Data Acquisition for the ATLAS Inner Tracker beyond 2026

...

Jens Dopke for the ATLAS ITk Collaboration

Thoughts/Comments

Science and Technology Facilities Council

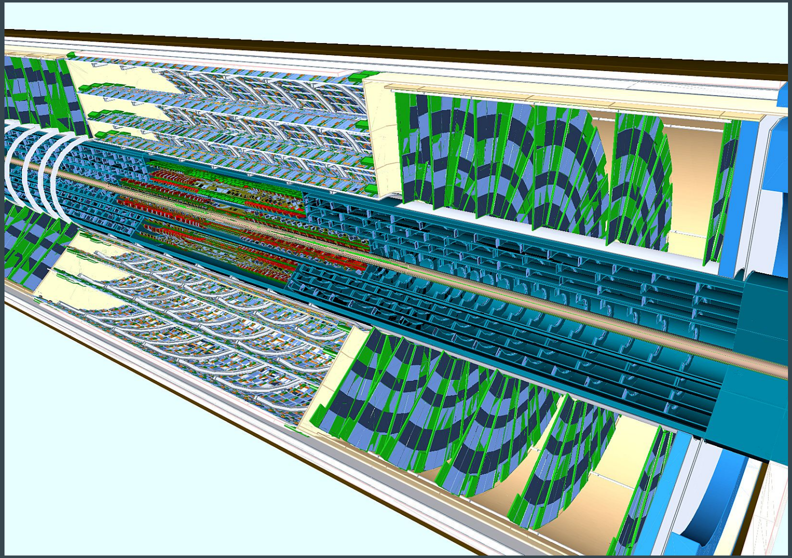
ATLAS EXPERIMENT

General statements

- Past experience has shown that major problems lie in setting up a reliable data acquisition system
  - Detector Data-flow needs to be understood
  - Timing needs to be understood
  - Off-detector Data-flow needs to be understood
- Any of these doesn’t work and you
  - Don’t get your data
  - Get data, but not the data you wanted
  - Get too much, but still wrong data
- Non-functional DAQ can cause off-times of hours until problems are solved
- Trend seems to be towards centralised DAQ systems, based on common hardware and custom software

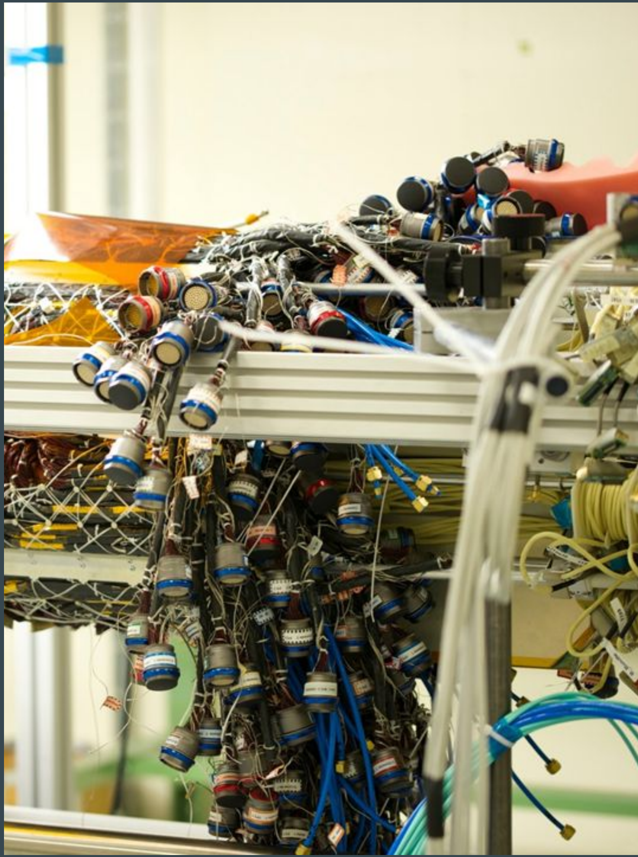
The Phase II ATLAS Inner Tracker (Just a reminder)

- All silicon tracker system
  - 5 pixelised barrels
  - 4 double sided strip barrels
  - Lots of endcaps (no space there)
- Minbias guesstimate for # of tracks:
  - On the order of 10k
  - Strip barrel expects about 1% Occupancy
  - Data rate out of each strip Hybrid Controller: 640Mb/s
  - Total data rate out of strip detector: O(20 Tb/s)
  - Pixels: O(Gb/s) per module, O(10k) modules
- Trigger rate scenarios:
  - 1 MHz, full readout
  - ~4MHz/~600kHz
    - Full rate regional readout followed by track reconstruction and later readout of full detector



