https://doi.org/10.1051/epjconf/202533701037

Towards a GPU-enabled electron seeding algorithm in the CMS experiment

Charis-Kleio Koraka[1,*]

[1]On behalf of the CMS collaboration

Abstract. Electrons are one of the key particles that are detected by the CMS experiment and are reconstructed using the CMS software (CMSSW). Reconstructing electrons in CMSSW is a computational intensive task that is split into several steps, seeding being the most time consuming one. During the electron seeding process, the collection of tracker hits (seeds) is significantly reduced by selecting only seeds that are compatible with a hypothesized electron trajectory. This contribution will describe the process of redesigning the electron seeding algorithm in a parallelizable way that will exploit the massive parallelism that GPUs can offer. The new algorithm code base is implemented using the Alpaka library, a performance portability library that allows having a single code base for execution on different types of hardware.

1 Introduction

The High Luminosity era of the Large Hadron Collider (HL-LHC) [1] will bring increased luminosity and higher levels of pile-up. In addition, the CMS detector [2, 3] is set to undergo a series of significant upgrades to its various components [5–11], resulting in a more complex detector with additional readout channels. These changes will introduce unprecedented computational challenges, demanding substantial improvements in data processing efficiency. To manage the anticipated increase in CPU resource requirements while staying within the allocated CPU budget, the CMS experiment is exploring various R&D activities. Projections of the CPU requirements over time are shown in Fig. 1.

Among these R&D initiatives, the CMS experiment is focusing on redesigning key components of its event reconstruction pipeline to take advantage of Graphics Processing Units (GPUs) and heterogeneous resources. This document discusses the development of a GPU-accelerated electron seeding algorithm and its potential to enhance the efficiency of CMS event processing. The electron seeding algorithm identifies hit patterns consistent with electron trajectories and is the most computationally demanding task in the CMS electron reconstruction chain [12].

2 Electron reconstruction and seeding in CMS

Electrons are electromagnetically interacting particles that leave a distinctive signal in the CMS electromagnetic calorimeter (ECAL) as an isolated energy deposit. On top of this, due

---

[*] e-mail: charis.kleio.koraka@cern.ch

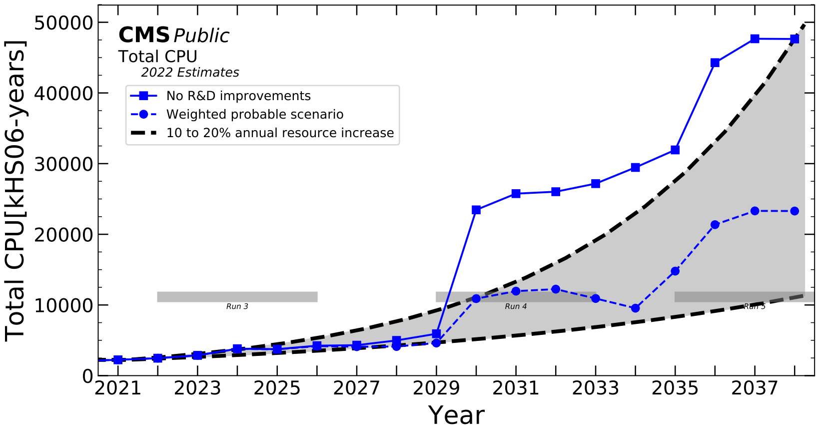

Figure 1. Projection of needed CPU resources into HL-LHC. The gray band represents the projected capacity of the CPU resources within flat budget. Two lines are drawn, each corresponding to one of two scenarios considered, a baseline scenario is which is described as "No R&D improvement" and the "Weighted Probable scenario" which corresponds to the most probable outcome incorporating R&D activities. Taken from Ref. [17].

to the fact that they carry charge, electrons are also associated with a trace of hits in the silicon trackers. The reconstruction software in CMS consists of several sequential steps that uses the aforementioned information to reconstruct electrons. These include:

- Superclustering in the ECAL, where energy deposits in different crystals are combined into a single super-cluster (SC) that captures the energy of the original electron
- Electron seeding, where tracker hit patterns consistent with electron trajectories are identified to reduce the number of candidate tracker hits that will be used to reconstruct the electron tracks.
- The Gaussian Sum Filter (GSF) track fitter which is used to include radiative losses from bremsstrahlung and produce the electron track information.
- Finally, algorithms exist for refining and calibrating the electron energy.

