PATATRACK
Heterogeneous events selection
at the CMS Experiment

Felice Pantaleo
CERN
Experimental Physics department
felice@cern.ch
Is there a place for GPUs in all this?

• At trigger level:
  – Controlled environment
  – High throughput density required

• On the WLCG:
  – Software running on very different/diverse hardware
    • Starting from Pentium 4 to Broadwell
  – Today’s philosophy consists in “one size fit all”
    • Legacy software runs on both legacy and new hardware
  – Experiments pushing to higher and higher data rates
  – WLCG strategy: live within ~fixed budgets
  – Make better use of resources: the approach is changing

• Power consumption is becoming a hot-spot in the total bill
  – Especially in European Data Centers

• This will be even more important with the HL-LHC upgrade
  – Cope with 2-3x the amount of data
Today the CMS online farm consists of ~22k Intel Xeon cores
- The current approach: one event per logical core

- Pixel Tracks are not reconstructed for all the events at the HLT
- This will be even more difficult at higher pile-up
  - More memory/event

---

![Graph showing average processing time vs. average inst. luminosity.]
Today the CMS online farm consists of ~22k Intel Xeon cores.

- The current approach: one event per logical core.

Pixel Tracks are not reconstructed for all the events at the HLT.

This will be even more difficult at higher pile-up.

- More memory/event.

full track reconstruction and particle flow e.g. jets, tau
• Evaluation of Pixel Tracks combinatorial complexity could easily be dominated by track density and become one of the bottlenecks of the High-Level Trigger and offline reconstruction execution times.

• The CMS HLT farm and its offline computing infrastructure cannot rely on an exponential growth of frequency guaranteed by the manufacturers.

• Hardware and algorithmic solutions have been studied
Pixel Tracks on GPUs starting from Run-3
• Project started in 2016 by a very small group of passionate people, right after I gave a GPU programming course…

• Soon grown:
  – CERN: F. Pantaleo, V. Innocente, M. Rovere, A. Bocci, M. Kortelainen, M. Pierini, V. Volkl (SFT), V. Khristenko (IT, openlab)
  – INFN Bari: A. Di Florio, C. Calabria
  – INFN MiB: D. Menasce, S. Di Guida
  – INFN CNAF: E. Corni
  – SAHA: S. Sarkar, S. Dutta, S. Roy Chowdhury, P. Mal
  – TIFR: S. Dugad, S. Dubey
  – University of Pisa (Computer Science dep.): D. Bacciu, A. Carta
  – Thanks also to the contributions of many short term students (Bachelor, Master, GSoC): Alessandro, Ann-Christine, Antonio, Dominik, Jean-Loup, Konstantinos, Kunal, Luca, Panos, Roberto, Romina, Simone, Somesh

• Interests: algorithms, HPC, heterogeneous computing, machine learning, software eng., FPGAs…

• Lay the foundations of the online/offline reconstruction starting from 2020s (tracking, HGCal)
From RAW to Tracks during run 3

- Profit from the end-of-year upgrade of the Pixel to redesign the tracking code from scratch
  - Exploiting the information coming from the 4th layer would improve efficiency, b-tag, IP resolution
- Trigger avg latency should stay within max average time
- Reproducibility of the results (equivalence CPU-GPU)
- Integration in the CMS software framework
- Targeting a complete demonstrator by 2018 H2

- Ingredients:
  - Massive parallelism within the event
  - Independence from thread ordering in algorithms
  - Avoid useless data transfers and transformations
  - Simple data formats optimized for parallel memory access
- Result:
  - A GPU based application that takes RAW data and gives Tracks as result
Algorithm Stack

Input, size linear with PU

- Raw to Digi
- Hits - Pixel Clusterizer
- Hit Pairs
- CA-based Hit Chain Maker
- Riemann Fit

Output, size ~linear with PU + dependence on fake rate
Integration studies
• Different possible ideas depending on:
  – the fraction of the events running tracking
  – other parts of the reconstruction requiring a GPU
• Every FU is equipped with GPUs
  – tracking for every event

• Rigid design
  + easy to implement
  - Requires common acquisition, dimensioning etc
Integration in the Cloud/Farm

- A part of the farm is dedicated to a high density GPU cluster
- Tracks (or other physics objects like jets) are reconstructed on demand
- Simple demonstrator developed using HPX by STE||AR group
  - Offload kernels to remote localities
  - Data transfers will be handled transparently using percolation

**Diagram:**
- Filter Units
- DL Inference Accelerators
- Option 2
- Builder Units or disk servers
- GPU Pixel Trackers

- Flexible design
  + Expandible, easier to balance
  - Requires more communication and software development
• Builder units are equipped with GPUs:
  – events with already reconstructed tracks are fed to FUs with GPUDirect
  – Use the GPU DRAM in place of ramdisks for building events.

