The ATLAS Data Flow System for LHC Run II

Andrei Kazarov* on behalf of the ATLAS Collaboration

*CERN, on leave from Petersburg NPI NRC KI

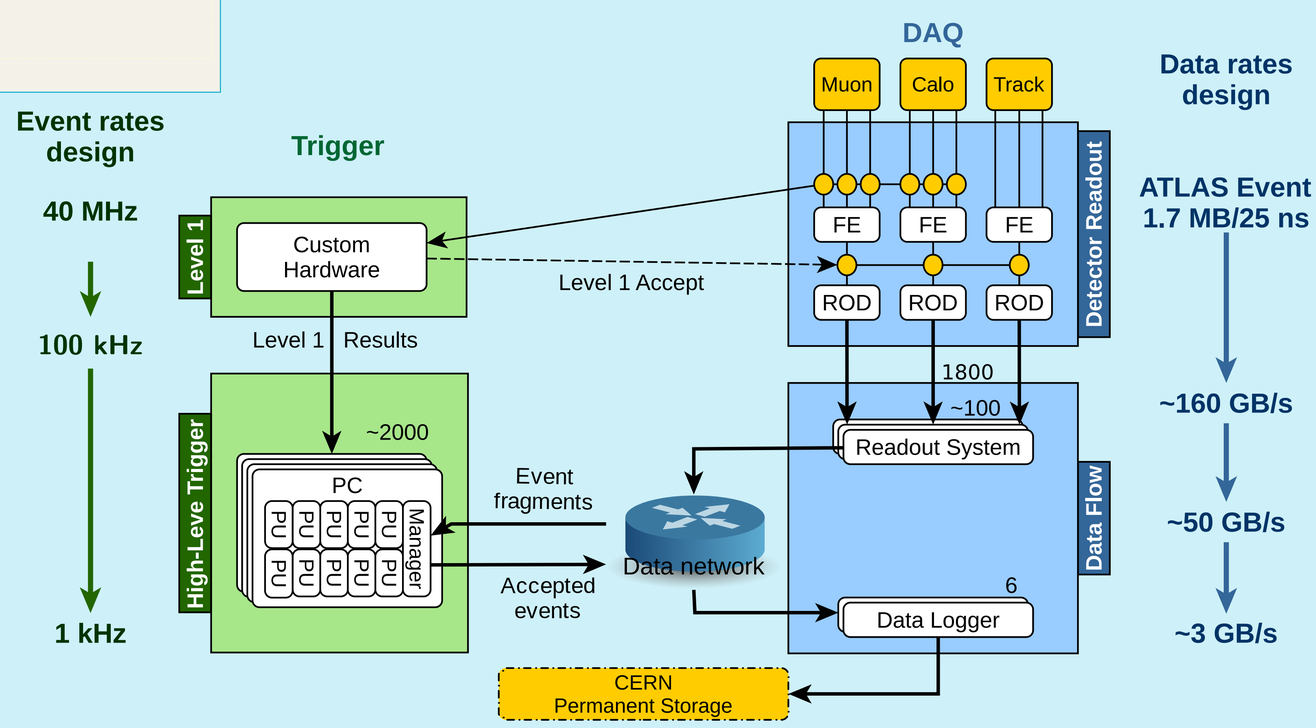
Introduction

In the ATLAS experiment[1], the Trigger and Data Acquisition (TDAQ) system has been upgraded to deal with the increased event rates. The Data Flow (DF) element of the TDAQ is a distributed hardware and software system responsible for buffering and transporting event data from the readout system to the High Level Trigger (HLT) and to the event storage. The DF has been reshaped in order to profit from newer technologies and to maximize the flexibility and efficiency of the data selection process.

The updated DF is radically different from the previous implementation both in terms of architecture, used hardware and expected performance.

New requirements for DAQ in Run II

	bunch spacing, ns	Inst. Luminocity cm^{-2} s^{-1}	L1 accept rate, kHz	Readout fraction, %	Even building bandwidth, GB/s	Recording rate, kHz/GB/s	N of readout channels	Event size, MB (max)
Run I	50	$8x10^{33}$	70	15	10	1/1.6	1600	1.6
Run II (design)	25	$1.7x10^{34}$	100	25	50	2/3	1800	2.0

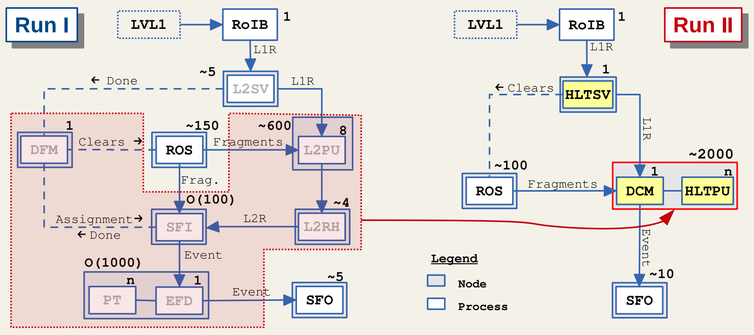


New Data Flow architecture

The updated DF architecture is radically different from the previous implementation. Two levels of software filtering, known as L2 and the Event Filter, are now merged into a single level, performing incremental data collection and analysis on the nodes of the HLT farm. This design has many advantages, among which are:

- the radical simplification of the architecture
- the flexible and evenly balanced utilization of the computing resources over the farm, based on the active HLT algorithms
- the sharing of code and services on nodes

In addition, logical farm slicing, with each slice managed by a dedicated supervisor, has been dropped in favor of global management by a single master (HLTSV) operating at 100 kHz. This simplifies management of the farm, at the cost of higher reliability requirements to the central node.



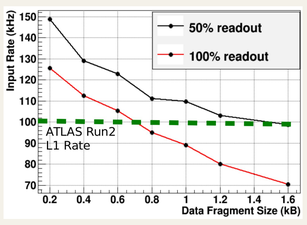
The Region of Interest (RoI) concept has been kept, and processing and data collection proceeds in stages, beginning with fast algorithms based on RoIs. The decision when to build the event is flexible, and afterwards more off-line style algorithms have access to the full event.

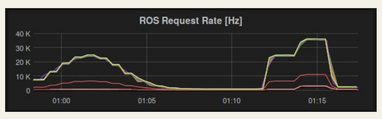
- The Readout System (ROS) buffers front-end data from the detectors and provides a standard interface to the DAQ
- The Region of Interest Builder (RoIB) receives L1 trigger information and RoIs and combines the information for the HLT Supervisor
- The HLT supervisor (HLTSV) schedules events to the HLT farm and handles eventual time-outs
- The Data Collection Manager (DCM) handles all I/O on the HLT nodes, including RoI requests from the HLT and full event building
- The HLT processing tasks are forked from a single mother process to maximize memory sharing and run the ATLAS Athena/Gaudi framework
- The Data Loggers (SFO) are responsible for saving accepted events to disk, and for sending the files to EOS

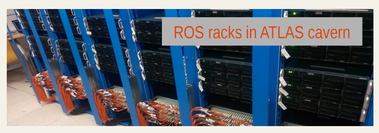
New Readout System

Addition of new detectors to ATLAS, higher luminosity and L1 trigger rates requires more dense and more performant readout system.
Fully new ROS PCs feature:

- 2U form factor instead of 4U
- 4 x 10GbE Ethernet per ROS PC (was 2x GbE)
- Higher density of optical link connectors: 12 per card, 2 cards per PC
- A fully connected ROS (24 links) ROS can sustain up to 50% readout for a wide set of request patterns (e.g. 35kHz of L1 during VdM scans on the plot below)
- A ROS with fewer input links and/or small enough fragments can run at 100 kHz.



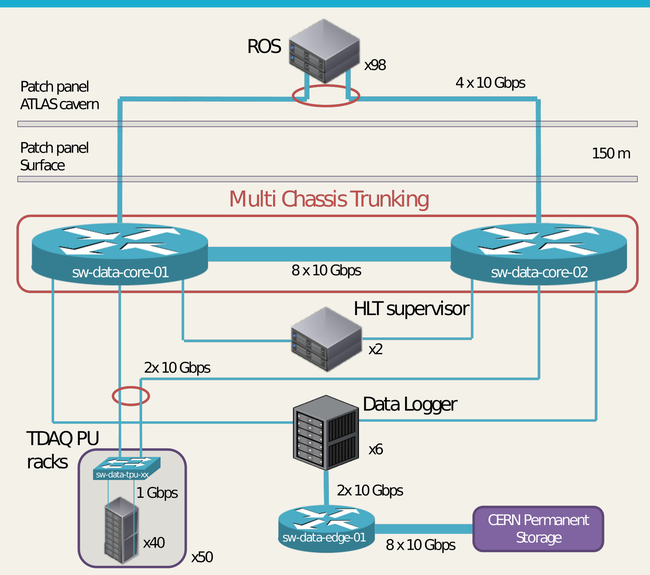




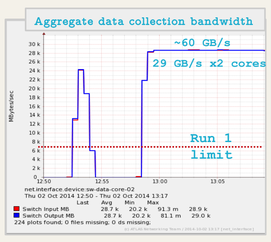
New powerful ROBIN (S-Link input and buffer hardware): ALICE C-RORC[2] with ATLAS firmware and software

- PCI Express 8x lanes
- FPGA: Xilinx Virtex-6 @ 125MHz
- Buffer memory: DDR3-1600 SO-DIMM RAM 2x4GB
- 12 Input channels

Data and Control Networks



The data flow network has seen a significant upgrade and simplification.
Multi-Chassis Trunking (a Brocade technology that allows multiple switches to appear as single logical switch) of the core routers provides load balancing and link redundancy to the network.
Virtual output queue mechanism avoids head of line blocking and allows the routers to run in non-blocking mode.
The control network has been made more redundant with active backup solutions for all important components.



