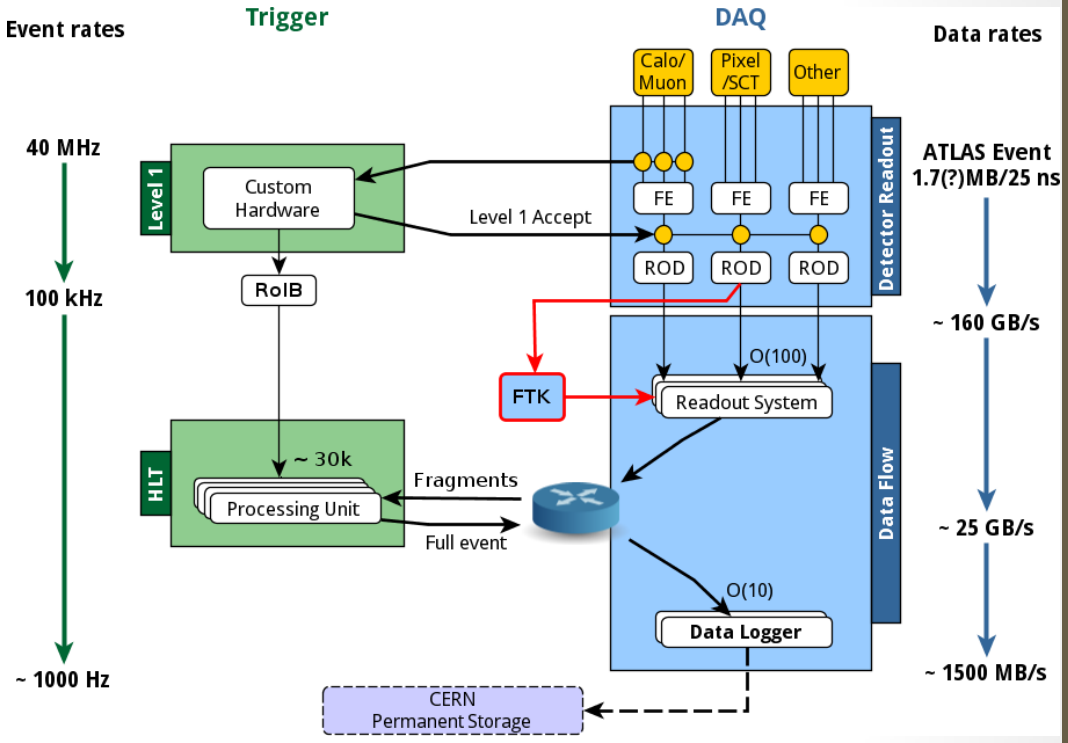
Triggering Events with GPUs at ATLAS

S. Kama, J. Augusto Soares, J. Baines, M. Bauce, T. Bold, P. Conde Muino, D. Emeliyanov, R. Goncalo, A. Messina, M. Negrini, L. Rinaldi, A. Sidoti, A. Tavares Delgado, S. Tupputi, L. Vaz Gil Lopes

ATLAS Trigger and DAQ

- Composed of hardware based Level-1 (L1) and software based High Level Trigger(HLT)
- Reduces 40MHz input event rate to about 1kHz (~1500MB/s output)
- L1 identifies interesting activity in small geometrical regions of detector called Region of Interests (RoI)
- RoIs identified by L1 are passed to the HLT for event selection

