EB
- 61200 (EB) + 7324 x 2 (EE) = 75848 independent channels

Pulse Shape Analysis

Collisions at 40 MHz, sampling at 25ns: in each sample there could be energy deposit from previous and following collisions.

Need to disentangle multiple contributions : in-time signal + out-of-time pulses





$$\chi^2 = \sum_{i=1}^{10} \frac{\left(\sum_{j=1}^{M} A_j \times p_{ij} - S_i\right)^2}{\sigma_{S_i}^2}$$

$$\chi^2 = \left(\sum_{j=1}^{N_{\text{pulse}}} A_j \vec{p}_j - \vec{S}\right)^T \mathbf{C}^{-1} \left(\sum_{j=1}^{N_{\text{pulse}}} A_j \vec{p}_j - \vec{S}\right)$$

covariance matrices $\mathbf{C} = \mathbf{C}_{\text{noise}} \oplus \sum_{j=1}^{N_{\text{pulse}}} A_j^2 \mathbf{C}_{\text{pulse}}^j$

- S_i digitized amplitudes (data) ($i = 0, \ldots, N_{TS}$ with $N_{TS-HCAL} = 7$ & $N_{TS-ECAL} = 9$
- $\boldsymbol{C}$ pulses covariance matrices (including noise)
- A_j amplitudes from the pulse at bunch cross C (A_0 in-time deposit)
- $\vec{p}$ pulses shapes extracted from LHC single bunch. Identical, 25ns shifted