ATLAS Trigger and DAQ Projects

Rainer Bartoldus

Outline

- History of SLAC Involvement
- Brief Introduction to ATLAS High Level Trigger (HLT) and DAQ
- Areas of SLAC Leadership
  - **HLT Configuration Infrastructure**
    - How to (re)configure 1000s of trigger processors without losing beam time
  - **Partial Event Building**
    - How to record much higher rates of calibration events during physics without breaking the budget
  - **Online Beamspot Measurement**
    - How to measure the LHC luminous region online and feed it back to the machine operators and the High Level Trigger
  - **Cathode Strip Chambers (CSC) Read-Out Drivers**
    - How to help a system in trouble and learn from it for the upgrade
- Short List of Other Activities and Contributions
- Conclusions

History of SLAC Involvement

* ATLAS High Level Trigger and Data Acquisition
  - Presented itself as one of the areas with critical needs
  - Ideal match to experience/expertise of SLAC staff (*e.g.* BaBar)
  - Significant work carried out even before we joined in June 2006
* TDAQ Projects
  - Started by taking on one of the most critical technical projects with the HLT configuration (scalability)
  - Consequently involved in the commissioning of the HLT farm and its scaling from 4 to today's 27 racks (>800 PCs)
  - Expanded into a wider portfolio of Trigger and DAQ projects, event building, HLT algorithms, online beam spot

Scientific and Technical Staff: Rainer Bartoldus, Philippe Grenier, Andy Haas, Andy Salnikov, Su Dong

Postdocs and Students: Ignacio Aracena, Sarah Demers, David Miller, Dan Silverstein

ATLAS Trigger

Three-level hierarchy:

64 TB/s

40 MHz

– **Level 1:** (Region-of-Interest): Identify high-p_T lepton or jet candidates based on coarse information from calorimeter and muon chambers; 2.5 μs latency

160 GB/s

100 kHz

– **Level 2:** Use L1 RoI (η,φ) as seed to guide reconstruction; typically 2% of the event is read at full granularity; 40 ms latency

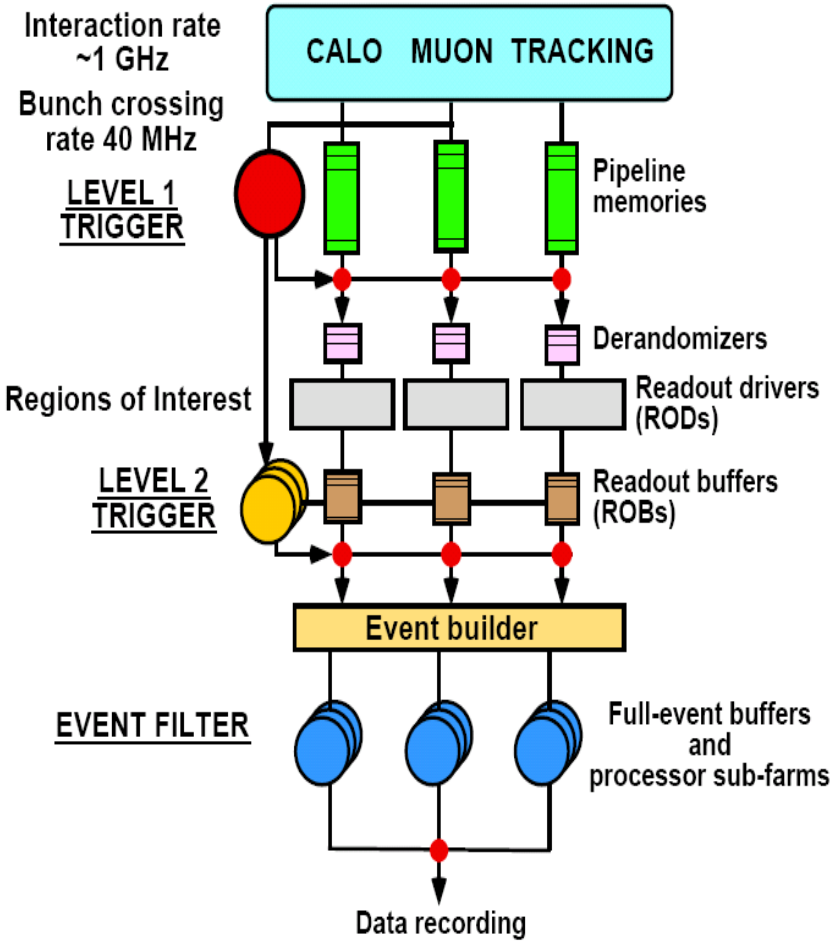
3-6 GB/s

3.5 kHz

– **Event Filter:** Processes fully built events to apply offline-like algorithms; ~1 s latency

300 MB/s

200 Hz



HLT Farm (part of it)

27 racks of XPUs

35% of final system

+ all infra-structure machines

Central File Servers (CFS)

Local File Servers (LFS)

1 per rack

L2/EF Processing Units (XPU)

31 per rack

Dual-Quad Core 2.5 GHz 16 GB "Harpertown"