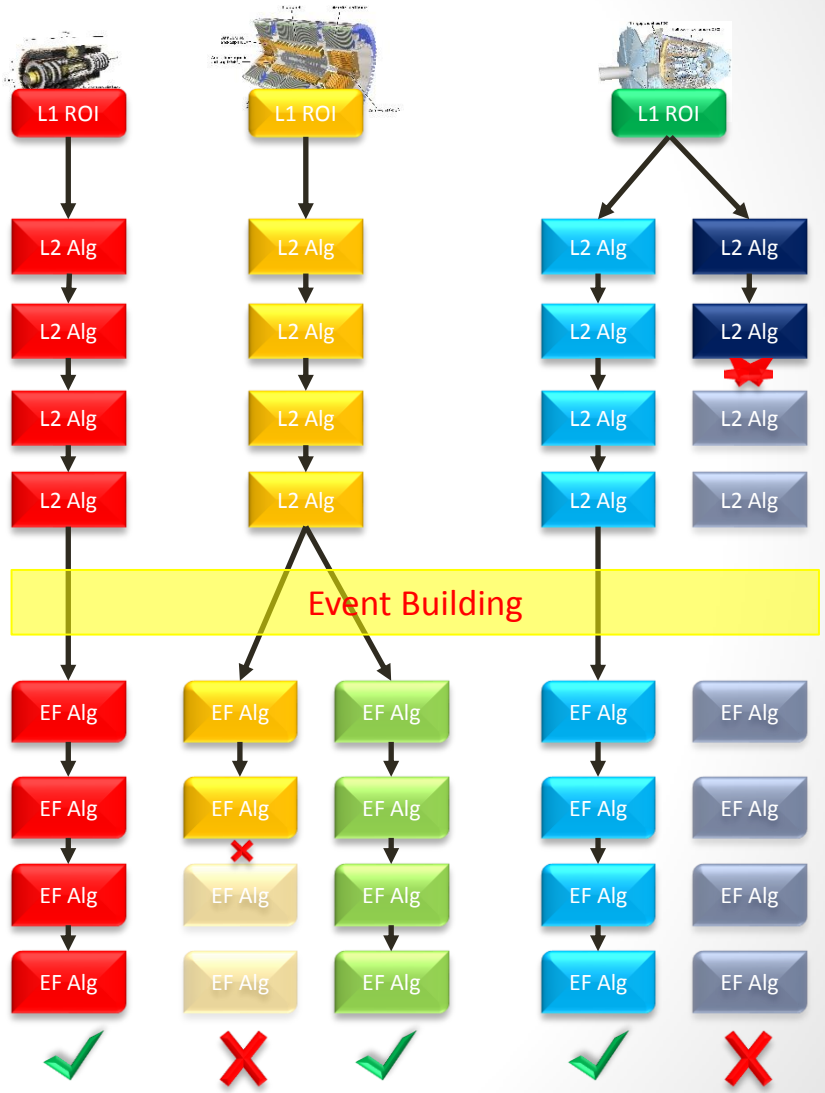


See R.Hauser's talk

With about 25k processes
~25k/100kHz = ~250ms/Event decision time

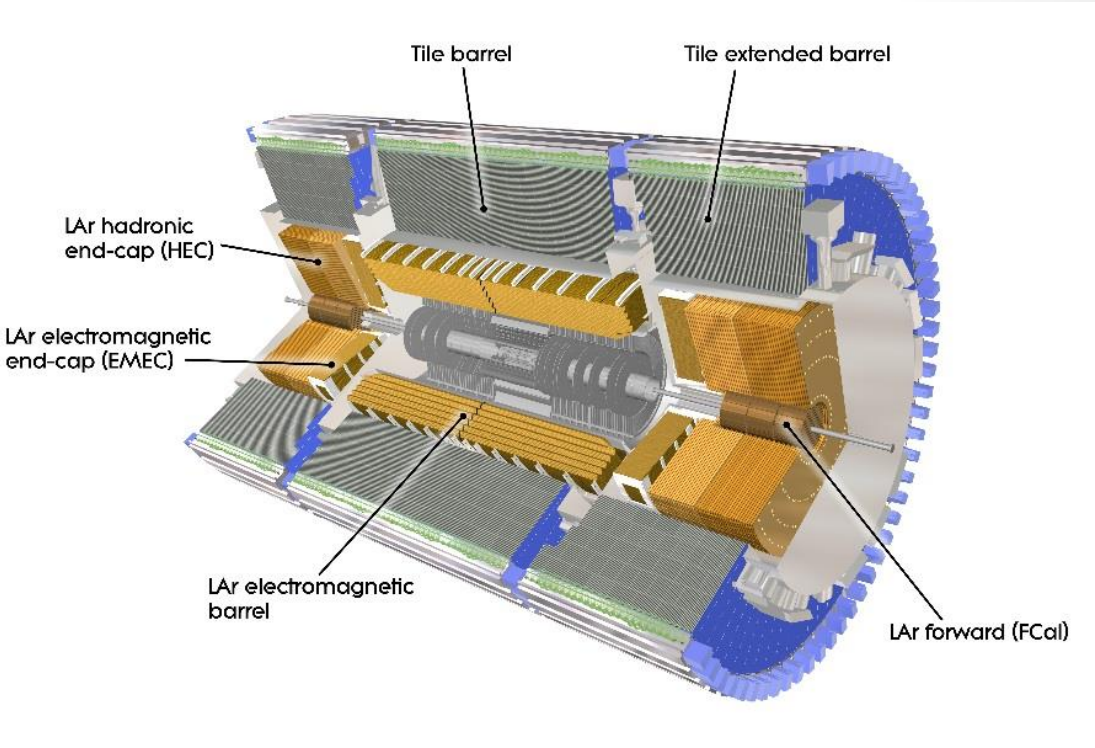
High Level Trigger

- HLT uses Stepwise processing of sequences called Trigger Chains(TC)
- Each chain is composed of Algorithms and seeded by RoIs from L1
- Same RoI may seed multiple chains
- Same Algorithm may run on different data
- An algorithm runs only once on same data (caching)
- If a RoI fails a selection step further algorithms in chain are not executed
- Initial algorithms (L2) work on partial event data (2-6%). Later algorithms have access to full event data



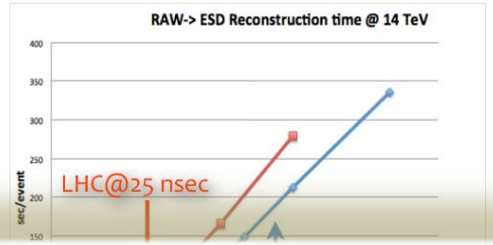
Inner Detector and Calorimeters

