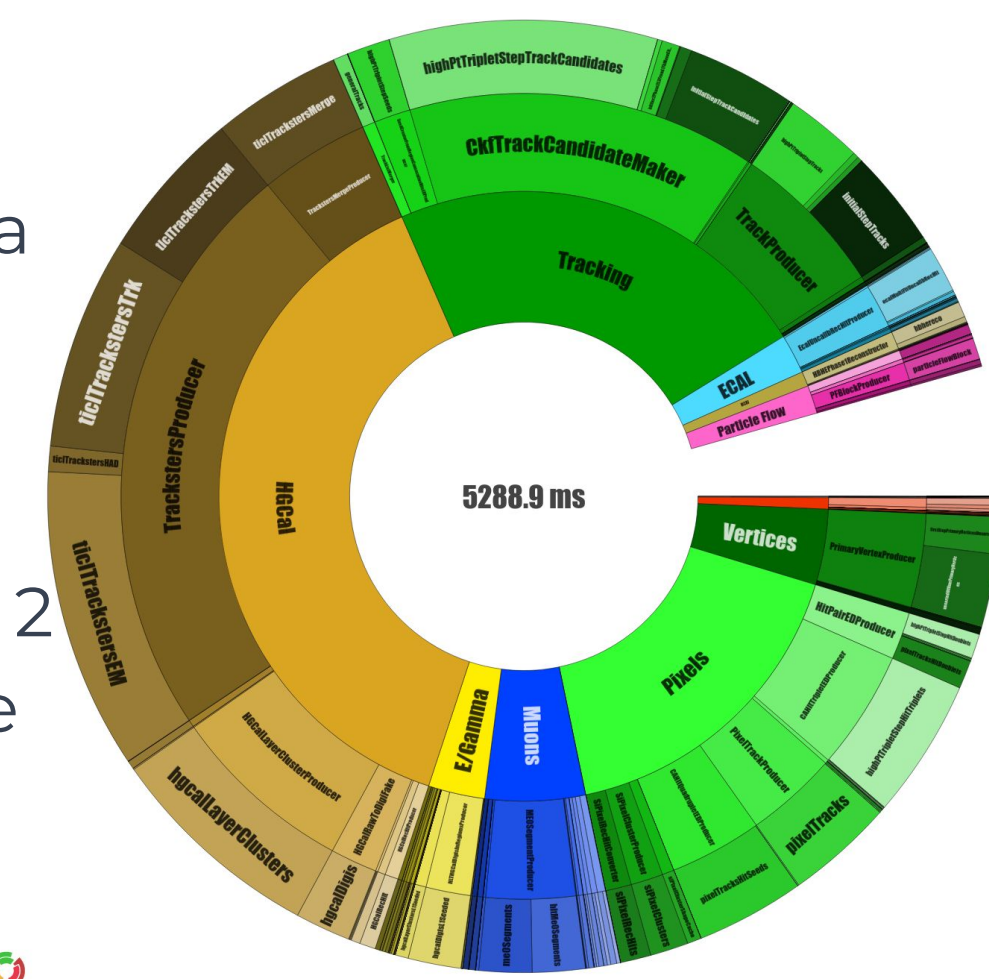
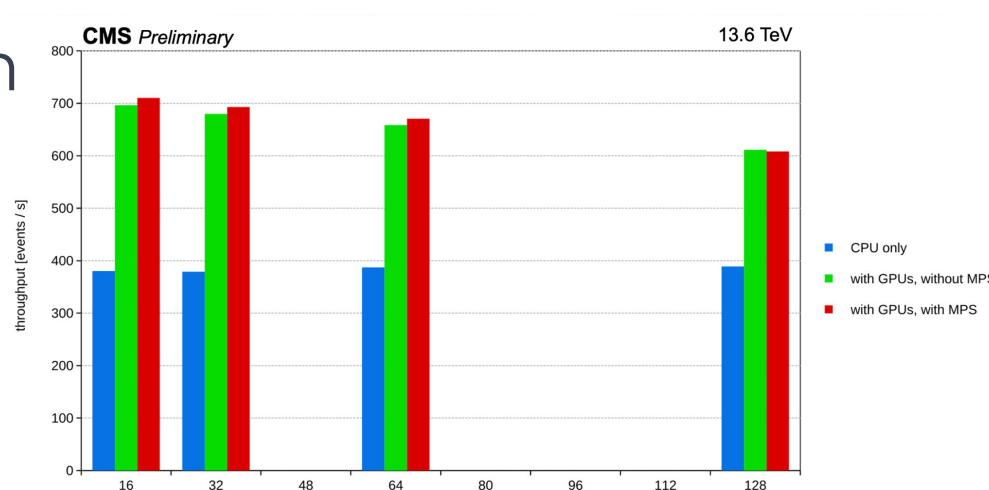
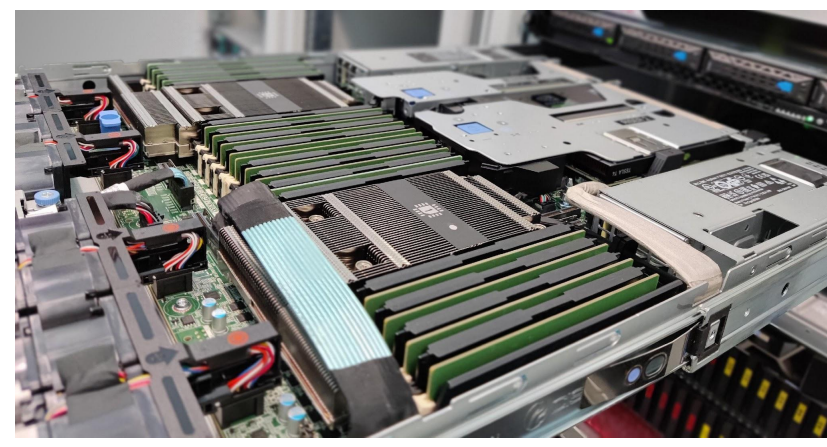