Among the above steps, the electron seeding is an iterative process which is particularly expensive in terms of computational resources, as it involves scanning through large numbers of tracker seeds, built by combining two (doublet seed) or three (triplet seed) hits residing in the inner (pixel) tracker, to identify potential electron candidates.

3 GPU-Enabled Electron Seeding Algorithm

In Run-3, the CMS High-Level Trigger (HLT) has already integrated GPU-based algorithms in its pipeline [14]. An example of the time spent in the various reconstruction modules, where some of the algorithms involved are running on GPU, can be seen in Fig. 2. The electron reconstruction (labeled as E/Gamma) accounts for up to 10% of the total reconstruction time, with the seeding step being the most computationally demanding one. This makes electron seeding a prime candidate for further optimization.

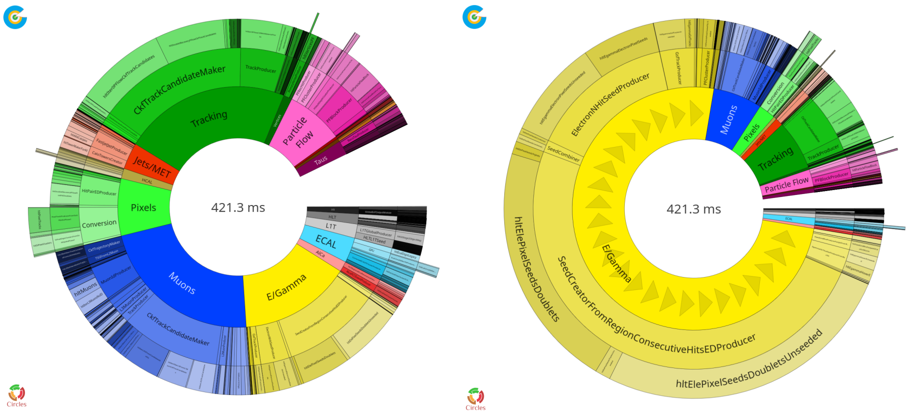

Figure 2. Left: Measurement of average reconstruction time spent per event at the High Level Trigger (HLT). The average time and the breakdown of the time spent in each CMS software (CMSSW) module in a CPU and GPU combined configuration is shown. The pixel tracking (light green slice), parts of the ECAL (light blue slice), part of the HCAL (light brown slice) and part of the Particle Flow (PF) reconstruction (magenta slice) is being offloaded to GPU. Right: Zoomed-in view of the light yellow slice in the left pie chart, detailing the breakdown of time spent specifically in the electron reconstruction workflow. Taken from [13].

3.1 Description of the electron seeding algorithm

Starting with two object collections, that of the ECAL SCs and that of the tracker seeds, the tracker seeds are formed from multiple hits in the inner (pixel) tracker, using combinations of two (doublets) or three (triplets) tracker hits.

The seeding process begins by selecting SCs that exceed a predefined energy threshold. For each SC, the corresponding electron trajectory is assumed to follow a helical path determined by the SC position, its transverse energy, and the magnetic field. The electron trajectory is then extrapolated from the ECAL toward the detector beamline under both positive and negative charge hypotheses. A geometric matching check is performed in the $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ or Δz x $\Delta\phi$ plane to determine whether the extrapolated trajectory falls within a predefined proximity of the innermost hit of a tracker seed. If the distance exceeds a set threshold, the seed is discarded, and the process moves to the next seed candidate. If the distance falls within the allowed range, the matching procedure continues. Next, the electron trajectory is further extrapolated to the beamline to refine the estimate of the z-coordinate of the primary vertex (PV), while the x- and y-coordinates are assumed to match those of the beamline position. Using this updated information, the trajectory is then propagated from the PV toward the surface of the second hit of the tracker seed. Another geometric matching check is performed to verify if the propagated state remains in the proximity of the second hit. This procedure is repeated for the third hit for triplet seeds. If all hits of the tracker seed are successfully matched, the seed is accepted and passed to the next stage of the electron reconstruction chain. An illustration of the above procedure is showcased in the schematic of Fig. 4.