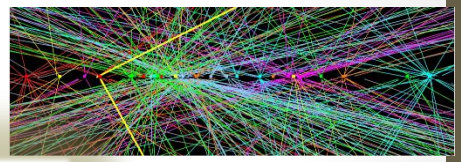
- Inner detector houses trackers and composed of
  - Pixel detector
  - Silicon strip detector (SCT)
  - Transition Radiation Tracker (TRT)
- Calorimeters contain electromagnetic and hadronic components and composed of
  - Liquid Argon (LAr)
  - Tile Calorimeters



GPU Demonstrator

- Increasing LHC instantaneous luminosity is leading to higher pileup
- CPU time rises rapidly with pile-up due to
- HLT farm size i





cks from primary interaction
pileup interaction vertices

ATLAS is developing and expanding demonstrator exploiting GPUs in:

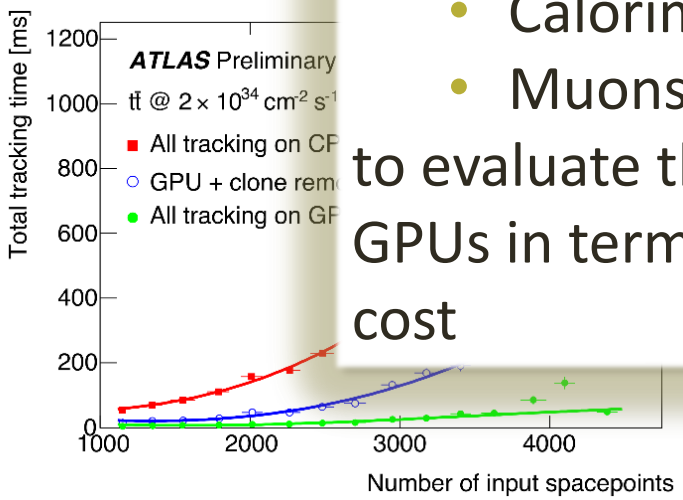
- Inner Detector-Tracking
- Calorimeter-Topo-clustering
- Muons-Tracking