HLT Supervisor

A single HLT supervisor replaces the set of L2 supervisors used in Run I:

- It uses a heavily multi-threaded asynchronous design using the Boost ASIO[3] library for communication and Intel Thread Building Blocks[4] for concurrent data structures
- 2 x 10 GbE to the data flow network
- A single application can handle the input from the RoIB and manage the HLT farm of ~2000 machines at ~115 kHz under realistic ATLAS conditions

In the future it is foreseen merging the RoIB and HLTSV functionalities into a single PC equipped with C-RORC cards.

Data Collection Manager

The DCM is a single application per HLT node that deals with all data requests from the multiple HLT processing tasks:

- It handles all requests to the ROS
- It communicates to the HLT tasks via sockets and shared Memory
- Its design is essentially single-threaded based on non-blocking I/O using the Boost ASIO library
- A credit based traffic shaping mechanism[5] is used to prevent overloading the incoming network link
- It compresses the event payload before sending it to the data logger

HLT Processing Unit

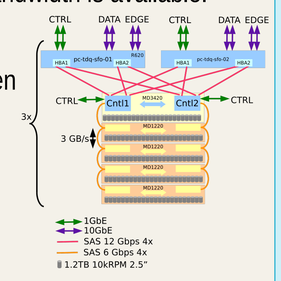
The HLT processing unit encapsulates the Athena framework that is running the actual HLT algorithms.

- It communicates with the DCM for I/O requests and provides the trigger decision for each event.
- On each node a mother process is started first and goes through all the configuration process. A set of child processes is forked when the run starts, using Linux kernel copy-on-write feature and thus maximizing the memory sharing by similar processes.
- Crashed HLT applications can be quickly replaced by forking another child instance.
- Tests with the full 2012 trigger menu show a memory consumption on a node of ~ 1.8 Gbyte (taken by the mother process) + N x 700 MByte (where N is number of forked child processes, e.g. N=12 for 12 core CPUs)

Data Storage

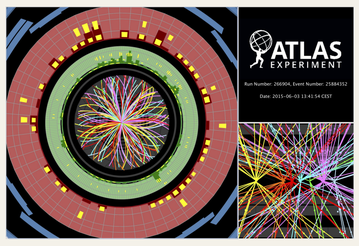
In Run I a data logger[6] was a PC with 3 internal Raid5 raid arrays of 8 disks each.
For Run II external SAS units with multiple front-ends and redundant data paths for fault tolerance and resilience are used.
Total capacity of the system is 340 TB, allowing ATLAS to record data being disconnected from offline storage for 24hrs.
Performance depends on trigger menu, number of streams, etc. In typical configuration, 4GB/s of peak recording bandwidth is available.

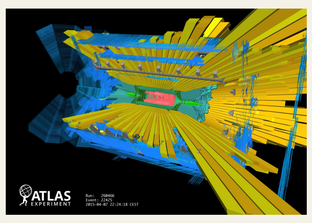
Background jobs copy the files to permanent storage, deleting them on the local disk only when they are safely on tape.

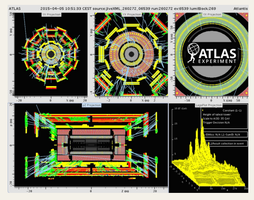


First runs in 2015

New Data Flow system was used in ATLAS cosmics runs, first beam splashes runs and in first 13TeV collisions runs in 2015







Conclusions

For LHC Run II, following new requirements, ATLAS Data Flow system has undergo a major upgrade in terms of architecture and used technologies, thus providing higher performance and data-selection flexibility for ATLAS TDAQ.
The system was validated on the test-bed, during ATLAS technical and cosmics runs and in these days it is taking physics data in 2015 runs.

References

[1] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider. 2008 JINST 3 S08003 1-407

[2] ATLAS Collaboration, The C-RORC PCIe Card and its Application in the ALICE and ATLAS Experiments. https://cds.cern.ch/record/1954495

[3] ASIO library, https://think-async.com/

[4] Intel TBB https://www.threadingbuildingblocks.org/

[5] ATLAS Collaboration,Data-flow performance optimization on unreliable networks: the ATLAS data-acquisition case, J. Phys.: Conf. Ser. 523 (2014) 012019

[6] Battaglia, Andreas; Beck, H.P.; Dobson, M.; Gadomski, Szymon; Kordas, K.; Vandelli, W., The Data-Logging System of the Trigger and Data Acquisition for the ATLAS Experiment at CERN, Nuclear Science, IEEE Transactions on , vol.55, no.5, pp.2607,2612, Oct. 2008