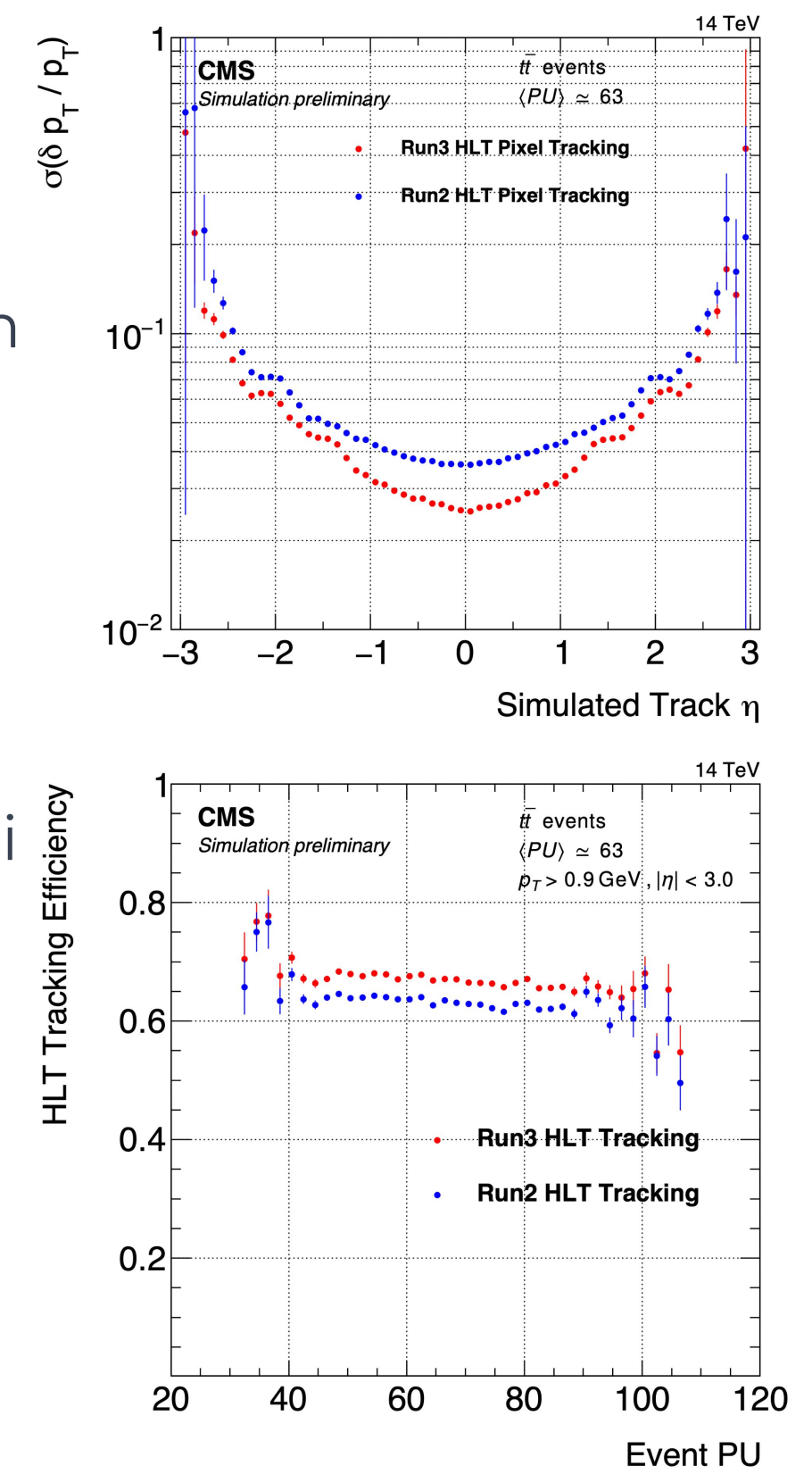
L'High Level Trigger (HLT) di CMS

- L'High Level Trigger di CMS è un sistema usato per filtrare e selezionare in tempo reale i numerosi eventi che vengono prodotti dalle collisioni ad alta energia dal Large Hadron Collider e misurati dai diversi sotto detector di CMS.
- Il rivelatore produce dati circa 40 milioni di volte al secondo, che è troppo alto per essere memorizzato considerando il peso di ~ 1 MB per evento.
- Un primo filtro degli eventi prodotti dal rivelatore è il L1 Trigger che riduce il rate da 40 MHz a circa 100 kHz a Run 3
- L'HLT è il secondo filtro e utilizza algoritmi sofisticati per analizzare i dati e selezionare rapidamente gli eventi più interessanti.