to evaluate the potential benefit of GPUs in terms of throughput per unit cost

compute

onstrator

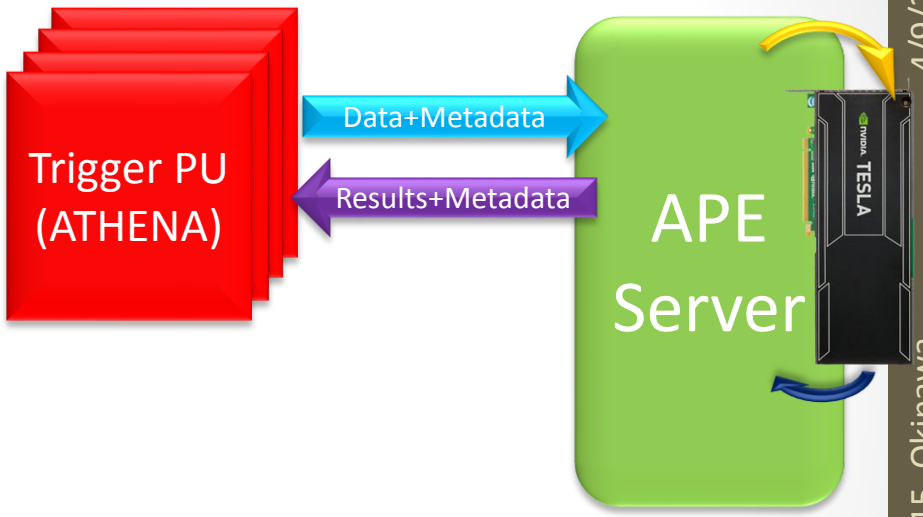
up to 12x)*



*ATL-DAQ-SLIDE-2014-635

Offloading Mechanism

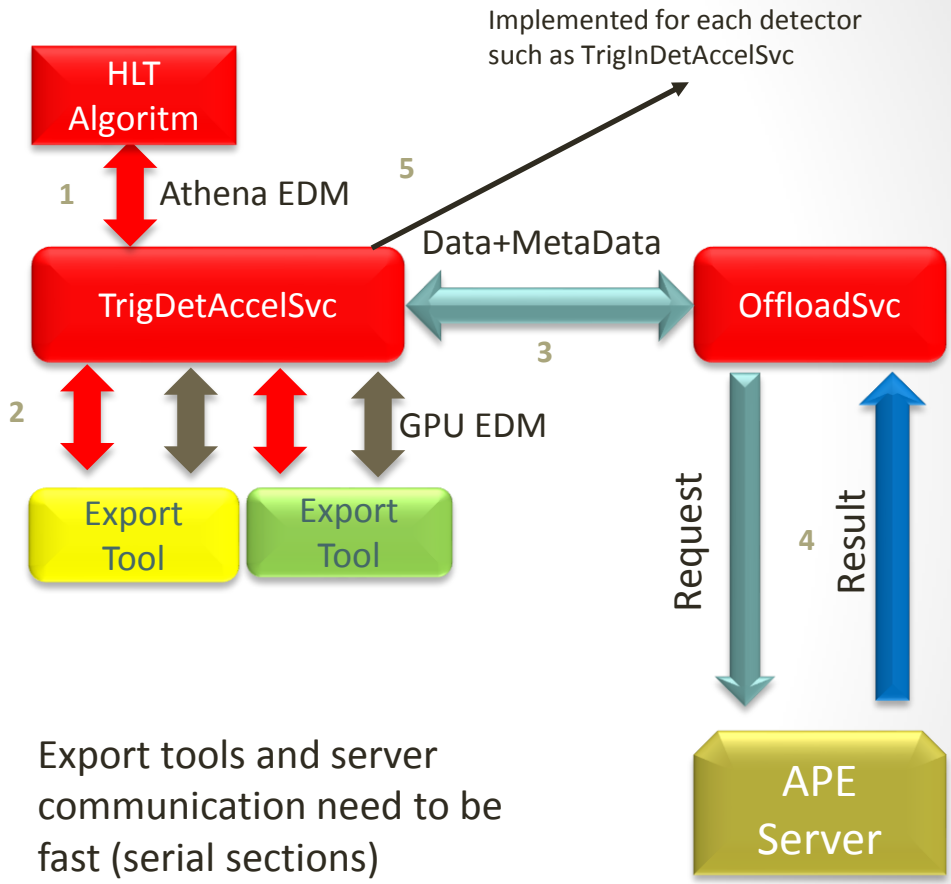
- Trigger Processing Units integrate ATLAS offline software framework, ATHENA, to online environment
- Many PU processes run on each Trigger host
- A client-server approach is implemented to manage resources between multiple PU processes.
- PU prepares data to be processed and sends it to server
- Accelerator Process Extension (APE) Server manages offload requests and executes kernels on GPU(s)
- It sends results back to process that made the offload request



Server can support different hardware types (GPUs, Xeon-Phi, CPUs) and different configurations such as GPU/Phi mixtures and in-host off-host accelerators.