High Level Trigger (HLT) Configuration

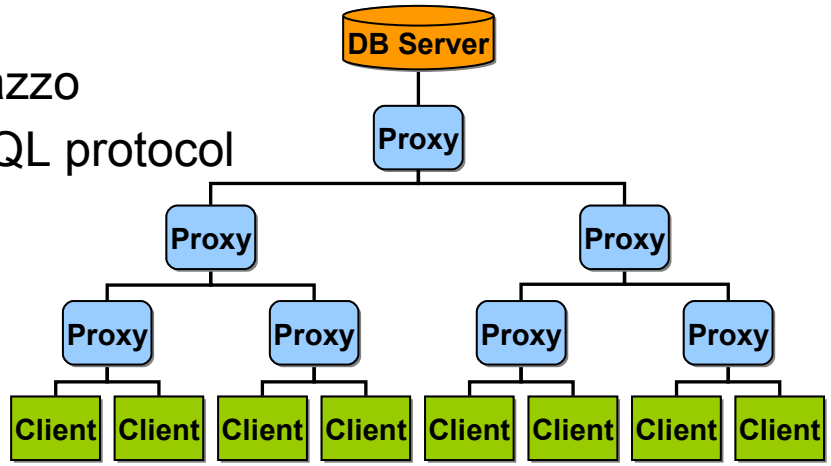
High Level Trigger (HLT) Configuration

* Configuration Challenge
  - When we joined ATLAS TDAQ three years ago, one of the critical path items was the distribution of configuration data to the thousands of HLT clients
    - With O(2000) nodes (400 L2 + 1600 EF) x 8 cores/node x 1 client/core x O(10-100) MB/client, the system has to generate and **deliver O(0.1-1) TB of data within O(10) s**
  - No single server can handle that kind of a load
    - Not feasible to handle 16000 connections to begin with...
    - Even if one managed to stagger them, and assuming 70 MB/s (unrealistic because of small packet/round trip overhead), it could take > **6 hours to configure the entire system**
      - Even if one compressed the data to 10 MB it would still be 40 minutes between making a change and being ready to run
    - Clearly, the trigger menu, prescales etc., will have to be adjusted more frequently, in particular at the beginning

→ Needed to find a way to turn the configuration around fast

DbProxy

* Strategy
  - Must reduce the number of connections (→ via *multiplexing*) and the network traffic (→ via *caching*)
  - In principle: add more servers, and bring them closer to the clients
* "DbProxy" Solution
  - Originally developed by Amedeo Perazzo (now LCLS), based on the open MySQL protocol
  - Successfully deployed in ATLAS TDAQ in 2007
  - Since then an integral part of technical runs (TDAQ only) and commissioning periods/combined runs (with all subdetectors)
    - Early running was against a MySQL server, or MySQL replicas of the ORACLE master



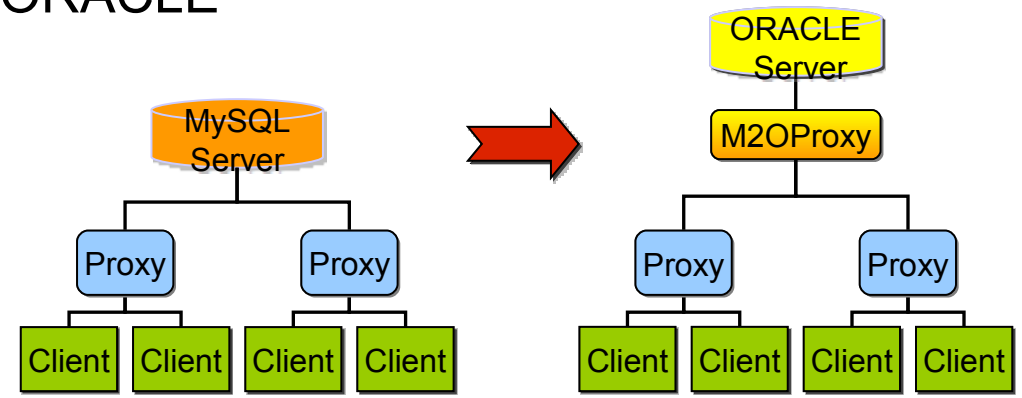
This solution enabled HLT commissioning and scaling from 4 to 27 racks

CORAL Server

* Background
  - LHC experiments use a relational abstraction layer called CORAL (C++ client API to relational DBs)
  - This lets the client select at run-time between various technology plugins: MySQL, Oracle, SQLite, FroNTier
  - However, such a client plugin renders the proxy hierarchy technology-dependent, as in our current MySQL implementation
    - ORACLE's closed-source (proprietary) protocol prohibits server implementations
  - We decided the best long-term solution was to push the technology choice behind the proxy tree, *i.e.*, define an independent protocol
* Joint Project with CERN/IT
  - We started a dialog with our CERN/IT friends, who had arrived at the same idea for independent reasons, mostly concerning security
  - So we launched a joint CORAL Server Project in the fall of 2007 that will solve this

“M2O” Translator Proxy

- ATLAS needed an immediate solution when the online database server was switched to ORACLE
- We managed to put a temporary solution in place and developed a “MySQL-to-Oracle” bridging proxy (Andy Salnikov)



- This was implemented on a very short timescale and successfully deployed in summer 2008
  - It has received attention (and praise) from experts who looked for (or had tried) something similar
- We don't consider it a permanent solution, due to its dependence on versions and non-standard type conversions
- However, it bought valuable time for the CORAL project, enabled HLT commissioning, and *it has been performing extremely well*

Proxy Configuration Performance

* Translation
  - Our early tests already established that configuration through the M2O proxy can in fact be *faster* than a direct ORACLE connection
    - This has to do with CORAL protocol overhead, due to schema discovery, which is effectively being cached by the M2O proxy
* Scaling
  - The infrastructure of the distributed proxy tree with the translator on top makes the entire farm behave like a single client (our design goal)
    - It makes 10000 clients all see a local MySQL server, while the ORACLE server sees a single client (*i.e.*, *one* L2 and *one* EF client)
  - It has been demonstrated that the configuration of the farm takes as long as the configuration of a single node (→ Scaling)

Partial Event Building

Event Building Capacity

- A fully built, raw ATLAS ("byte stream") event is ~1.6 MB
  - Including data from all 1600 Read Out Buffers (ROBs) of O(1k) each
  - Grows to ~3.5 MB when reading out 5 rather than 2 LAr samples
- Each event building node (SFI) handles 70 MB/s → 80 SFIs can sustain 5600 MB/s total I/O
  - At 1.6 MB this allows up to 3.5 kHz L2 accept rate
  - Generally the available bandwidth is:
    - Number of ROBs x fragment size per ROB x L2 accept rate
  - By reducing the number of ROBs one can gain in L2 accept rate
- For calibration events it is typically not necessary to read out the entire detector, but just a subdetector or region of interest
  - *e.g.*: no need to read out the LAr calorimeter (more than half of all ROBs) for Inner Detector (ID) alignment
- Can afford substantially higher rates by building partial events *during physics data taking* (depending on the trigger type)

