Particle Flow reconstruction in the CMS Level-1 Trigger for the HL-LHC

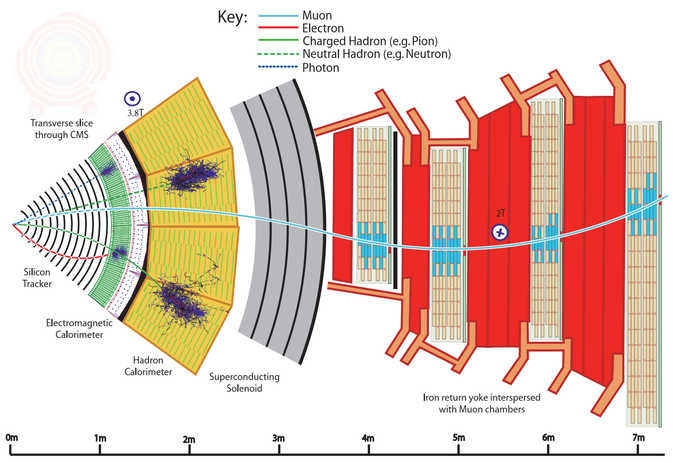
Giovanni Petrucciani (CERN) on the behalf of the CMS Trigger Project

Introduction to Particle Flow

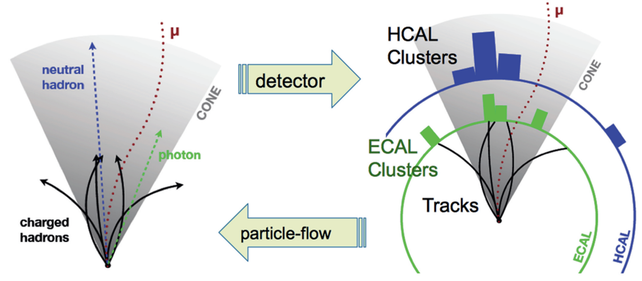
The **Particle Flow (PF)** approach aims to **reconstruct and identify individually all particles** produced in CMS by c**ombining information from all subdetectors**.[1]

Since LHC Run 1, PF has been widely used in CMS offline and High Level Trigger (HLT) event reconstruction. Areas where benefits from PF were most evident include:

- **Jet and p_T^{miss} performance**, especially at low p_T's relevant for e.g. top quark physics, ttH, compressed supersymmetry, …
- **τ_h reconstruction and identification**
- as **input to pileup mitigation strategies**, e.g. per-particle pileup identification[2] (PUPPI)

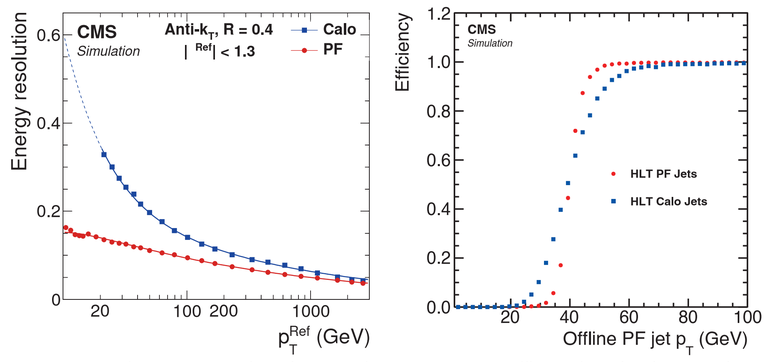


A sketch of the specific particle interactions in a transverse slice of the CMS detector [1]



Two necessary ingredients for PF are efficient charged particle reconstruction in the inner tracker, and fine granularity calorimetry to resolve the contributions from neighbouring particles.

Neither is available in the current Level-1 Trigger system[3], limited to coarse granularity calorimetric information ($\Delta\eta \times \Delta\phi \sim 0.1 \times 0.1$), and reconstruction in the muon spectrometer.