Offloading - Client

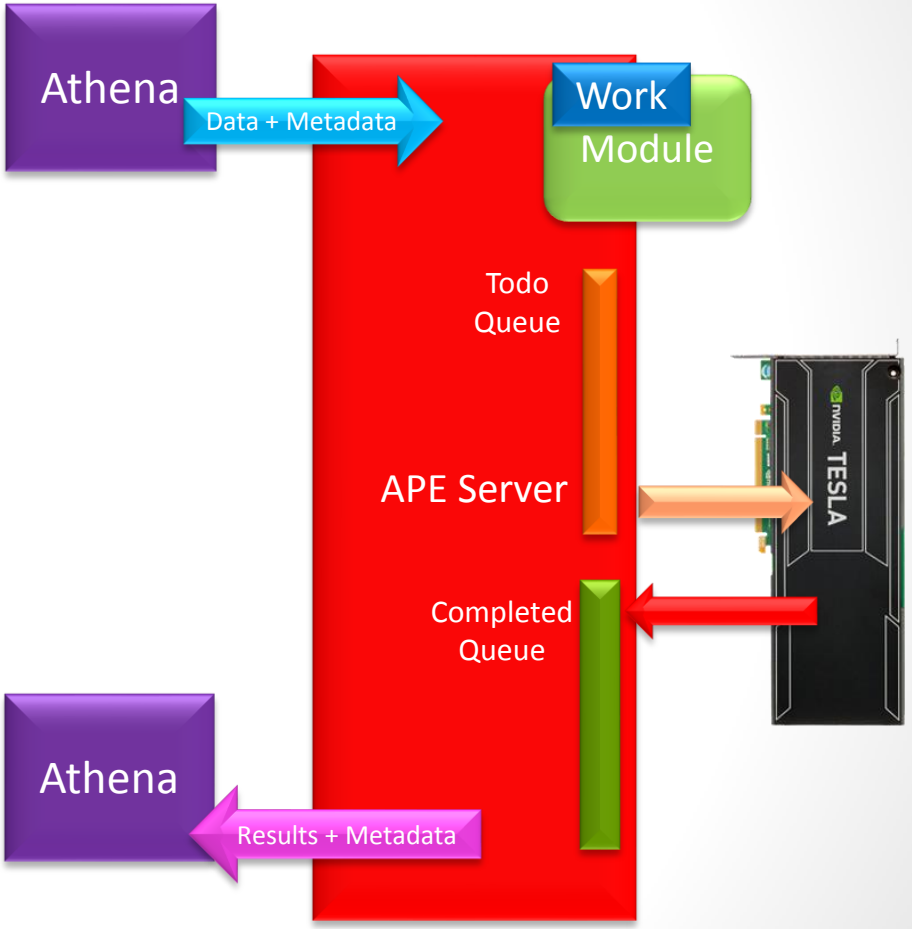
1. HLT Algorithm asks for offload to TrigDetAccelSvc
2. TrigDetAccelSvc converts C++ classes for raw and reconstructed quantities from the Athena Event Data Model(EDM) to GPU optimized EDM through Data Export Tools
3. Then it adds metadata and requests offload through OffloadSvc
4. OffloadSvc manages multiple requests and communication with APE server
5. Results are converted back to Athena EDM by TrigDetAccelSvc and handed to requesting algorithm



Export tools and server communication need to be fast (serial sections)

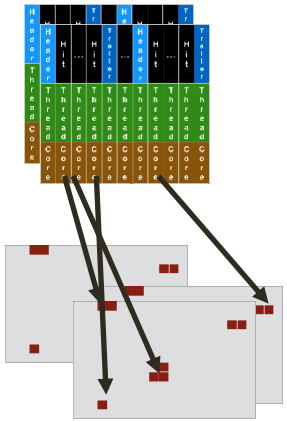
Offloading - Server

- APE Server uses plug-in mechanism It is composed of
- **Manager** handles communication with processes and scheduling
  - Receives offload requests
  - Passes the request to appropriate module
  - Executes Work items
  - Sends the results back to requesting process
- **Modules** manage GPU resources and create work items
  - Manage constants and time-varying data on GPU
  - Bunch multiple requests to optimize utilization
  - Create appropriate work items
- **Work** items run GPU kernels such as clusterization on given data and prepare results
- Each detector implements their own module