3.2 Parallelization strategy

To adapt the electron seeding algorithm for execution on GPU, several modifications have been implemented:

- **Data Structures:** The algorithm makes use of the CMS generic Structure-of-Arrays (SoA) format to ensure efficient memory access patterns of the various data structures, reducing latency and optimizing performance. Both the input collections of SCs and tracker seeds as well as the output collection of matched electron seeds use the SoA paradigm. A schematic representation of the SoA access pattern is illustrated in Fig. 3.
- **Utility Functions:** Several components, not necessarily specific to the electron reconstruction, were adapted for GPU execution using the Alpaka performance portability library. These include the algorithm used for the helix propagation towards different surfaces (e.g. plane, cylinder, arbitrary surface), structures that hold information about the position, direction and rotation of surfaces and a structure that holds information about matched pairs of tracker seeds and SCs. Additionally, due to the difficulty of having a full description of the magnetic field for all the detector volume on the GPU, a simplified parabolic parametrization of the magnetic field was adapted and utilized.
- **Algorithm Logic:** The procedure outlined in Section 3.1 is repeated for every possible combination of a tracker seed, a SC, and an electron charge hypothesis. To maximize efficiency, we first considered the typical sizes of these collections. Under Run-3 conditions, the SC collection typically contains $O(10)$ elements, while the tracker seed collection is significantly larger, ranging from $O(10^2)$ to $O(10^4)$. Given the fact that the tracker seed collection is orders of magnitude larger, we choose to parallelize over these. In the GPU-enabled algorithm, each GPU thread processes a single electron seed, iterating over ECAL SCs and charge hypotheses while updating the hit-to-hypothesized-track matching state. The same track propagation as described in 3.1 with an inverted matching logic is therefore utilized.

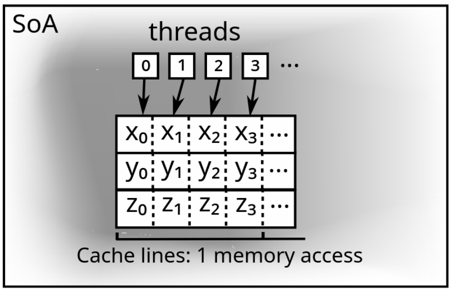

Figure 3. Example of an SoA access pattern. Adapted from [15].

4 Performance Studies

Initial performance studies of the GPU-enabled algorithm were conducted to evaluate its performance and to compare with the current utilized (legacy) CPU based algorithm. The first check involved examining the effect of using the parametrized magnetic field to propagate the helix trajectories through the tracker volume up to the ECAL surface. Figure 5 shows a comparison of kinematic and angular variables for reconstructed electrons matched with generator-level electrons, both using the legacy CPU approach, with the full magnetic field or the parametrized magnetic field assumption. The results demonstrate excellent agreement. Additionally, in Fig. 6 a comparison between the legacy CPU approach and the new GPU-enabled algorithm for the same set of variables is shown, again showcasing a good agreement