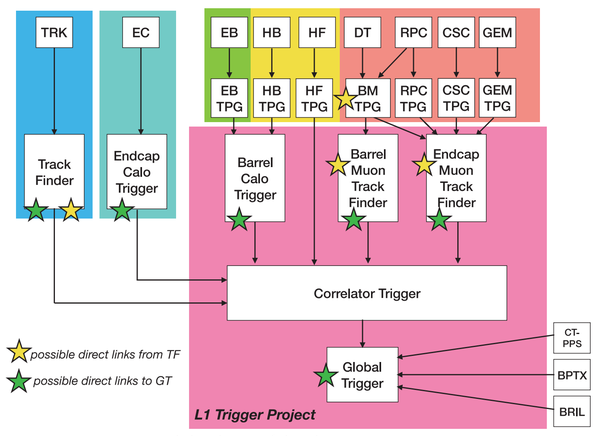


Improvement from PF compared to calorimeters for jet reconstruction offline (Left) and at the HLT (right). Performance is shown on simulated events, with no pile-up. [1]

CMS Detector Phase 2 upgrade

The **CMS detector upgrade** for HL-LHC[4] will bring key improvements that **will make PF possible at the L1 Trigger** for the first time:

- New **inner tracker** supporting **40 MHz readout** of *stubs*, pairs of correlated hits in consecutive layers from high p_T particles, and a backend track finder system to **reconstruct all tracks with p_T > 2 GeV, $|\eta|$ < 2.4** within a latency of 4 μs.
- Updated backend for the **electromagnetic barrel calorimeter** (EB) supporting **full granularity readout at L1** ($\Delta\eta \times \Delta\phi \sim 0.02 \times 0.02$)
- **New endcap calorimeter** (EC) detector with high granularity
- New L1 trigger system[5] with improved processing power, a **global correlator layer**, and deeper buffers in all subsystems to allow a total latency up to 12.5 μs



High-level view of the Phase-2 L1 trigger [5]

Architecture and Constraints

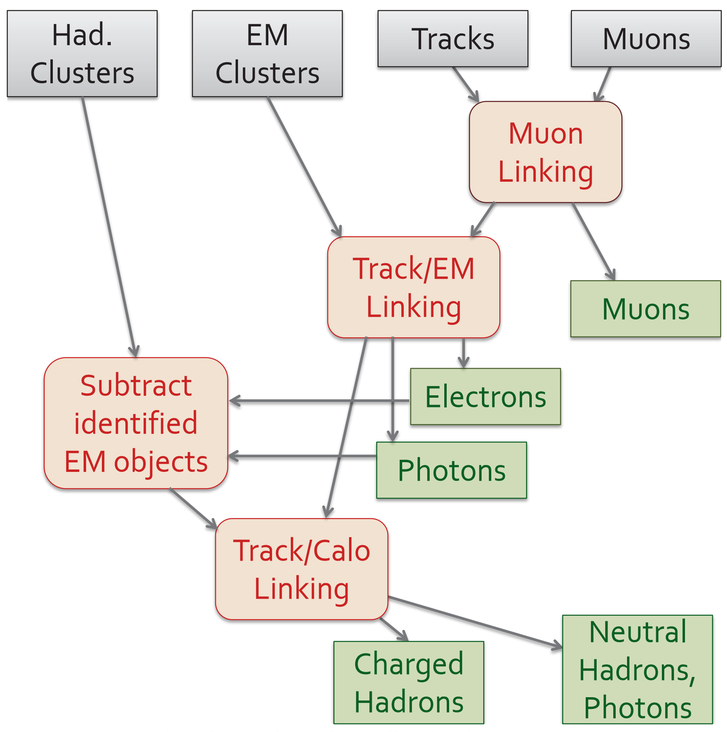
The PF algorithm was re-designed from first principles, as the algorithm used offline and at HLT was not suitable to the L1 environment

- The L1 PF algorithm must process events at the **40 MHz** collision rate, and output reconstructed particles after a **fixed latency ≤ 1μs**.
  - For comparison, the current HLT PF algorithm is run at a rate of ~20 kHz, and has a latency of ~100 ms
- The algorithm has to run on **FPGA** hardware instead of CPUs
  - FPGAs provide a **very large number of simple processing components that all operate in parallel**, but to operate efficienctly the data flow and tasks have to be statically defined.

L1 PF Inputs

A first prototype L1 PF algorithm has been developed using:

- **Tracks** from an early implementation of the L1 Track Finder
- **Electromagnetic clusters** with fine granularity from EB and the electromagnetic layers of EC, to allow reconstruction and identification of non-isolated photons and electrons, e.g. from π^0 decays in jets and conversions
- **Hadronic clusters** from EB+HB and the full EC, with coarser granularity and larger size to ensure showers from individual hadrons are contained in a single cluster.
- **Muons** from the current Phase 1 muon system trigger