Partial Event Build Applications

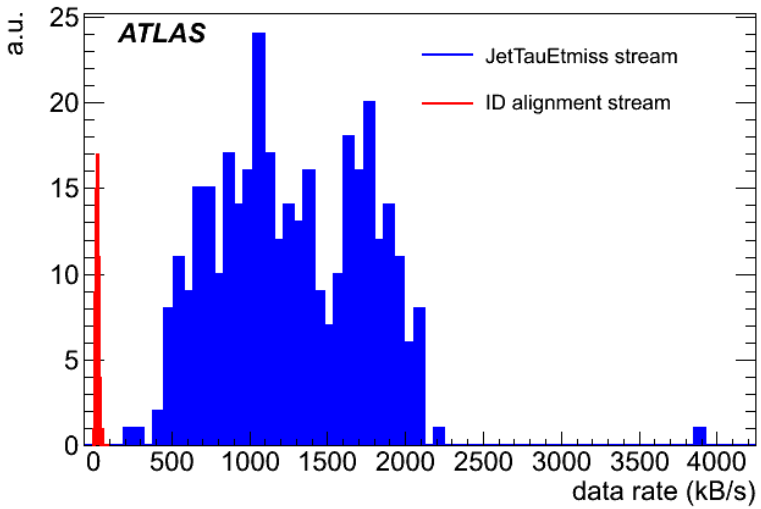
* Inner Detector Alignment
  - Needs 6-8 M isolated tracks in 6 hours for Pixel/SCT and in 24 hours for TRT
  - Uses 50 Hz of isolated track trigger, ~30 kB/event
    - This would be 25% of the nominal ATLAS logging rate, but at < 30kB per event it is only 0.5% of the bandwidth!
* LAr Calibration
  - Study pulse shape of individual cells
  - Requires 5 Hz to achieve precision of <1%
  - Select calibration events using photon triggers, ~50kB/event

All practically sole responsibility of Ignacio Aracena

* Tile Calorimeter
  - Illuminate full calorimeter with dedicated calibration system (laser pulser)
  - Runs during empty bunch crossings
  - Uses dedicated L1 triggers, ~230 kB/event

Partial Event Building (cont)

* Validation
  - Offline tests on pre-series farm with enhanced bias events (L1)
  - Compare ID alignment with Jet/Tau/EtMiss physics stream, (exclusive rates) →



* Ongoing Development
  - Working on monitoring of the partial event build
    - Provide diagnostics at various levels in the dataflow chain
    - Allows to determine “savings” compared to full build
      - Help to optimize bandwidth for each subsystem
  - Implement additional algorithms
    - Working with Muon System on muon alignment sample
  - Our aim is to help more subsystems benefit from this capability

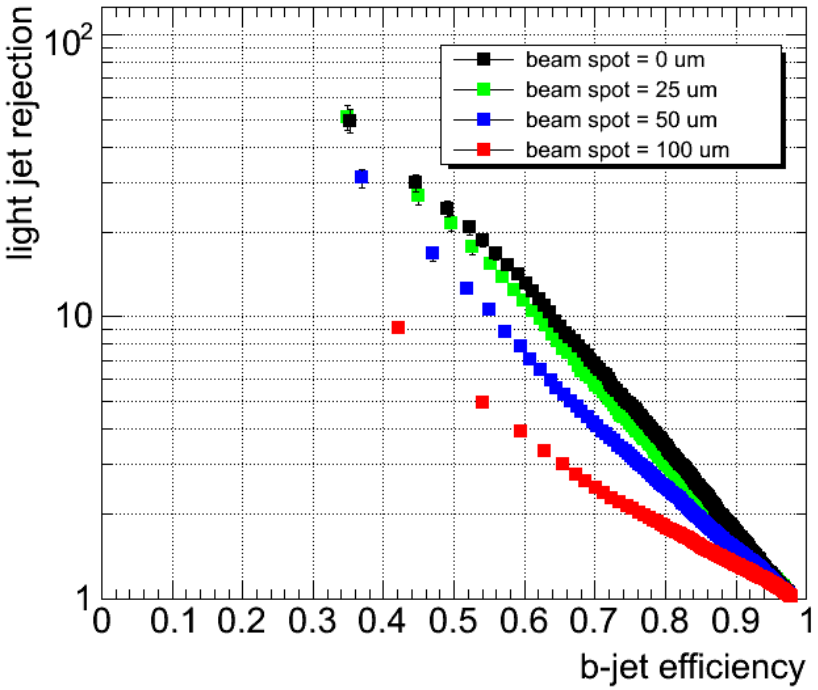
Online Beam Spot Measurement

Online Beam Spot Measurement

* Machine parameters (sizes and angle)
  - At design luminosity (10^{34} cm^{-2} s^{-1}): σ_{xy} ~ 17 μm
  - Startup pilot run (10^{31} cm^{-2} s^{-1}, β^* ~ 4 m): σ_{xy} ~ 45 μm
  - Luminous region length σ_z ~ 5.4 cm
  - Crossing angle: 0.3 mrad
* Variations during running (in position)
  - Expected to be ± 300-600 μm **during running** for the first 1-2 months
  - Eventually down to ± 30-50 μm under stable beams
  - Still ± O(1mm) jumps possible between fills
* Measurement with the ATLAS HLT
  - Useful in many respects, especially as feedback to LHC operators
  - *Critical* for algorithms that depend on IP position such as *b*-tagging

Beam Spot Information in the HLT

* HLT Algorithms Depending on the Interaction Point
  - The HLT can not only be source for the beam position, it also needs to know it in algorithms such as *b*-tagging
  - So far, the *b*-tagging slices and performance studies have assumed (0,0,z)
  - *b*-jet efficiency/light-jet rejection are much degraded already at shifts of O(50 μm)
  - It is apparent that the HLT beam position must be bootstrapped at the beginning of each fill, and likely also be updated during the run
  - This is a critical need for the experiment which has simply been overlooked until we started working on this in 2008