- Selezionando solo gli eventi più interessanti, il sistema HLT riduce la quantità di dati che devono essere memorizzati ed analizzati di circa 1000 volte. Ciò consente ai fisici di studiare fenomeni di fisica rari e di cercare nuove particelle che potrebbero aiutare a spiegare i misteri dell'universo.
- La versione rinnovata per Run3 dell'HLT farm è composta da 200 macchine, ognuna dotata di 2 socket utilizzati da AMD EPYC 7763 "Milan" (processore a 64 core con 128 thread hardware) e **2 GPU Nvidia T4**.
- L'upgrade di HLT [1] è stato fatto anche in vista della Fase 2 di LHC che prevederà un aumento di collisioni istantanee fino a 200 (per Run 3 se ne hanno 60)
- In figura sono riportati il throughput a Run 3 con e senza GPU e le misure temporali che ci si aspetta ad HLT per l'upgrade di Fase 2



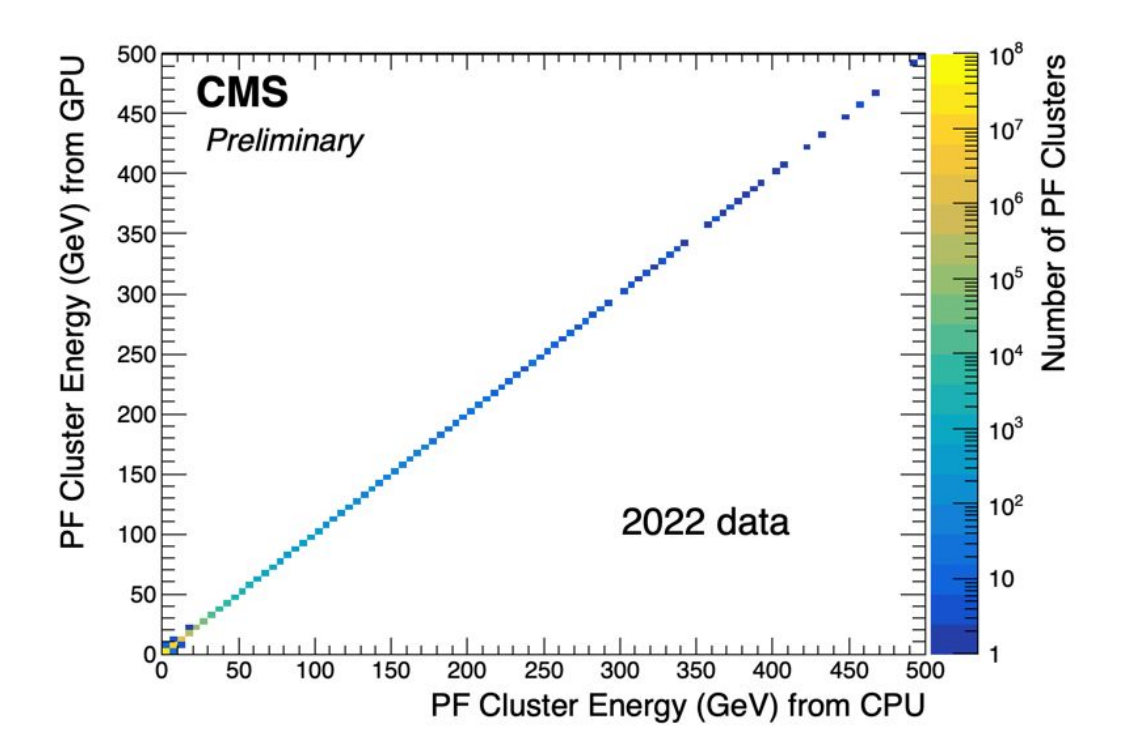
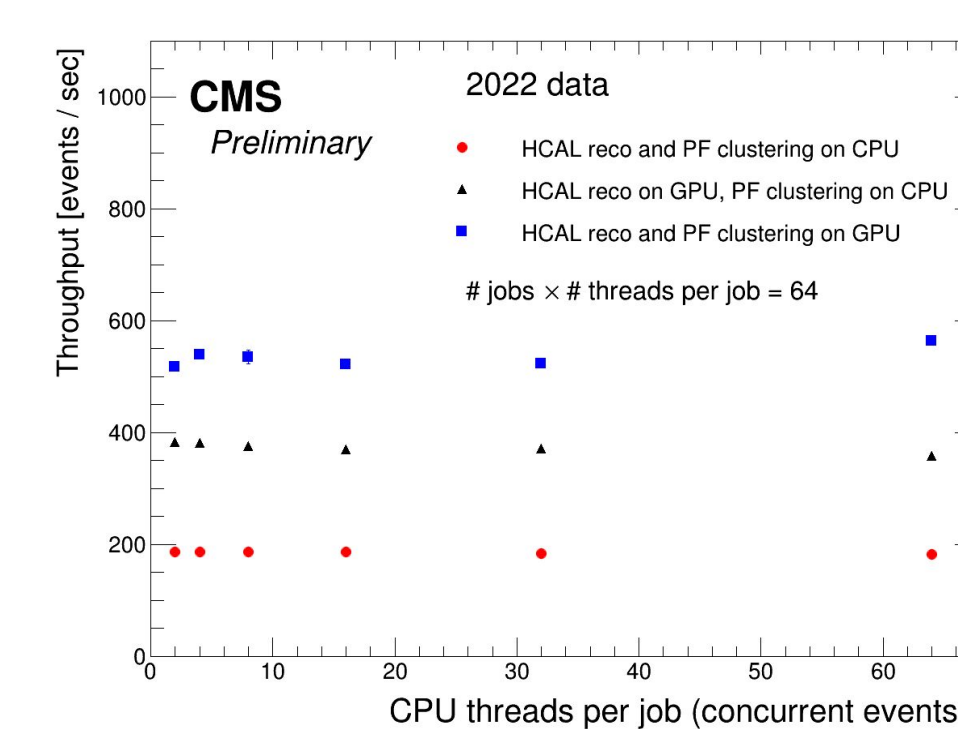
Ricostruzione delle tracce e vertici con Pixel ad HLT per Run 3

