CERN AdI Follow-Up Meeting

Ricardo Abreu
CERN
AdI
Supervised by Joannes Andreas Bogaerts

January 26, 2010

Summary

- Context
  - ATLAS
  - TDAQ
- My work
  - HLTTestApps
  - Histmon
  - Dataflow upgrade prototype

- ATLAS detector and ATLAS experiment
  - Searching for new outcomes from high-energy proton-proton collisions - better understanding the Universe
  - Its domain is "general particle physics"
- Working on computational aspects
  - Data Acquisition (DAQ)
  - High Level Trigger (HLT)
  - HLT + DAQ $\simeq$ HLT/DAQ (or simply TDAQ)
- Incredible amounts of data are produced
  - Cannot store it all
  - Cannot (and need not) be all analyzed by physicists
  - To a great extent, has to be processed in real-time

Data from the detector grouped in *events* - overloaded term

Event

At this stage, it refers to the data collected during one *bunch-crossing* (plus any delayed data from previous bunch-crossings)

- Events are analyzed and filtered
  - Specialized algorithms make the decision for each event
- As it travels the dataflow path, an event is either discarded or kept and enlarged with analysis data
- Ultimately, if not rejected, an event is recorded in offline storage facilities for further processing

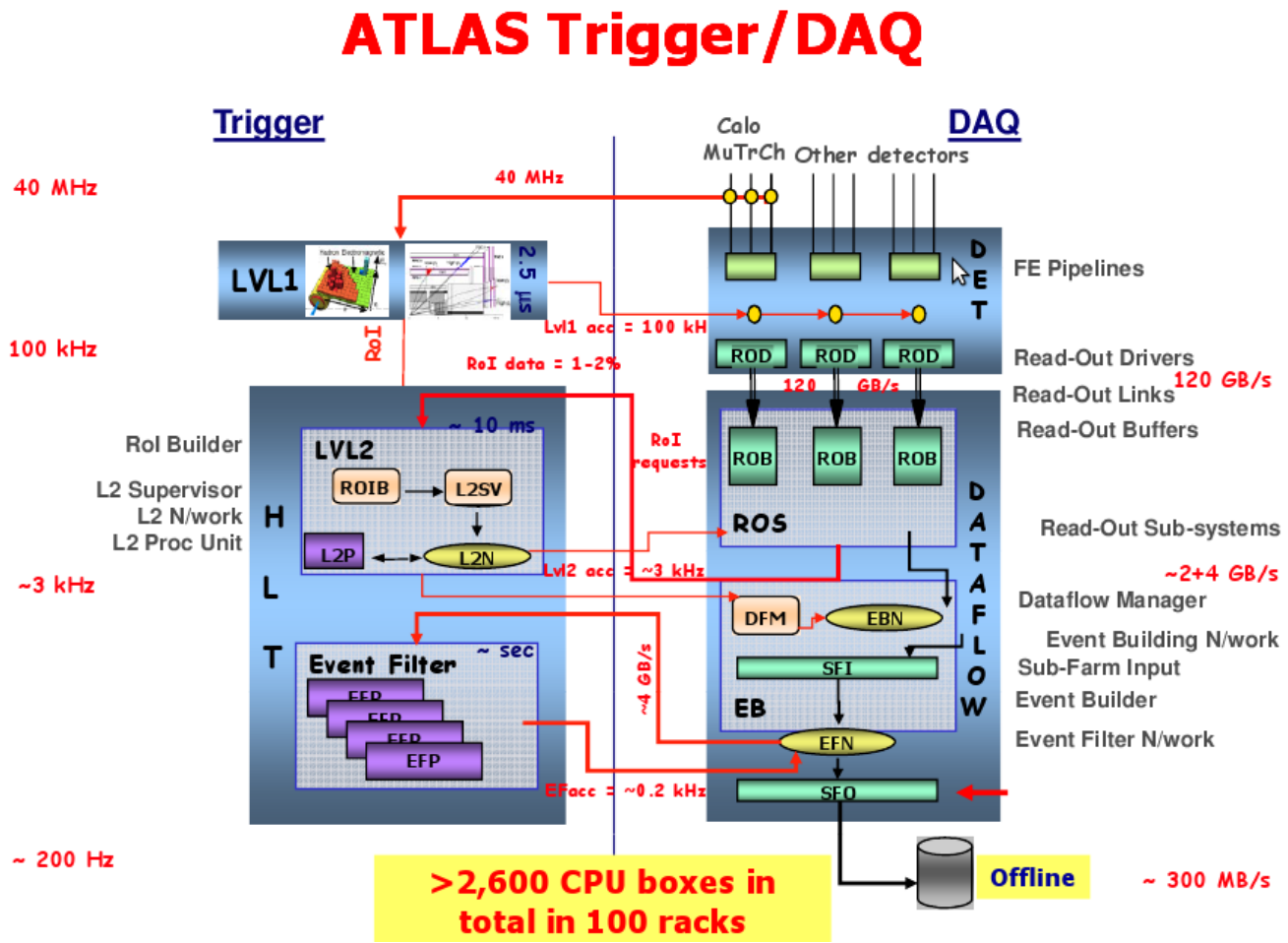
The TDAQ system provides the environment for the execution of these algorithms