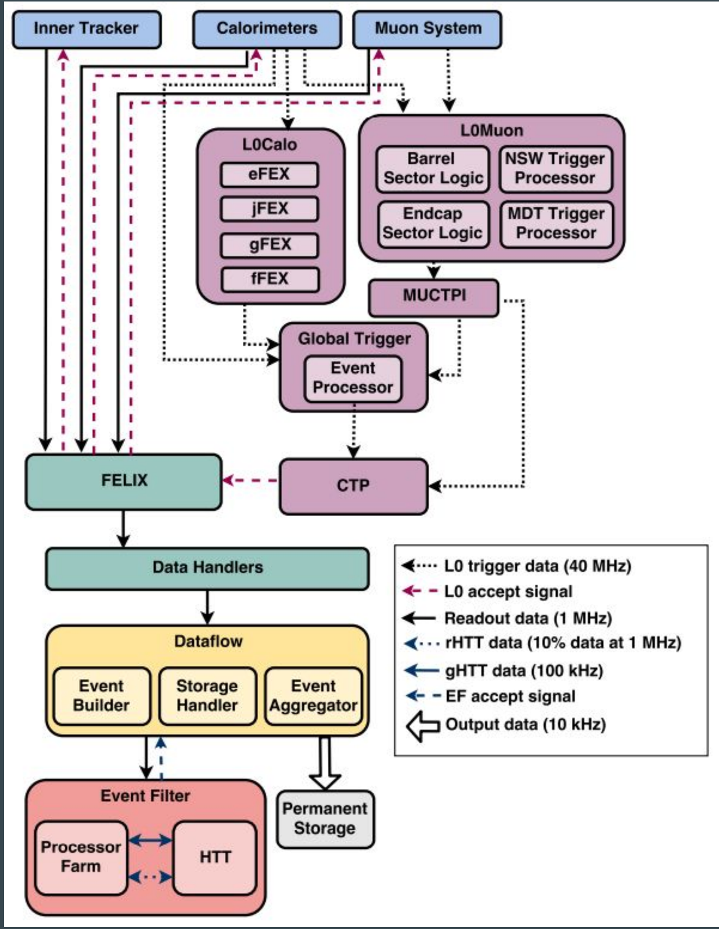
DAQ - what is required

- Connection to and from the detector
    - Optical preferred for bandwidth/material/electronic reasons
- Command and configuration
- Phase adjustment with respect to the machine
    - Could be done for objects, rather than per hit...
- Data connections
    - Require data comprehension eventually (i.e. decoding)
    - Error detection and tracking
- Triggers
- Buffering somewhere down the line
    - De-randomizing?



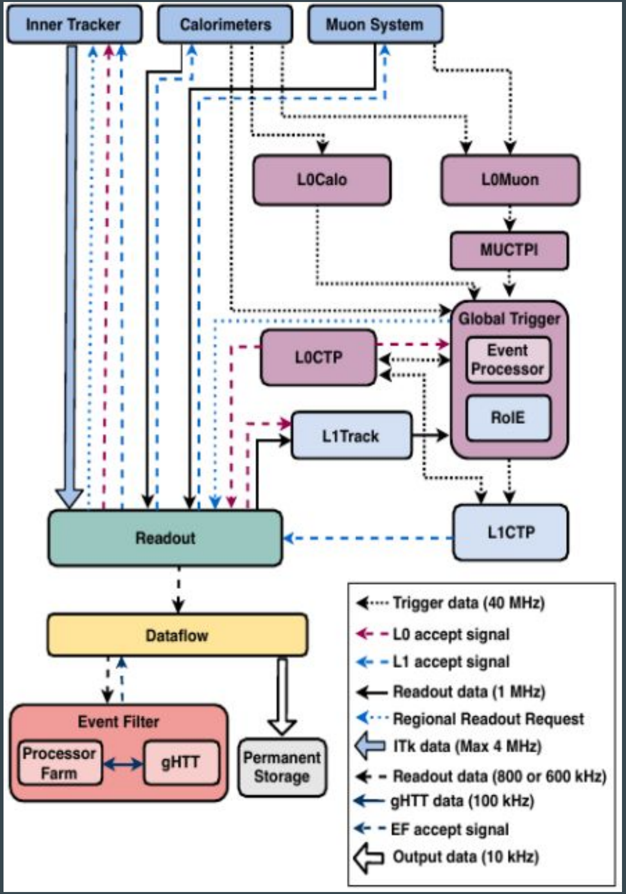
ITk Trigger DAQ (TDAQ) in a Nutshell

- Read more in: CERN-LHCC-2017-020
- Full Readout with L0 trigger:
  - Data gets stored locally by detector FEs
  - Fast detectors generate (through an elaborate trigger system) an L0 accept signal
  - Slow detector buffers get read out (1MHz)
  - Processing through event filter farm with 10kHz rate to disk
- Tracker readout is fully passive, i.e. all data sent out on reception of that level 0 trigger - no intelligence required
  - Buffer size maximum 10us