Beam Spot Algorithms

Two complementary algorithms are available

* Track-based:
  - A. Cerri (CERN) implementing CDF L2 "d_0 versus φ" algorithm: extract IP position from amplitude and phase of sinusoidal shape
* Vertex-based:
  - D. Miller (SLAC) successfully implemented an event-by-event vertex method
* Performance
  - The vertex algorithm has been demonstrated to yield 20-30 μm **per vertex** (*i.e.* times $1/\sqrt{N}$) with timing of < 300 μs running parasitically (feeding off already fitted tracks), well suited for L2
    - Assuming 25 Hz of usable events, this would result in a few μm position measurement every few seconds, and spot sizes and angles to similar precision within a few 10s to 100s of seconds

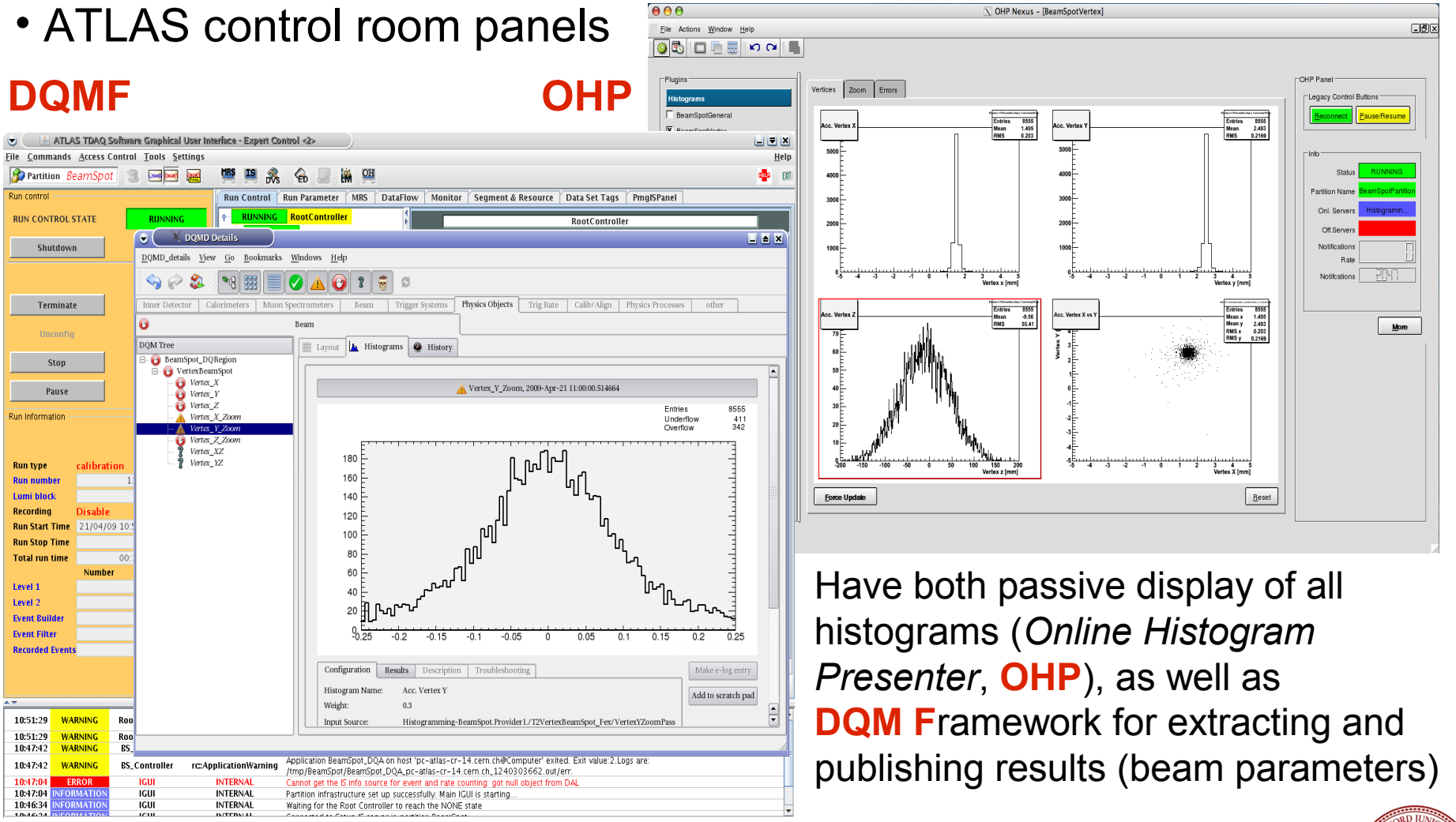
Beam Spot Technical Infrastructure

* Collecting the information from the farm
  - Building on existing infrastructure for online monitoring but need faster feedback than for bulk of histograms, O(1000) per client
  - Established a dedicated “Express Gatherer” to serve just beamspot and perhaps related information (*e.g.* online luminosity)
* Feeding it back to the machine
  - Information to be extracted from histograms, translated to ATLAS slow control and pushed into standard LHC interface (called DIP)
* Feeding it back into the HLT (ambitious)
  - This is technically the most difficult part; we are currently exploring a scheme involving the Central Trigger Processor (CTP) fragment telling the L2 processes to fetch new parameters from the database
  - This would take advantage of the proxy infrastructure and could be applied to update the beamspot, HLT prescale factors or dead/noisy channel masks as well