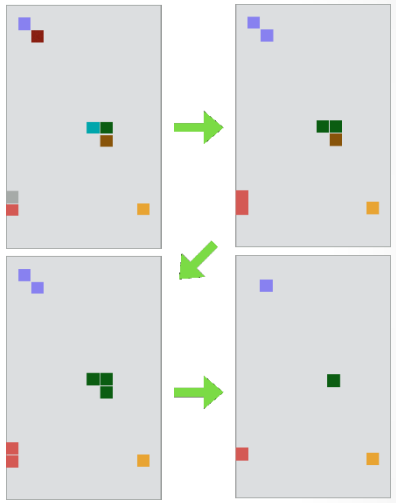


ID Module

- Tracking is most time consuming part of trigger
- ID module implements several CPU intensive steps on GPU
- Bytestream decoding converts detector encoded output to hits on Pixel and SCT modules
- Charged particles typically activate one or more neighboring strips or pixels. Hit clustering merges these by a Cellular Automata algorithm



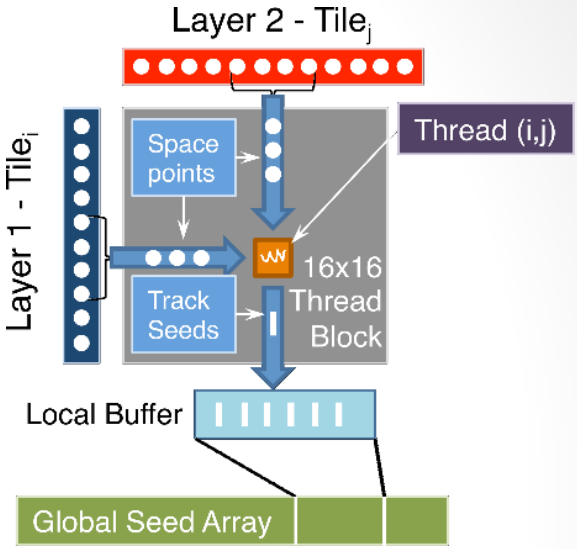
Each thread works on a word and decodes it. Data contains hits on different modules on different ID layers

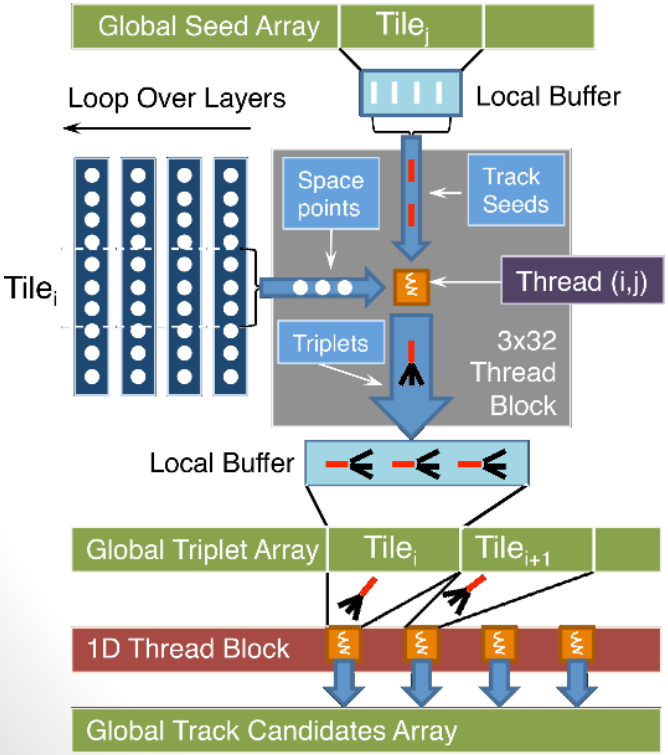


Each hit start as an independent cluster. Adjacent clusters are merged in each step until all adjacent cells belong to same cluster. Each thread works on a different hit

ID Tracking

Tracking starts with pair forming. A 2-D thread array checks for pairing condition and selects suitable pairs