- CMS ha implementato una ricostruzione completa delle tracce pixel su GPU [2], implementando nuovi algoritmi e tecniche di fitting
- L'approccio di tracking iterativo utilizzato in Run 2 con 3 (o 4 per il 2018) iterazioni è stato sostituito da un approccio a singola iterazione in cui vengono ricostruiti gli ntupletti (tripletti e quadrupletti)
- La singola iterazione è fatta a partire da tracce pixel Patatrack con almeno tre pixel hit e $p_T > 0.3$ GeV.
- I vertici primari sono ricostruiti ad HLT come cluster di tracce di pixel aventi il punto d'impatto sull'asse del fascio, utilizzando solo la coordinata z.
- A Run 3 vengono considerate solo le tracce di pixel con almeno 4 hit di pixel e $p_T > 0.5$ GeV per la ricostruzione dei vertici.



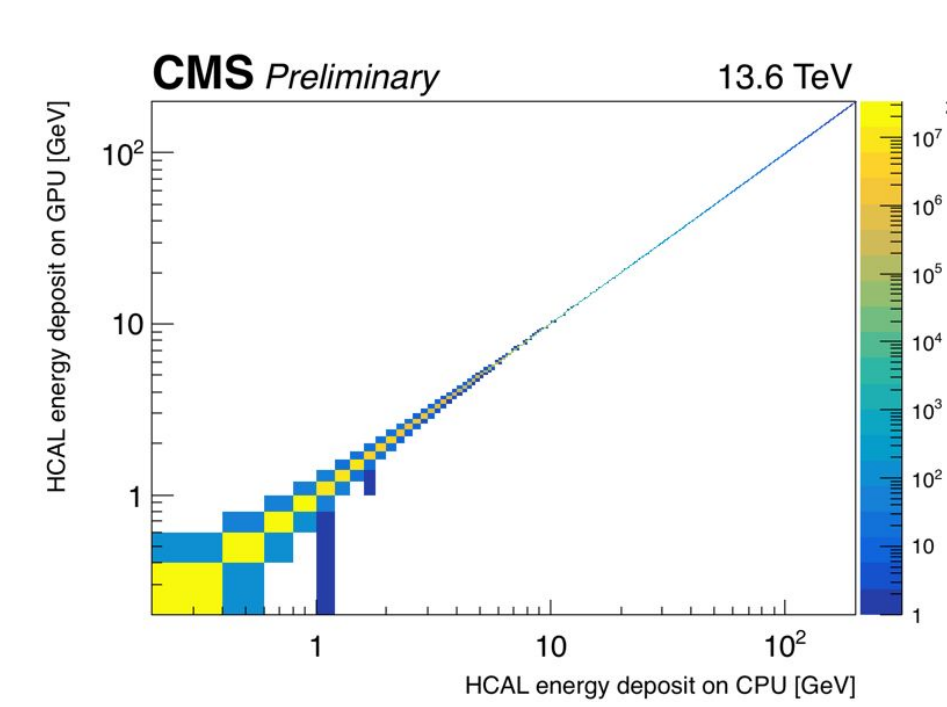
Particle Flow per HCAL su GPU

- Il metodo **Particle Flow** (PF) esegue la ricostruzione globale dell'evento utilizzando i dati provenienti da tutti i sotto-rivelatori di CMS per generare candidati di particelle nello stato finale per ciascun evento