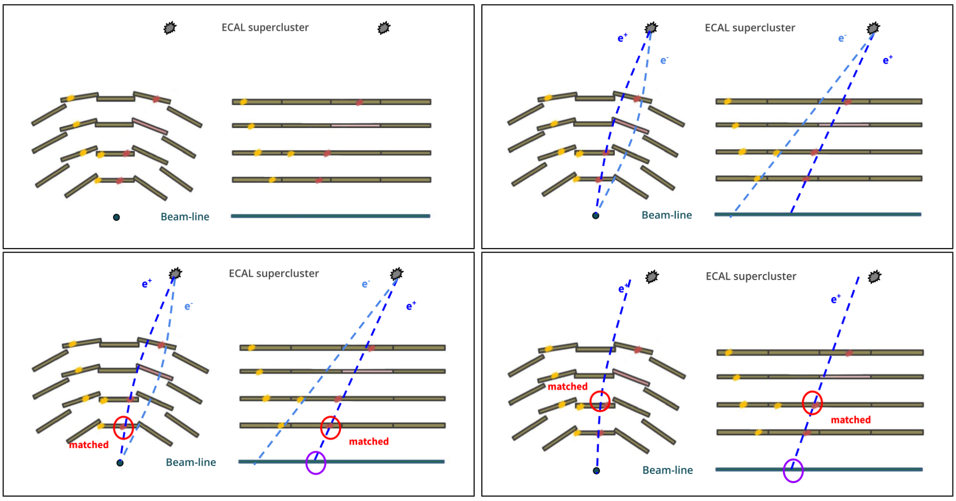

Figure 4. Schematic of the electron seeding process. We start by considering a single SC and a tracker seed (top left). Hypothetical electron trajectories are extrapolated from the supercluster (SC) toward the beamline under both charge hypotheses (top right). A geometric matching check determines if the trajectory aligns with a hit of the tracker seed (bottom left). If the hit is matched, a more precise z-coordinate for the PV is derived and a track is propagated from the PV towards the second hit of the seed (bottom right). If all seed hits are successfully matched, the seed is accepted for the next stage of electron reconstruction.

with the legacy implementation. The primary differences arise from varying quality cuts and some refinements that are still being fine-tuned. Moving forward, more thorough validation of the physics performance is expected to be performed.

5 Conclusion and Future Work

This document summarizes the ongoing development of a GPU-enabled electron seeding algorithm for the CMS experiment. The initial stages focused on redesigning the algorithm to expose parallelism and adapting the associated data structures and utility functions for GPU compatibility. By leveraging GPU parallelism and utilizing the Alpaka library for cross-platform portability, we anticipate significant improvements in both computational efficiency and software flexibility. Early performance studies have shown promising results, with the new GPU-enabled algorithm demonstrating strong agreement with the legacy CPU-based approach. Future work will focus on further optimizing the algorithm, exploring additional parallelization strategies, validating its physics performance, and integrating it into the full CMS reconstruction pipeline. These advancements are key to meeting the computational challenges of the HL-LHC era and will lay the groundwork for further GPU-driven improvements in CMS reconstruction.