Online Monitoring in ATLAS Control Room

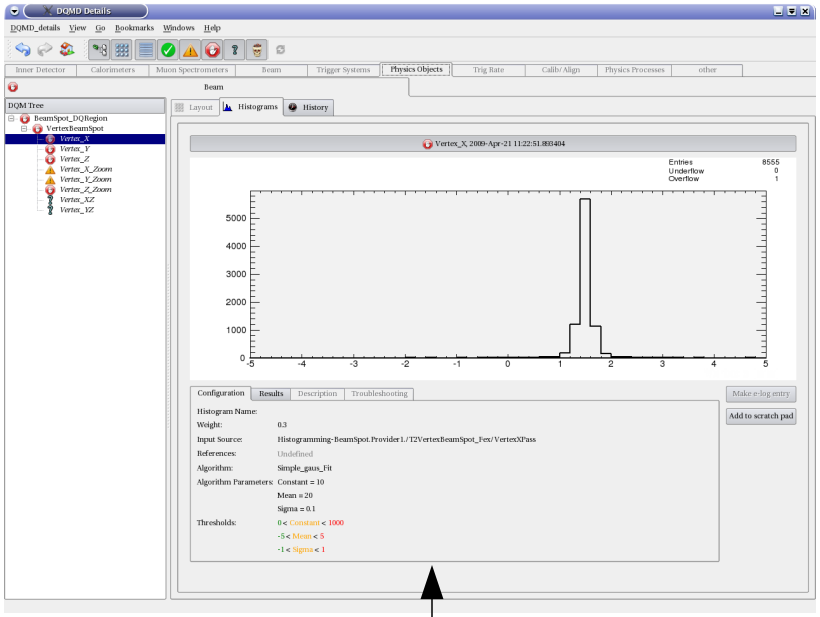
- ATLAS control room panels

DQMF **OHP**

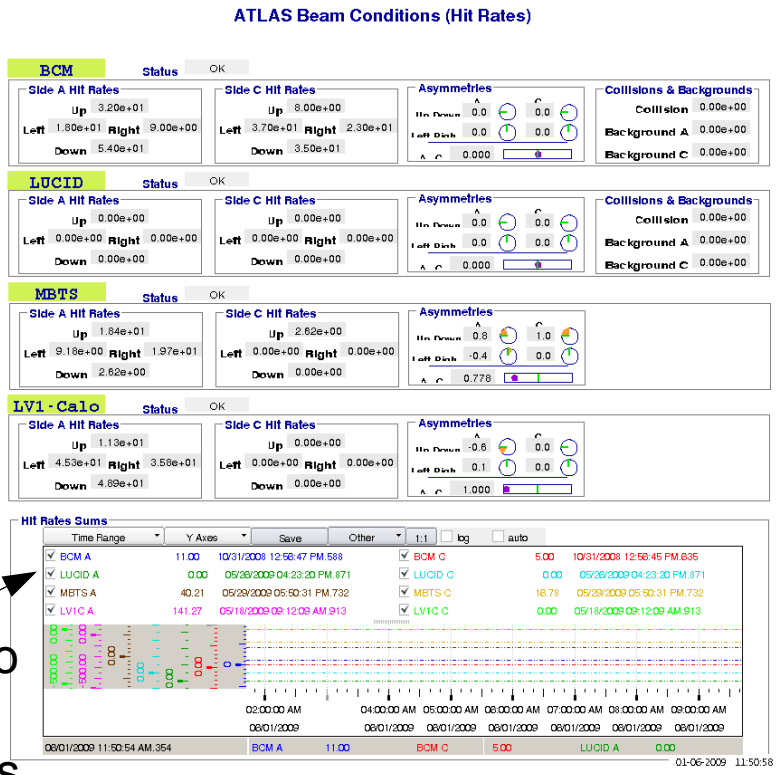


Have both passive display of all histograms (*Online Histogram Presenter*, **OHP**), as well as **DQM F**ramework for extracting and publishing results (beam parameters)