- È stato esaminato l'uso della computazione eterogenea con l'offloading di alcuni calcoli alle GPU [4] tramite il linguaggio di programmazione CUDA per accelerare il PF per la raccolta dati di Run 3.
- Il PF richiede due tipi di input: le tracce cariche e i loro rispettivi hit nel sistema muonico che vengono utilizzati per formare tracce PF, e gli hit dei calorimetri (rechits) vengono utilizzati per formare PF rechits e quindi cluster PF
- Il PF collega le tracce alle corrispondenti shower nei calorimetri (sotto forma di cluster PF)
- La produzione di rechits PF, il seeding dei cluster e la produzione di cluster PF sono stati implementati in CUDA vista il loro tempo di esecuzione su CPU che occupa circa la metà del tempo di elaborazione e pochi punti percentuali del tempo di elaborazione totale di HLT.
- Poiché la ricostruzione locale di ECAL e HCAL è già eseguita su GPU a HLT, la produzione di rechits PF prende in input gli hit che sono già sulla GPU.



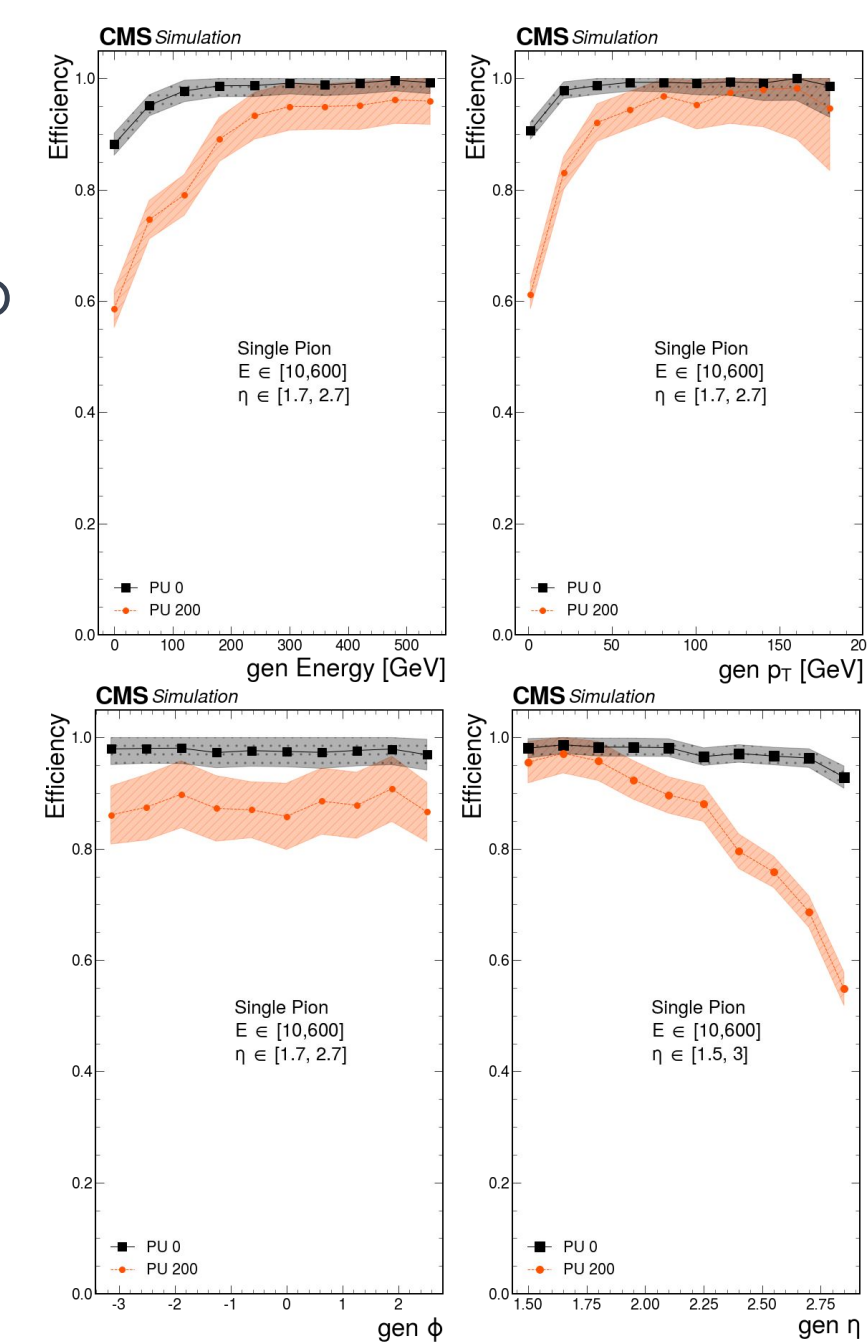
Ricostruzione Locale di ECAL ed HCAL ad HLT per Run 3