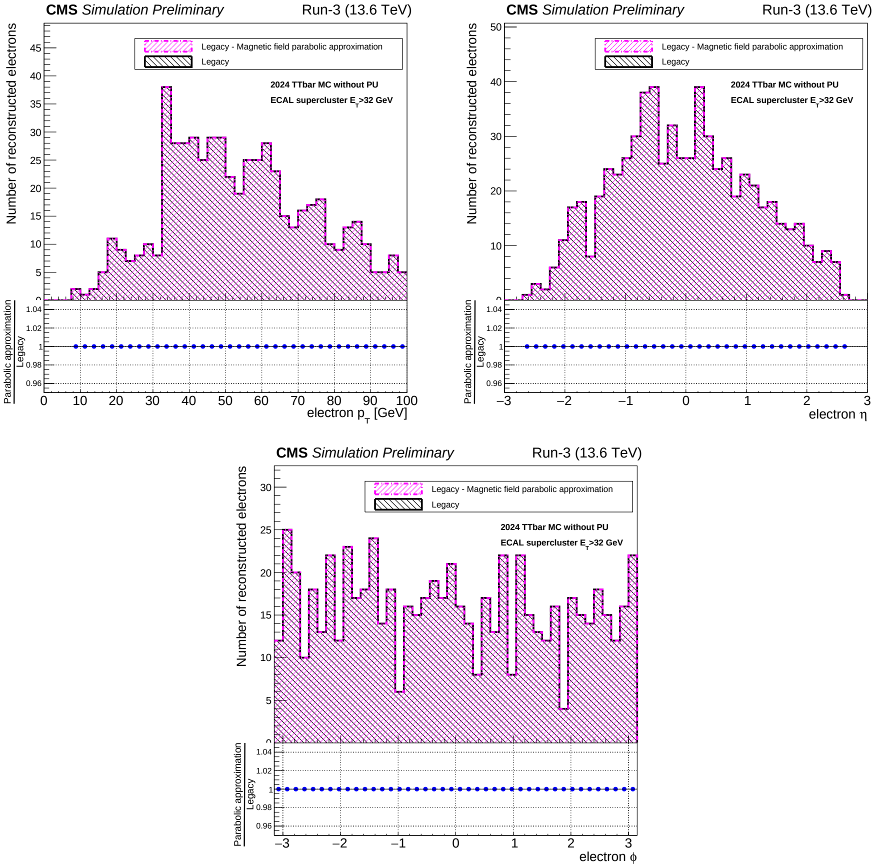

Figure 5. The number of reconstructed electrons matched to a generator level electron is shown as a function of the electron transverse momentum p_T (the top left), pseudorapidity η (top right) and azimuthal angle (bottom) considering the current (legacy) pixel matching algorithm with the legacy approach for the magnetic field (black line) and considering the current (legacy) pixel matching algorithm with the simplified approach for the magnetic field (magenta line). The ratio between the distributions obtained with the legacy approach and the simplified approach for the magnetic field is displayed in the lower panels and shows excellent agreement. The results have been produced using 1800 simulated pair-produced top quark events, where both top quarks decay into leptons, generated without additional overlapping interactions (pile-up). Taken from Ref. [16].