Block diagram of the L1 Particle Flow algorithm prototype

Prototype algorithm

- For each input muon the best-matching track in the inner tracker by $\Delta R^2 = \Delta\eta^2 + \Delta\phi^2$ and p_T, is tagged as a muon and excluded from further processing in the PF algorithm.
- Each track is linked to the closest electromagnetic cluster (EM), if any is found within ΔR < 0.04. For all clusters, the sum p_T of all associated tracks is computed and compared to the EM energy.
  - Clusters with no associated tracks are tagged as photons
  - If $p_T^{cluster} \geq \sum p_T^{track}$ within uncertainties, tracks are tagged as electrons, and any significant p_T excess is promoted as photon
  - Otherwise, the cluster is discarded (it is likely of hadronic origin)
- Surviving EM clusters are linked to the closest hadronic cluster.
  - The energy of the hadronic clusters is updated subtracting that of all linked EM clusters, and the cluster is discarded if it no significant energy is left
- Tracks are linked to the closest hadronic cluster in ΔR^2 and p_T.
  - $p_T^{calo} \geq p_T^{track} - 2\,\sigma(p_T^{calo})$ is required to reject fake high p_T tracks
  - Tracks are promoted to charged hadrons if linked to a cluster, or if p_T^{track} < 10/20 GeV if the track passes loose/tight quality criteria
  - If p_T^{calo} significantly exceeds $\sum p_T^{track}$, the excess is promoted to a neutral hadron, or a photon if the excess is mainly electromagnetic. Otherwise, the cluster is discarded.
- Calorimeter clusters outside the $|\eta|$ coverage of the L1 track finder are also promoted to hadrons or photons

Firmware implementation

- Xilinx Vivado **High Level Synthesis**[6] (HLS) is used to compile the PF algorithm, coded in a subset of **C++ with annotations**, into a reusable firmware block (*IP core*).
- To best profit from the FPGA capabilities, all the computation is implemented using **integers** instead of floating-point, and the **mathematics is kept simple**: mainly additions and comparisons, few multiplications, only one division (implemented as a lookup table), and no square roots or special functions.
- To increase throughput the entire algorithm is **pipelined** to accept new inputs every 1 or 2 clock cycles. All combinatorical **loops**, e.g. on object pairs in the linking, are **unrolled** to compute all values in parallel, which also reduces the latency.
- The **PF relies only on local infomation**, so different detector regions can be processed independently. The complexity and FPGA resource use depend on the maximum allowed number of input objects, determined by the size of the detector region.
- Preliminary estimates show that 4 regions of size $\Delta\eta \times \Delta\phi \sim 0.62$ with 25 tracks and 20 clusters can be processed every 25 ns on a Xilinx **VU9P** FPGA[7] with **~40% resource** usage and **~0.5 μs latency**, in line with the requirements.

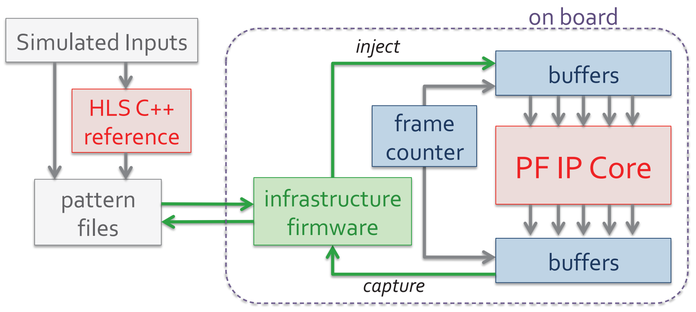
Test setups

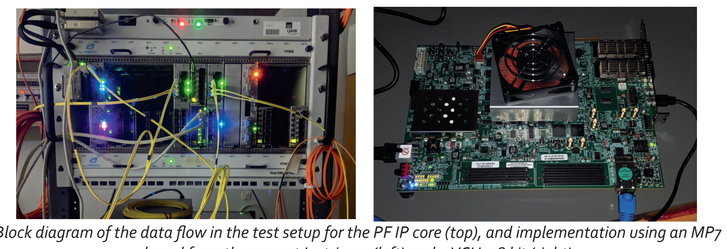
- The PF IP core has been successfully **tested on current and early prototype trigger boards** based on Virtex-7 FPGAs (for reduced object multiplicities), and on the **VU9P VCU118 development kit** and on Amazon AWS.
- VHDL or C++ code is used to interface the core with the board infrastructure using IPbus[8] or AXI-PCIe to **inject input patterns** from CMS detector simulation into the core, and the **output is checked for bitwise identity** with the expections from HLS.