La ricostruzione del calorimetro elettromagnetico (ECAL) e del calorimetro adronico (HCAL) è stata riscritta in CUDA [3] per essere eseguita su GPU, rendendo la ricostruzione del deposito di energia di un singolo canale, indipendente dai canali vicini. L'algoritmo di ricostruzione si basa sulla tecnica dei "Fast Non-negative Least Square" per stimare il deposito energetico nel singolo canale. Le differenze nelle validazioni di fisica sono trascurabili e dovute al calcolo con float su GPU invece che con double.



Ricostruzione di HGCal con TICL

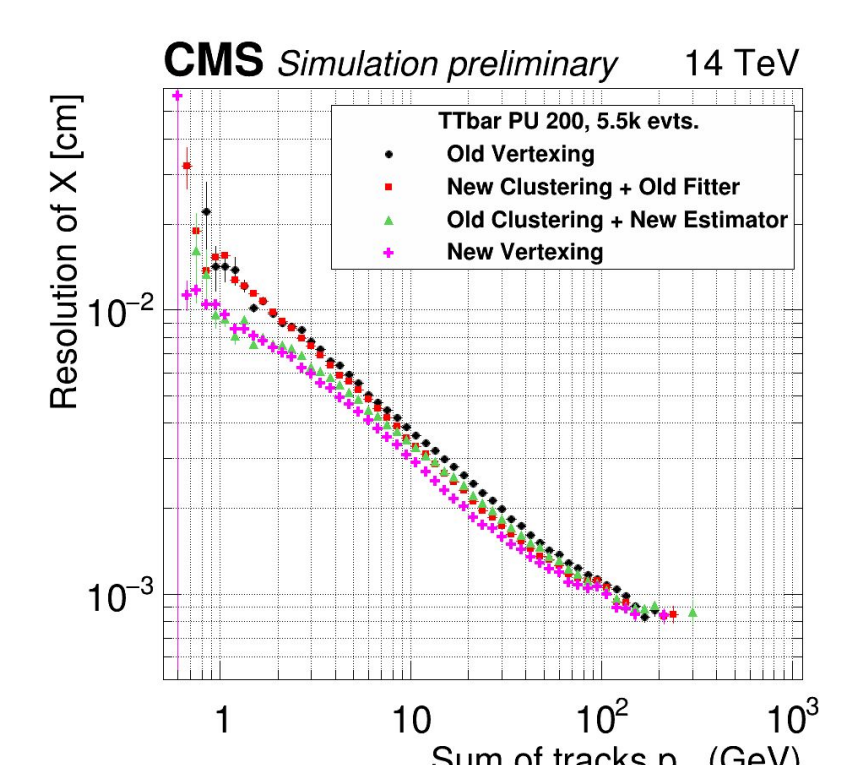
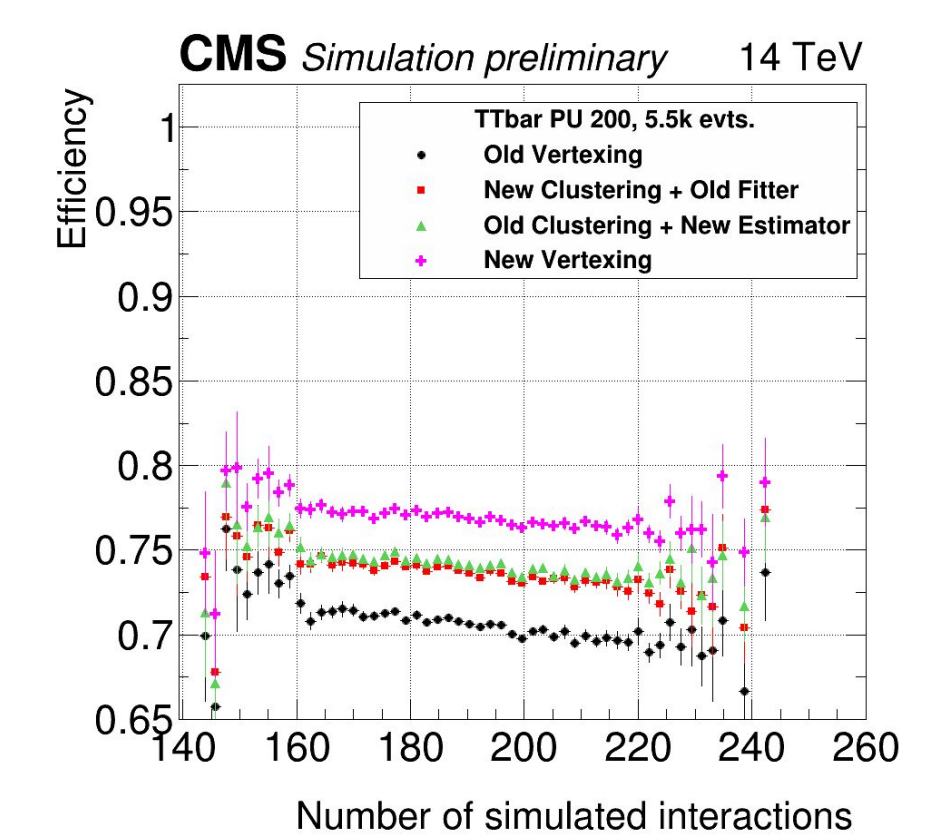
- Il nuovo calorimetro per gli endcap della Fase 2, **HGCal**, avrà un'alta granularità per mitigare gli effetti di pileup
- Un nuovo framework di ricostruzione per HGCal è stato sviluppato [5], TICL (The Iterative CLustering), per poter sfruttare a pieno l'alta granularità.
- TICL è un potenziale candidato per sostituire il PF, essendo progettato per il computing eterogeneo
- L'algoritmo di clustering all'interno dei layer di HGCal è CLUE, il quale effettua un clustering in base alla densità di energia, ricostruendo cluster di rechits \rightarrow Layer Clusters
- Per connettere i cluster 2D, un algoritmo di pattern recognition, CLUE-3D, ha il compito di ricostruire gli oggetti tridimensionali (le shower di particelle) chiamati tracksters
- Linkando i diversi trackster originati dalla stessa particella e gli stessi con le opportune tracce è possibile identificare la particella e ricostruire l'energia rilasciata fornendo come informazione finale un TICLCandidate