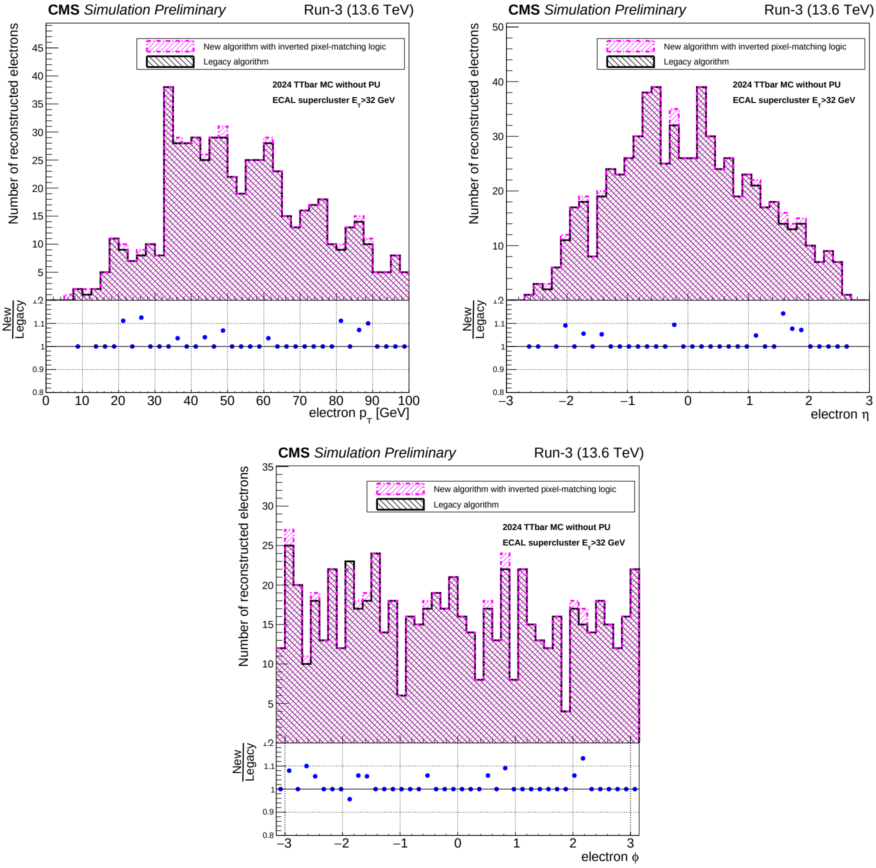

Figure 6. The number of reconstructed electrons matched to a generator level electron is shown as a function of the electron transverse momentum p_T (the top left), pseudorapidity η (top right) and azimuthal angle (bottom) considering the current (legacy) pixel matching algorithm (black line) and the new pixel matching algorithm with the inverted matching logic, using the simplified approach for the magnetic field (magenta line). The ratio between the distributions obtained with the legacy and the new algorithm is displayed in the lower panels and shows good agreement. Slight differences in the number of reconstructed electrons can be attributed to some quality cuts that are not applied in the new algorithm and the different structure of the electron seed and ECAL supercluster association map. The results have been produced using 1800 simulated pair-produced top quark events, where both top quarks decay into leptons, generated without additional overlapping interactions (pile-up). Taken from Ref. [16].

References

[1] CERN, *High Luminosity LHC*, https://hilumilhc.web.cern.ch/

[2] CMS Collaboration, "The CMS experiment at the CERN LHC," *JINST* **3** (2008) S08004,

[3] CMS Collaboration, "Development of the CMS detector for the CERN LHC Run 3," *JINST* **19** (2024) P05064, .

[4] CMS Collaboration, "The Phase-2 Upgrade of the CMS Tracker," CERN-LHCC-2017-009, CMS-TDR-014, https://cds.cern.ch/record/2272264/, .

[5] CMS Collaboration, "A MIP Timing Detector for the CMS Phase-2 Upgrade," CERN-LHCC-2019-003, CMS-TDR-020, https://cds.cern.ch/record/2667167.

[6] CMS Collaboration, "The Phase-2 Upgrade of the CMS Barrel Calorimeters," CERN-LHCC-2017-011, CMS-TDR-015, https://cds.cern.ch/record/2283187.

[7] CMS Collaboration, "The Phase-2 Upgrade of the CMS Endcap Calorimeter," CERN-LHCC-2017-023, CMS-TDR-019, https://cds.cern.ch/record/2293646, .

[8] CMS Collaboration, "The Phase-2 Upgrade of the CMS Muon Detectors," CERN-LHCC-2017-012, CMS-TDR-016, https://cds.cern.ch/record/2283189.

[9] CMS Collaboration, "The Phase-2 Upgrade of the CMS Beam Radiation Instrumentation and Luminosity Detectors," CERN-LHCC-2021-008, CMS-TDR-023, http://cds.cern.ch/record/2759074.

[10] CMS Collaboration, "The Phase-2 Upgrade of the CMS Level-1 Trigger," CERN-LHCC-2020-004, CMS-TDR-021, https://cds.cern.ch/record/2714892.

[11] CMS Collaboration, "The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger," CERN-LHCC-2021-007, CMS-TDR-022, https://cds.cern.ch/record/2759072.

[12] CMS Collaboration, "Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC," JINST **16** (2021) P05014, doi:10.1088/1748-0221/16/05/P05014, [arXiv:2012.06888 [hep-ex]].

[13] CMS Collaboration, "2024 HLT timing and throughput results," 2024, http://cds.cern.ch/record/2914421.

[14] CMS Collaboration, "Commissioning CMS online reconstruction with GPUs," CMS-DP-2023-004, https://cds.cern.ch/record/2851656.

[15] Eric Cano, *Implementation of generic SoA data structure in the CMS software*, CERN, CMS-CR-2023-040, Geneva, 2023. Available at: http://cds.cern.ch/record/2872252.

[16] CMS Collaboration, *Towards a GPU compatible electron seeding algorithm*, 2024, http://cds.cern.ch/record/2914420.

[17] CMS Offline Software and Computing, *CMS Phase-2 Computing Model: Update Document*, CERN, Geneva, 2022, https://cds.cern.ch/record/2815292.