Pileup rejection

PF reconstructs all particles, also from pile-up interactions. To isolate the leading interaction, once the **primary vertex** (PV) is identified the **Pile Up Per Particle Identification (PUPPI)** algorithm is used:

- Charged hadrons are selected if the track coordinate along the beam line is compatible with the PV
- For all neutral hadrons, the probability *w* for the particle to belong to the PV is estimated as function of $\sum p_T^2/\Delta R^2$, computed from all selected charged hadrons within ΔR < 0.4, and the particle momenta is scaled by that probability.
  - Particles with *w* < 1% or whose scaled pT is below a threshold dependent on the pile-up level are discarded.
  - Outside the tracker coverage, the summation is performed using all particles, and higher thresholds are used.





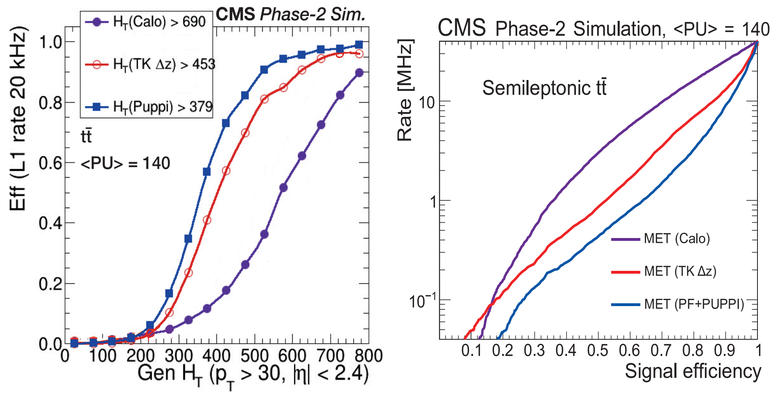
Block diagram of the data flow in the test setup for the PF IP core (top), and implementation using an MP7 board from the current L1 trigger (left) and a VCU118 kit (right)

Physics Performance

The performance of the L1 PF + PUPPI algorithm for physics objects has been evaluated using using simulated events for the CMS Phase 2 detector and HL-LHC pile-up conditions.

- A benchmark L1 trigger for **hadronic signatures** is defined based on $\mathbf{H_T}$, the scalar sum of calibrated p_T of jets with p_T > 30 GeV, $|\eta|$ < 2.4, build from calorimeters clusters, tracks associated to the PV, or PF+PUPPI particles. For a fixed L1 accept rate of 20 kHz, **PF+PUPPI inputs allow a lower threshold and a sharper the turn-on curve** of the efficiency as function of the true H_T in simulated tt events.
- For **missing energy signatures**, p_T^{miss} computed from PF+PUPPI particles is likewise found to yield a better trade-off between L1 rate and signal efficiency (on tt events with true p_T^{miss} > 100 GeV)

Applications of PF reconstruction to other physics objects and integration in the L1 trigger menu studies for Phase 2 is ongoing.



Performance of L1 reconstruction using PF+PUPPI particles compared to calorimeters clusters or L1 tracks from associated to the PV, for H_T (left) and p_T^{miss} (right) [5]

References

[1] CMS Collaboration, "*Particle-flow reconstruction and global event description with the CMS detector*", JINST 12 (2017) no.10, P10003, arXiv:1706.04965
[2] D. Bertolini et al., "*Pileup Per Particle Identification*", JHEP 1410 (2014) 059, arXiv:1407.6013
[3] CMS Collaboration, "*CMS Technical Design Report for the Level-1 Trigger Upgrade*", CERN-LHCC-2013-011, https://cds.cern.ch/record/1556311
[4] CMS Collaboration, "*Technical proposal for the Phase-II Upgrade of the Compact Muon Solenoid*", CERN-LHCC-2015-010, https://cds.cern.ch/record/2020886
[5] CMS Collaboration, "*The Phase-2 Upgrade of the CMS Level-1 Trigger - Interim Technical Design Report*", CERN-LHCC-2017-013, https://cds.cern.ch/record/2283192
[6] Xilinx, "*Vivado High-Level Synthesis*", https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html
[7] Xilinx, "*Ultrascale+ FPGA product table*", https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale-plus.html#productTable
[8] C. Ghabrous Larrea et al., *"IPbus: a flexible Ethernet-based control system for xTCA hardware"*, JINST 10 (2015) no.02, C02019, https://ipbus.web.cern.ch/ipbus/