Vertexing su GPU

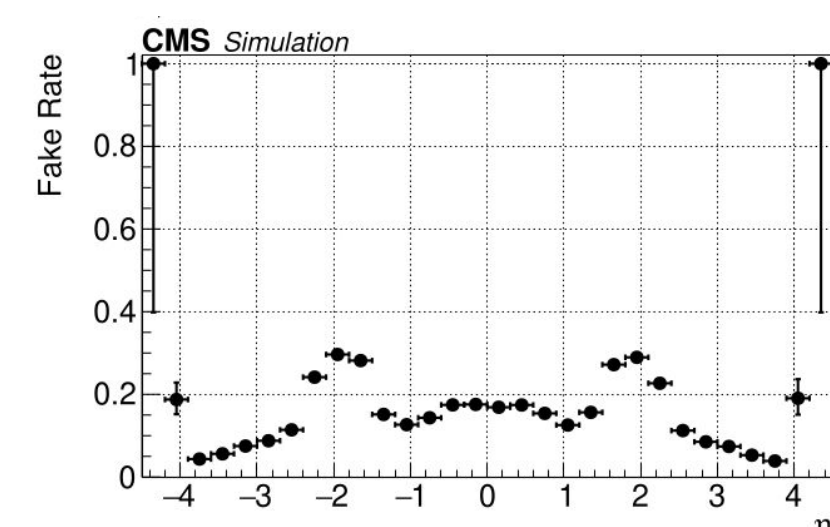
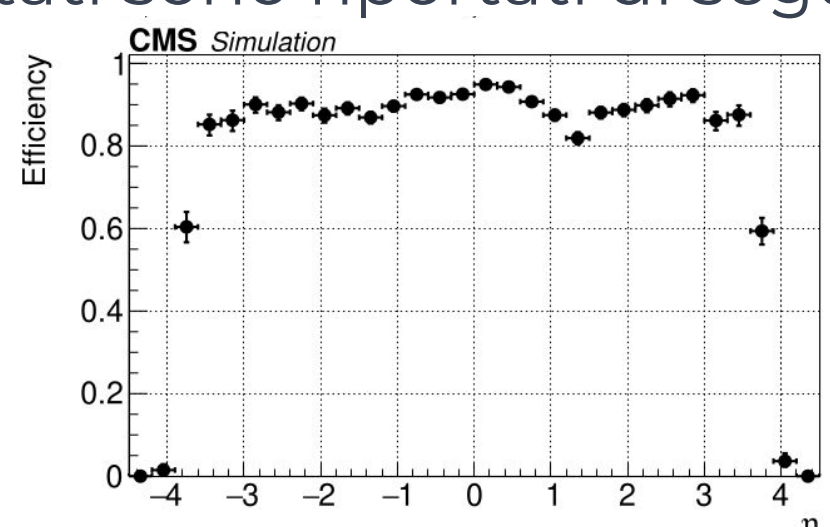
La ricostruzione dei vertici è una procedura svolta in due passaggi a CMS: la ricerca dei vertici e il fitting. Durante Run 2 e Run 3 CMS ha rispettivamente utilizzato il Deterministic Annealing e l'Adaptive Vertex Fitteri. Dato l'elevato impatto sul tempo di ricostruzione di questa procedura (fino al 10% del tempo totale con le condizioni HL-LHC), sono stati sviluppati nuovi algoritmi per i due passaggi [6]:

- il nuovo procedimento di clustering ordina le tracce nella coordinata z, le divide in blocchi di uguale dimensione con una frazione di sovrapposizione fissa tra i blocchi e esegue in modo indipendente il "Deterministic Annealing" su tutti i blocchi;
 - il nuovo fitter stima in modo iterativo le coordinate 3D del vertice e gli errori utilizzando la media pesata del punto di impatto delle tracce nella posizione del fascio e l'incertezza. Le iterazioni includono la rimozione degli outlier per migliorare le prestazioni.
- Le prestazioni fisiche, riportate a destra, sono state ottenute con un aumento del throughput di un fattore 6 per la combinazione dei due nuovi algoritmi.



Line Segment Tracking

La tecnica di Line Segment Tracking (**LST**) [7] costruisce le tracce combinando l'informazione dell'**Outer Tracker** (OT, dove $|\eta| < 2.5$) sfruttando i nuovi "moduli pT" a doppio layer e le informazioni dell'Inner Tracker (IT, dove $|\eta| < 4$). I Mini-Doublets (MD) vengono costruiti in ogni layer per tracce con $p_T > 0.8$ GeV e vengono utilizzati in segmenti di linea che collegano MD compatibili utilizzando una mappa predefinita basata sulla geometria del rivelatore. I segmenti di linea vengono quindi utilizzati per costruire le tracce. I risultati sono riportati di seguito:



Conclusioni

- L'aggiornamento della farm HLT di CMS con le GPU ha aumentato l'interesse verso nuovi algoritmi e l'implementazione di versioni intrinsecamente parallele dei precedenti algoritmi di ricostruzione, mantenendo le stesse prestazioni di fisica ma riducendo drasticamente il tempo di calcolo
- Con le tracce dei Pixel e la ricostruzione locale di ECAL e di HCAL a HLT, il throughput è già stato aumentato per Run 3 di un fattore 1.7x
- Dati i risultati ottenuti con lo sviluppo delle GPU per Run 3, molte indagini per la Fase II sono attualmente in corso, concentrandosi principalmente su HGCal, MTD e il tracking.
- Gli sviluppi software ottenuti per HLT beneficeranno anche la ricostruzione che viene fatta offline, una volta che saranno stati trasposti i nuovi algoritmi
- Vista la recente integrazione nel software di CMS della portability library **Alpaka** che permette di sviluppare un unico codice e di compilare lo stesso per diverse architetture eterogenee, sarà necessario reimplementare gli algoritmi scritti in CUDA di HLT con questa nuova libreria

Bibliografia

- Referenze:
- [1] [The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger](#), CMS Collaboration
 - [2] [Heterogeneous reconstruction of tracks and primary vertices with the CMS pixel tracker](#), A. Bocci et al.
 - [3] [Heterogeneous computing for the local reconstruction algorithms of the CMS calorimeters](#), CMS Collaboration
 - [4] [Heterogeneous Particle Flow Algorithms for Event Reconstruction in CMS](#), F. Pantaleo
 - [5] [The Iterative Clustering framework for the CMS HGCal Reconstruction](#), CMS Collaboration
 - [6] [Offline Primary Vertex Reconstruction for Heterogeneous Architectures](#), CMS Collaboration
 - [7] [Line Segment Tracking in the HL-LHC](#), G. Niendorf et al.