ITk Trigger DAQ (TDAQ) in a Nutshell

- Staged trigger system:
  - Fast detectors generate the same L0 accept (up to 4MHz) followed by a region of interest readout (about 10%) of slow detectors
  - A fast track trigger system generates a second level decision about the event and an L1 accept is generated for all detectors (>=600kHz)
  - Event filter kicks in again, generating an output of about 10kHz
- Buffer Latencies:
  - Up to 35us for L1 arrival



What about calibration?

- Much of initial operation is in calibration:
  - Bigger “Events” with lots of hits (Pixels: a few thousand per module per event)
  - Synchronised internally (at least in the usual setup today)
- Prototyping started off from FPGA test boards, e.g. Digilent Inc. ATLYS/Nexys Video
  - Allow testing of single chips and modules, parallel operation of multiple modules is possible, but stretches the systems
- Through some CERN custom cards (SPEC, GLIB) but also higher end FPGA test boards, e.g. GENESYS 2, Trenz TEF1001
  - Allow to communicate with a large objects (up to O(50) modules) through few high speed links (O(10Gb/s))or with Pixel Modules (Gb/s links)
  - Starting to understand how to use full designs to communicate with individual modules and what problems might arise
- Idea is to build a generic FPGA based PCIe board (FELIX project)
  - Local memory on board, transfer out via PCIe into PC memory, then shipping off via commodity network hardware

Summary

- Much of the ATLAS ITk DAQ Design is driven by some of the constraints:
  - Total expected data rate
  - Radiation hardness and SEU rate
  - Trigger scheme
  - Availability and functionality of high-speed transceivers/aggregators
- Would we do the same for an electron type machine?

ATLAS Slides are over

Personal Thoughts towards an e+e- collider DAQ

Why even talk about ATLAS ITk DAQ at this workshop?

- Situation at an e+e- collider vastly different:
  - Lower track count
  - Far less radiation damage, O(100-1000) NIEL, O(10) TID
  - Repetition rate low for high occupancy physics
- Doors are open with regards to DAQ
  - Triggerless tracker?
  - Overlapping events (out-of-time)?

Requirements/Need-to-know(s)

- Backgrounds in tracker
  - Need an estimate for the total #hits per layer, probably only vertexer? (Given estimates in the CDR for Beam-Beam interaction, standard events would hold somewhere around O(10) tracks, need to cope with those in data)
- nTracks in tracker
  - CDR - All in all mostly background, some events:
    - Somewhere around 30 hits/event/Layer need to leave the detector, call it 100 for good measure
- Per track timing?
  - Do we need per hit? Depends on trigger setup/other detectors and computing model
- Trigger scenario
  - Do you want to trigger or go full software (or even full recording?)
  - Peak rates may be somewhere around Gbit/s but sustained rate seems more like Mbit/s
- **Calibration: How often do you want to calibrate, how much data does that take - probably a bigger limit on your DAQ system than actual data taking**
- Other subsystems: Can most detectors be fully read out, or would one subsystem imply an early trigger for raw data - not worth it to consider untriggered readout for tracker if other detectors imply reduced rate very early

Things to think about soon

Probably already done, but I am ignorant and will just say it anyways:
Less ignorant after a brief check of cross-sections...

- Data format:
  - Reduce transmission rate ?
  - Ease of processing (stick to byte boundaries, give bits a meaning, at least in unencoded data)
  - Stable transmission (any type of encoding, some sort of identifier, in particular in a triggered scenario)
- Geometry:
  - If outer tracker doesn't need pixels, make them strips (or strip-like, currently foreseen, afaik)
  - Even if pixels are easy enough to provide, reduce the amount of information transmitted, pixels can be strips ...
- Other detectors/software only trigger?

Things to not worry about too much:

- Testing DAQ:
    - More than one system around
        - Personally a big fan of YARR, mostly software defined readout
    - Work towards a final DAQ however would benefit from thinking about a “module” concept at an early stage - what is the essential unit of data coming back, and a system for processing that unit
- Total data rate:
    - Current prototypes for CMOS detectors send around 1Gbit/s of data out, depending on detector concept, that ends up being somewhere around 0.25 hits/cm^2 down to 0.01 hits/cm^2
- Background suppression:
    - High speed links can deliver empty data, FIFO being filled on an off-detector FPGA with actual data - rather, think about event formats/de-randomising - where do you want your data to end up, how will it get there -> Services
- Generally need to start a top-down holistic view - what do you want?