- 14 TeV proton-proton bunch-crossings every 25 ns
  - Events produced at a rate of 40 MHz
  - Only 200 Hz can be recorded
  - Factor of reduction of $O(100000)$
- Filtering done in stages
  - Level 1 - custom hardware - 100 KHz
  - HLT - software
    - **Level 2** - 3 KHz
    - **Event Filter** (the 3^{rd} level of filtering) - 200 Hz

Rationale

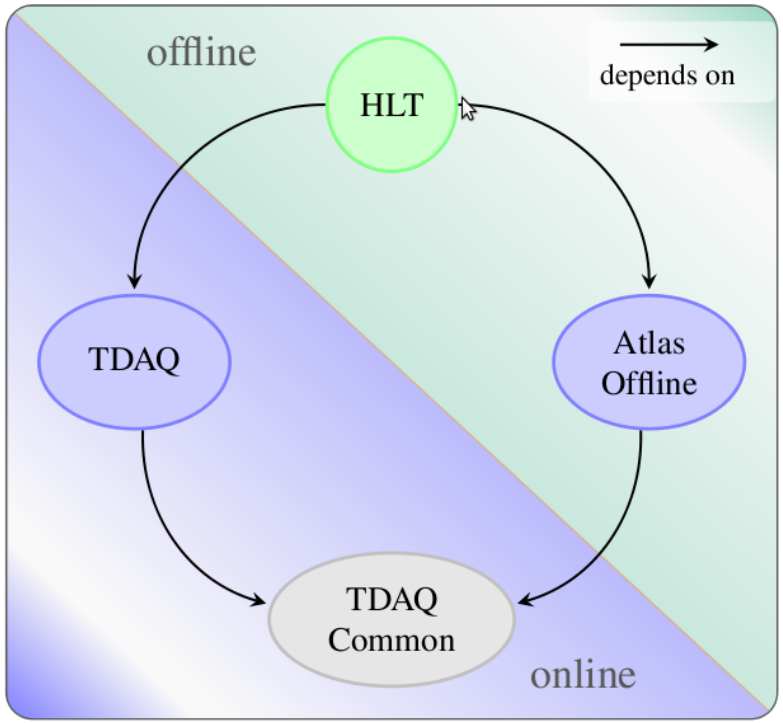
Discard obviously irrelevant events as soon as possible - more time left for data that demands more attention

LVL2 and EF rely on the many other components that the TDAQ infrastructure comprises, which are responsible for distinct but equally important tasks.



HLTTestApps

- A package of the HLT project
- Provides two tools: *athenaMT* and *athenaPT*
- Sits on top of TDAQ and *AttlasOffline* projects
- Integrates the offline and online environments



Test facility for online and offline components

- Last means for testing selection algorithms before they go online
- athenaMT and athenaPT emulate L2 and EF

- Support for many different use-cases. Examples are:
  - default mode - read events from file and process them in L2/EF
  - interactive mode with Python prompt
  - with histogramming - creates a localhost partition
  - monitoring mode - connecting to a running partition

Examples of improvements I made:

- Added support for *user commands* (client/server)
- Fault tolerance for partially invalid events
- Support for Google's *TCMalloc* library
- Signal catching to ensure proper cleanup