Delivering Beam Position Information to the LHC



HLT beamspot to be added to this LHC panel

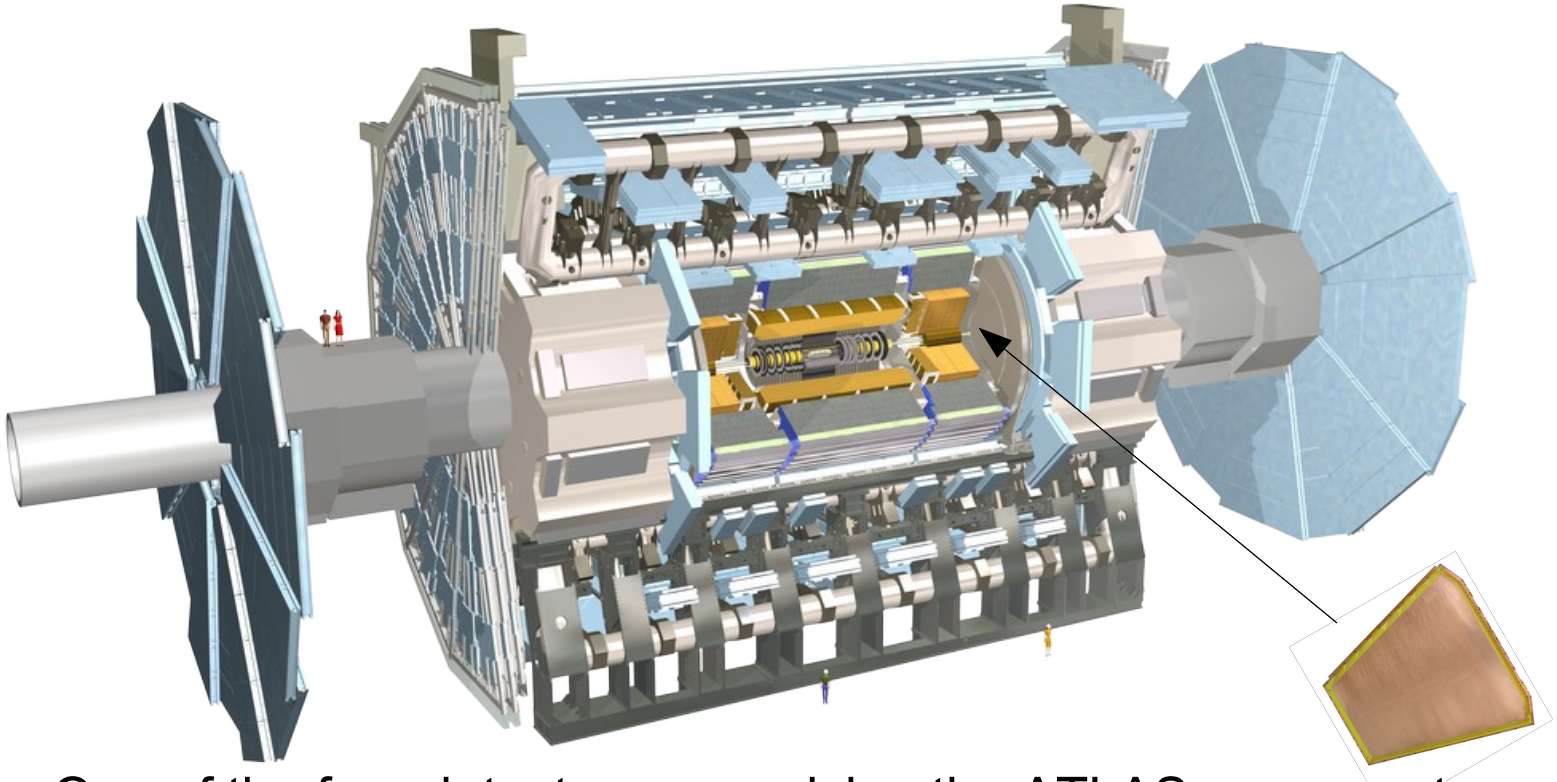


Beam position information is extracted using DQMF and fed into the central ATLAS-LHC communication system (called DIP)

ATLAS beam spot parameters are delivered to the LHC control room along with other info. such as hit rates in luminosity&trigger counters

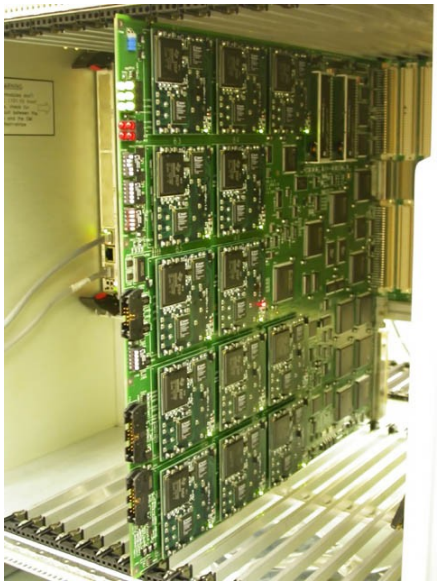
Cathode Strip Chamber (CSC) Read-Out Drivers (ROD)

Cathode Strip Chambers (CSC)



- One of the four detectors comprising the ATLAS muon system
- Proportional chambers, provide precision measurement in the forward region (spatial resolution 60 μm), 67k wires

CSC Read-Out Driver (ROD)



* Read-out problems
  - By last summer, the CSC became the only detector not able to participate in ATLAS combined running
  - RODs would lock up or send corrupted data after only a few external triggers
    - Problem had not been seen with internal trigger
  - Management and US-ATLAS asked for help
* SLAC Involvement
  - Our group was able to respond, first learning about the situation on site, then flying in three of our top DAQ experts (non-ATLAS)
  - Working closely with the CSC team, managed to identify some more obvious problems in FPGA firmware and DSP software
  - This let the system run through more events, exposing further, more subtle issues