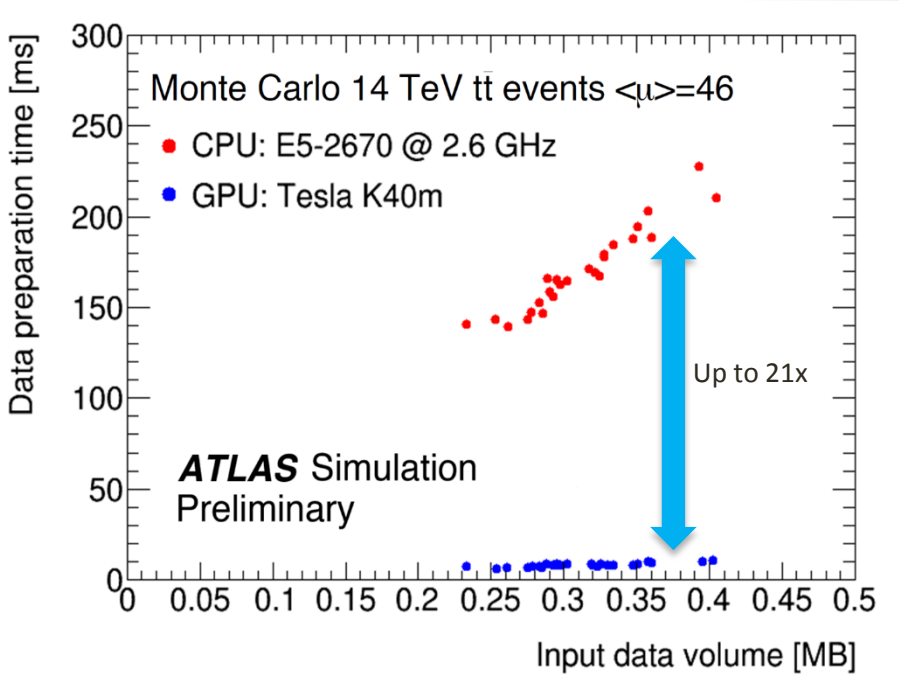




Then these pairs are extrapolated to outer layers by a 2D thread block to form triplets. Finally triplets are combined to form Track Candidates.
In clone removal step, track candidates starting from different pairs but having same outer layer hits are merged to form tracks

ID Speedup

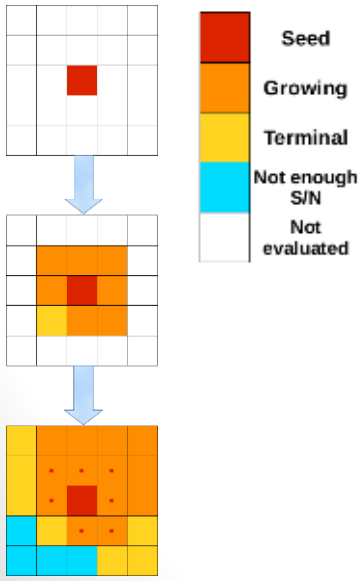
- Raw detector data decoding and clusterization algorithms form the first step of reconstruction(data preparation)
- Initial tests with Monte-Carlo samples shown up to 21x speed up compared to serial Athena job.
- A new upgraded tracking algorithm is being implemented



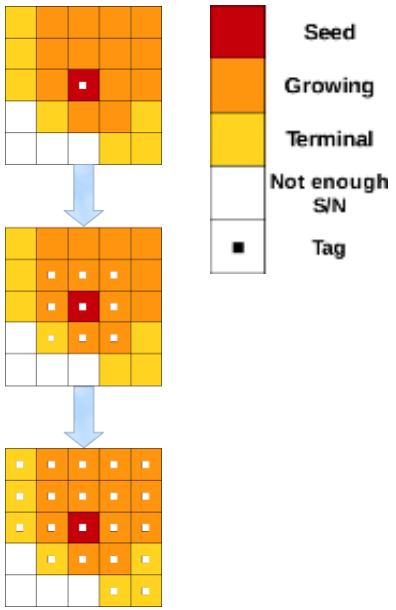
Calorimeter Topoclustering

- Topocluster algorithm classifies calorimeter cells by signal/noise
  - S/N>4 are Seeds
  - 4>S/N>2 are Growing cells
  - 2>S/N>0 are Terminal cells
- Growing cells around Seeds are included until they don't have any more Growing or Terminal cells around them
- Parallel implementation uses Cellular Automata algorithm to parallelize the task
- Implementation is underway

Standard algorithm



Parallel algorithm



1. Assign one thread for each cell
2. Start with adding Seeds to cluster.
3. At each iteration join to the cluster which contain highest S/N neighboring cell
4. Terminate when maximum radius is reached or can't include anymore cells

Next Steps

- Finalize porting of ID track finding and clone removal algorithms
- Finalize and include Calorimeter Topoclustering module
- Implement and include Muon tracking module
- Make detailed measurements including throughput per unit cost by the end of the year and use this to estimate potential benefit in future HLT farm