Histmon

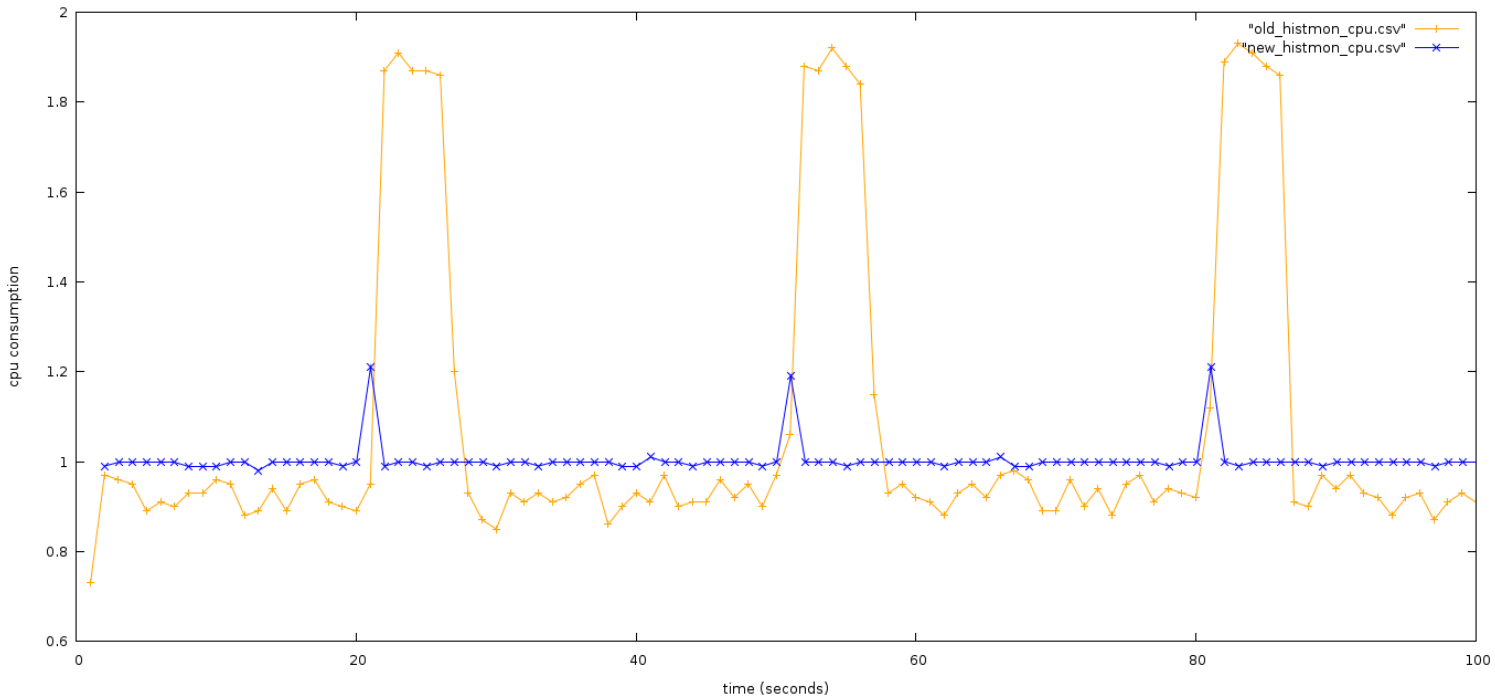
- A package that belongs to TDAQ
- Not an application but a plugin
- For publishing **histograms** produced by HLT/DAQ
  - Histograms are one of the two main vehicles of monitoring data for ATLAS

Overview of histogram flow path

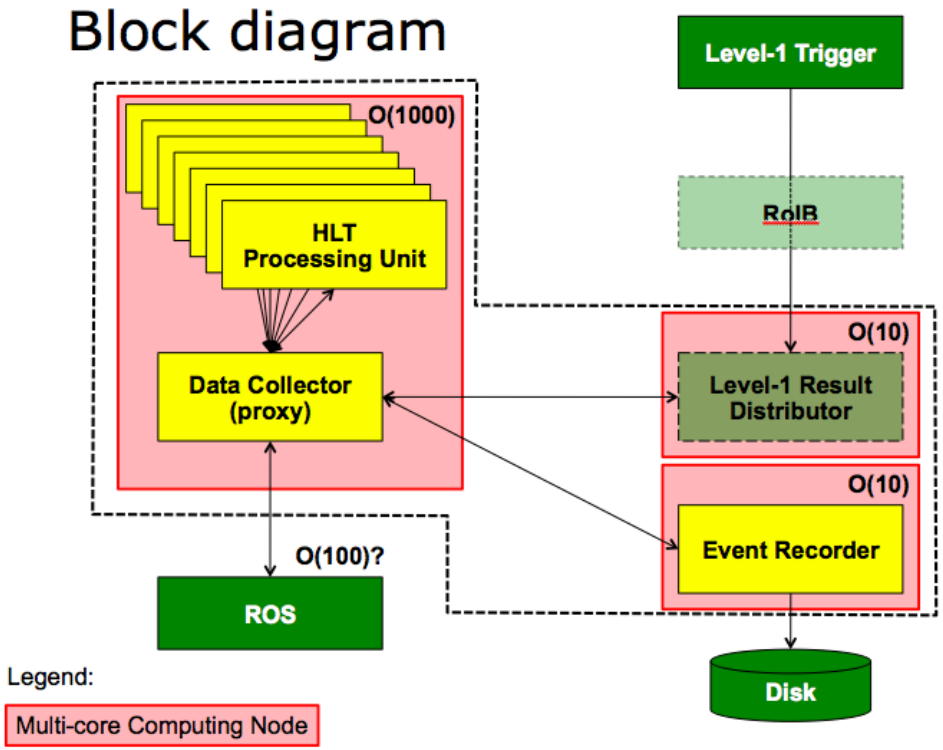
- Each application uses histmon for registering histograms
- Histmon periodically publishes histograms to local IS servers
- The *Gatherer* sums the histograms and forwards them to the next IS server in the hierarchy
- At the top IS server, summed histograms are available at a central point

Main improvements:

- Clearly separated interfaces from implementations
- Parallelized histogram publishing
  - Solved publishing period bug
  - Improved performance



Prototype for DataFlow Upgrade



- Merging *L2PU* and *EFPT* into *HLTPU*
- Automatic balancing
- Aiming at greater performance
- Simplification of the system - less applications
- Smaller code base
- More easily maintainable system

Conclusion

- Lots of learning
- Gaining experience
- Actively contributing
- Feeling as a part of the experiment

Thanks

Thank You!