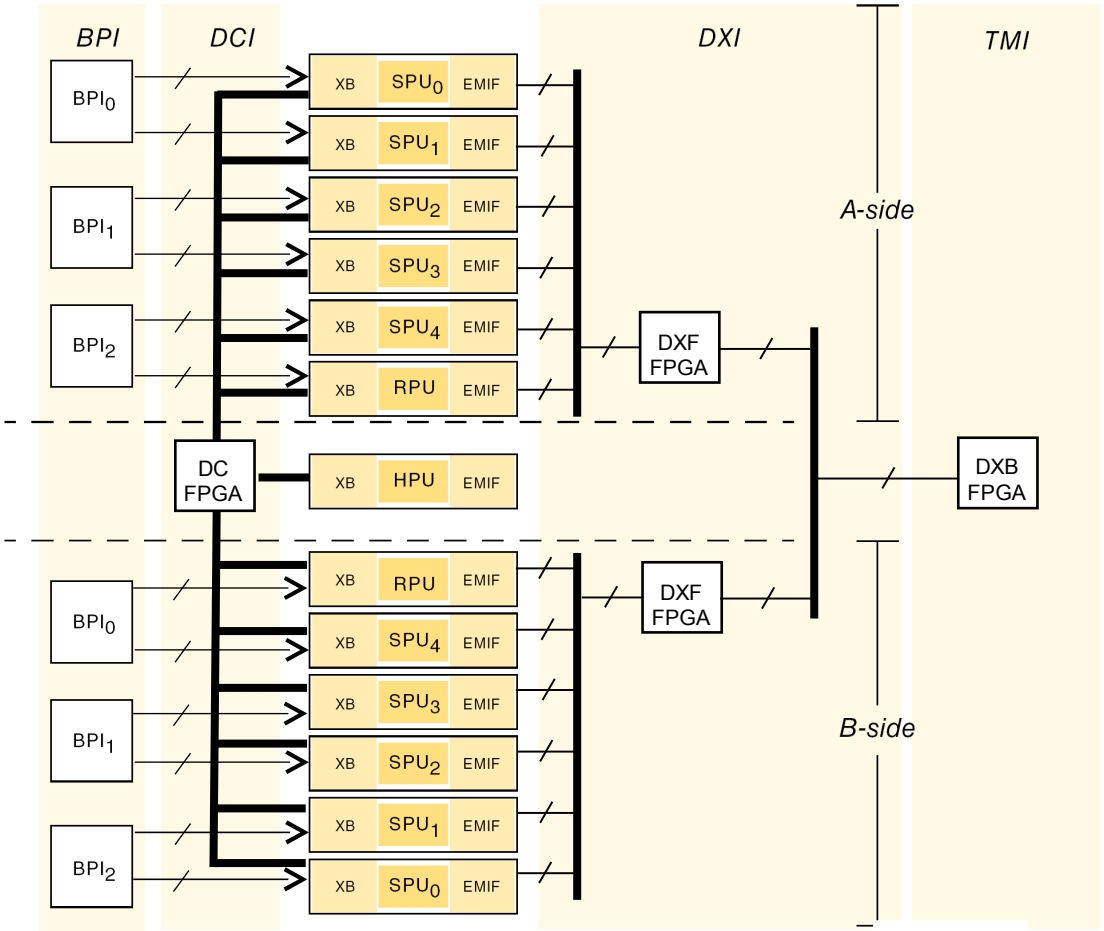
CSC ROD Upgrade

* SLAC put forward a plan to redesign the current system
  - Meeting constraints dictated by the very tight ATLAS schedule
    - No changes to existing hardware: no board modifications, cannot replace FPGAs
    - No changes to interfaces with TDAQ monitoring/control, event structure, calibration
  - Firmware changes
    - Two frame devices and two DMA controllers, Inter-Module-Communication (IMC), hosted on existing FPGAs
  - Software changes
    - DSP software replaced, based on new frame model
    - *Polling* rather than *interrupt-driven* model
  - Increase testability, performance and robustness
  - Went through an external review last February

Technical Staff: Ric Claus, Gunther Haller, Ryan Herbst, Mike Huffer, Jim Panetta, Leonid Sapozhnikov

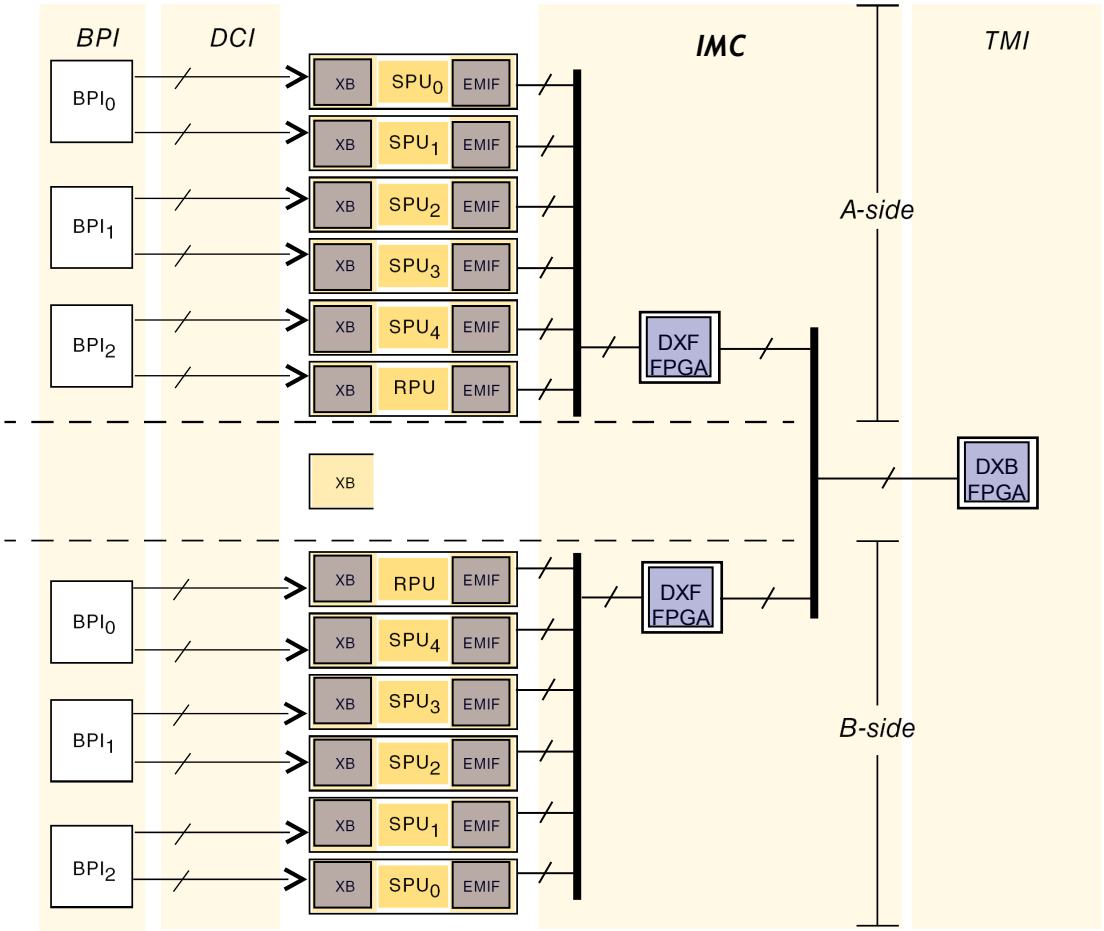
Design Changes

- Virtually all FPGA firmware either modified or replaced
- New Dataflow model
- Control *distributed* rather than *central*
- Data movement in parallel streams
- Event building moved to software
- **Increased performance, reliability, modularity**



Design Changes

- Virtually all FPGA firmware either modified or replaced
- New Dataflow model
- Control *distributed* rather than *central*
- Data movement in parallel streams
- Event building moved to software
- **Increased performance, reliability, modularity**



Next Steps and Building on CSC ROD Experience

- Our rapid engagement was possible due to the expertise of SLAC staff, and their broad experience from different experiments over the years, in DAQ, DSP and FPGA programming
  - This is supporting a university effort and a US-ATLAS responsibility
- The next steps include:
  - Deploying two staff to CERN (3 months/1 year) for commissioning
  - Installing new development environment
  - Completing CERN test stand and regress test
  - Deploying new firmware/software to P1 and regress test
  - Then: calibration s/w, flow control and implement “stop-less” recovery
- Our unexpected involvement with the Muon Read-Out Drivers has the additional benefit of gaining us close familiarity with a concrete ROD system in ATLAS
  - We aim to use this experience as input in guiding our thinking on the phase-II luminosity upgrade, which will demand a new ROD design
  - *cf.* TDAQ upgrade R&D activities in Su Dong's presentation