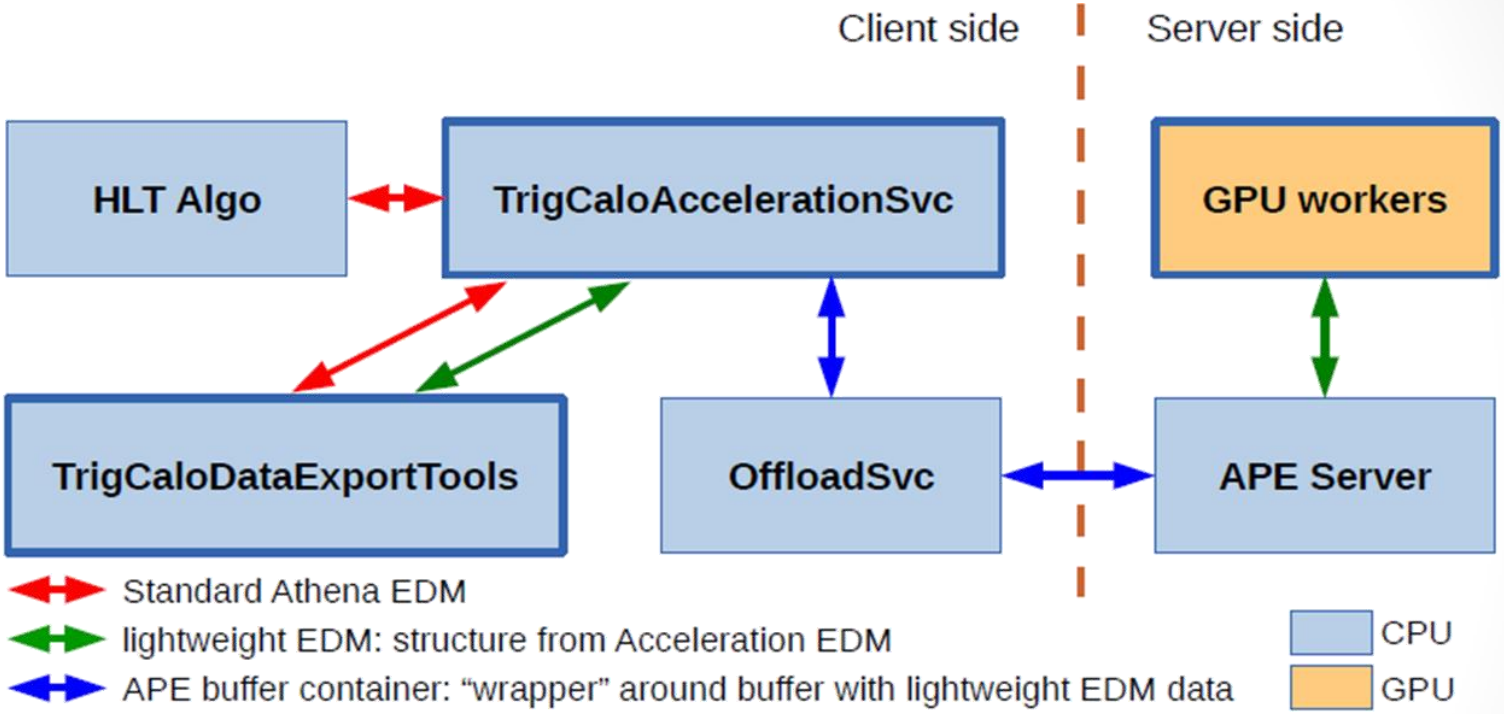
Summary & Outlook

- Increasing instantaneous luminosity of LHC necessitates parallel processing
- Massive parallelization of trigger algorithms on GPU is being investigated as a way to increase the compute-power of the HLT farm
- ATLAS developed APE framework to manage offloading from multiple processes
- Inner detector Trigger data preparation algorithms are successfully offloaded to GPU, resulting in ~21x speedup compared to CPU implementations
- Further algorithms for ID, Calorimeter and Muon systems are being implemented.

THANK YOU

BACKUPS

Architectural Design



Requires:

- Data conversion between Athena EDM and lightweight EDM.
- Lightweight data shared between client and server.
- Data transfer to/from GPU.

Implementation