• Very specific design
  + fast, independent of FU developments, integrated in readout
  - Requires specific DAQ software development: GPU “seen” as a detector element
Tests
• We acquired a small machine for development and testing:
  – 2 sockets x Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz (12 physical cores)
  – 256GB system memory
  – 8x GPUs NVIDIA GTX 1080Ti
• The rate test consists in:
  – preloading in host memory few hundreds events
  – Assigning a host thread to a host core
  – Assigning a host thread to a GPU
  – Preallocating memory for each GPU for each of 8 cuda streams
  – Filling a concurrent queue with event indices
  – During the test, when a thread is idle it tries to pop from the queue a new event index:
    • Data for that event are copied to the GPU (if the thread is associated to a GPU)
    • processes the event (exactly same code executing on GPUs and CPUs)
    • Copy back the result
  – The test ran for approximately one hour
  – At the end of the test the number of processed events per thread is measured, and the total rate can be estimated
What happens in 10ms
Rate test

Events processed by processing unit

CPU0: 3000000 events
CPU1: 2500000 events
CPU2: 2000000 events
CPU3: 1500000 events
CPU4: 1000000 events
CPU5: 500000 events
CPU6: 0 events

CPU7: 2500000 events
CPU8: 2000000 events
CPU9: 1500000 events
CPU10: 1000000 events
CPU11: 500000 events
CPU12: 0 events
CPU13: 2500000 events
CPU14: 2000000 events
CPU15: 1500000 events
CPU16: 1000000 events
CPU17: 500000 events
CPU18: 0 events
CPU19: 2500000 events
CPU20: 2000000 events
CPU21: 1500000 events
CPU22: 1000000 events
CPU23: 500000 events
Rate test

• Total rate measured:
  – 8xGPU: 6527 Hz
  – 24xCPUs: 613 Hz

• When running with only 24xCPUs
  – Rate with 24xCPUs: 777 Hz
Energy efficiency

- During the rate test power dissipated by CPUs and GPUs was measured every second
  - Nvidia-smi for GPUs
  - Turbostat for CPUs

- 8 GPUs: 1037W
  - 6.29 Events per Joule
  - 0.78 Events per Joule per GPU

- 24 CPUs in hybrid mode: 191W
  - 3.2 Events per Joule
  - 0.13 Events per Joule per core

- 24 CPUs in CPU-only test: 191W
  - 4.05 Events per Joule
  - 0.17 Events per Joule per core
Conclusion

• Tracking algorithms have been redesigned with high-throughput parallel architectures in mind

• Improvements in performance may come even when running sequentially
  – Factors at the HLT, tens of % in the offline, depending on the fraction of the code that use new algos

• The GPU and CPU algorithms run and produce the same bit-by-bit result
  – Transition to GPUs@HLT during Run3 smoother

• Integration in the CMS High-Level Trigger farm under study

• DNNs under development for early-rejection of doublets based on their cluster shape and track classification

• Using GPUs will not only allow to run today’s workflows faster, but will also enable CMS to achieve better physics performance, not possible with traditional architectures
Questions?
Back up
The compatibility between two cells is checked only if they share one hit:
- AB and BC share hit B

In the R-z plane a requirement is alignment of the two cells:
- There is a maximum value of $\theta$ that depends on the minimum value of the momentum range that we would like to explore
• In the transverse plane, the intersection between the circle passing through the hits forming the two cells and the beamspot is checked:
  - They intersect if the distance between the centers $d(C,C')$ satisfies:
    \[ r' - r < d(C,C') < r' + r \]
  - Since it is a Out–In propagation, a tolerance is added to the beamspot radius (in red)

• One could also ask for a minimum value of transverse momentum and reject low values of $r'$
• Hits on different layers
• Need to match them and create quadruplets
• Create a modular pattern and reapply it iteratively
• First create doublets from hits of pairs
RMS HEP Algorithm

- First create doublets from hits of pairs
- Take a third layer and propagate only the generated doublets
This kind of algorithm is not very suitable for GPUs:

- Absence of massive parallelism
- Poor data locality
- Synchronizations due to iterative process
- Very Sparse and dynamic problem (that’s the hardest part, still unsolved)
- Parallelization does not mean making a sequential algorithm run in parallel
  - It requires a deep understanding of the problem, renovation at algorithmic level, understanding of the computation and dependencies

The algorithm was redesigned from scratch getting inspiration from Conway’s Game of Life

- Traditional Cellular Automata excluded because 2x slower
  - quadruplets by triplets sharing a doublet
Quadruplets finding

blockIdx.x and threadIdx.x = Cell id in a Root LayerPair

blockIdx.y = LayerPairIndex in RootLayerPairs

Each cell on a root layer pair will perform a parallel DFS of depth = 4 following outer neighbors.