Other Activities, Contributions

Work on Trigger Algorithms ("Slices")

* *b*-tagging — Ariel Schwartzman, David Miller
  - Developed HLT *b*-tagging algorithm, optimization for hadronic top analysis
  - Nice synergy within SLAC, with our *b*-tagging analysis work, and with our involvement in the Pixel detector
* Jet/Tau/Missing-E_T — Ignacio Aracena
  - Online integration, algorithm improvements
  - LAr Front-End-Buffer jet reconstruction for L2
  - Synergy with our Jet/MET/*b*-tag analysis group
* Tau — Sarah Demers
  - Online integration; monitoring and validation of the tau trigger performance
  - Input from expertise gained in previous experiments

Operational Responsibilities

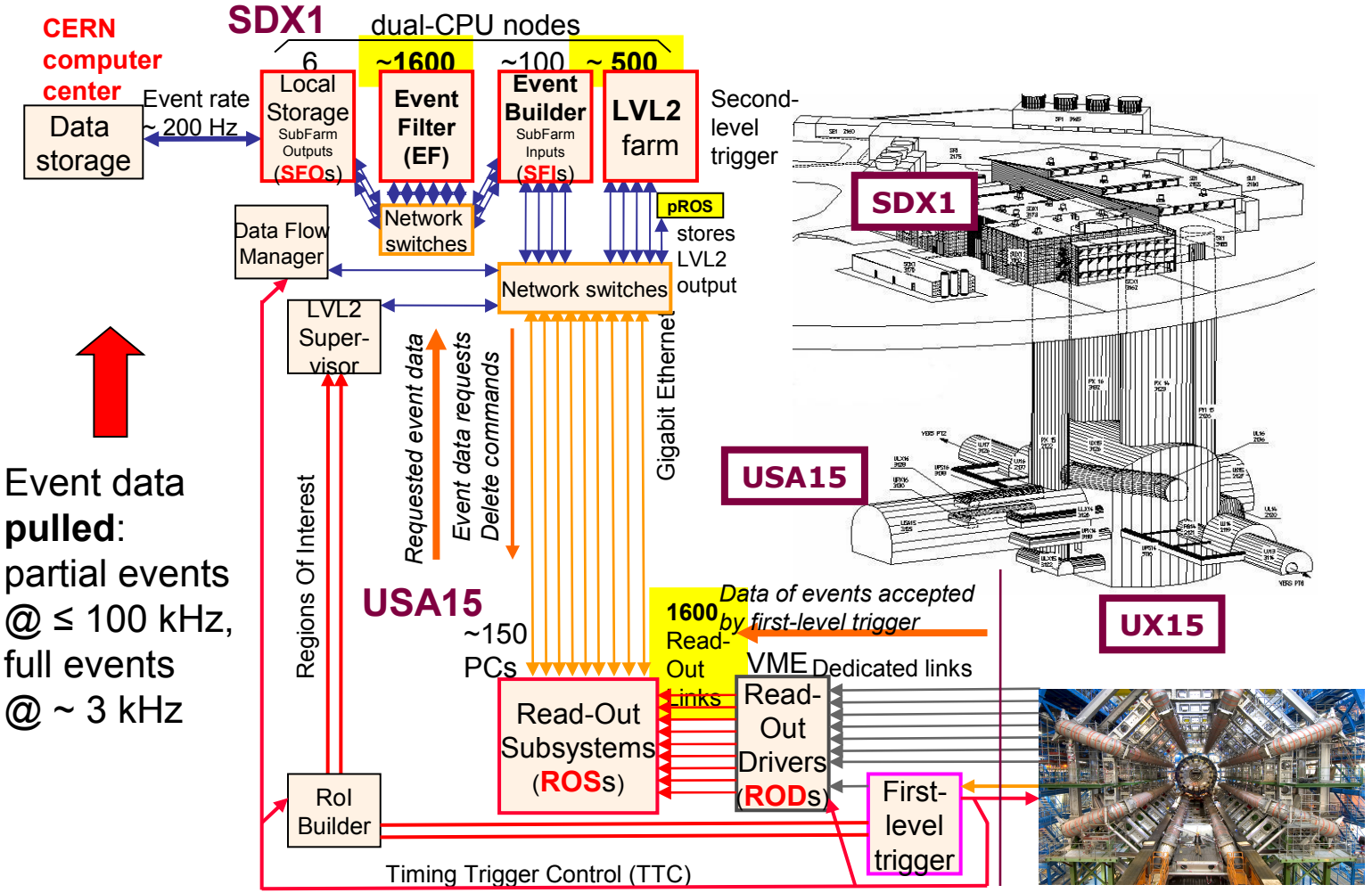
* The SLAC team has taken on a number of online responsibilities:
  - Sole responsibility for the DbProxy technology and the configuration infrastructure, involving conditions, geometry and trigger configuration databases (A. Salnikov, S. Demers, Su D., R.B.)
  - DAQ Partial Event Building (I. Aracena)
  - TDAQ expert status and support for technical runs and commissioning runs; Participation in HLT farm commissioning (S. Demers, A. Salnikov, R.B.)
  - Regularly taking over HLT release validation (I. Aracena)
  - Coordination of Online Beam Spot Project (Su D., R.B.)
* Future areas of responsibilities
  - CSC Read-Out-Drivers (R. Claus and others)

Summary

- With the DB proxy technology, we solved the **HLT configuration** problem in a reliable, efficient and scalable way, leveraging on our expertise in online databases and network programming.
  - The CORAL Server is an ongoing joint project between SLAC and CERN/IT and likely to find applications outside ATLAS
- We have been instrumental in the development and implementation of the **Partial Event Building** capability, which offers subsystems large factors in calibration event rates
- We are the lead of the **Online Beamspot** measurement for ATLAS, which will serve ATLAS monitoring, provide real-time feedback to the LHC operators, and feed back into the HLT itself for configuring beamspot-sensitive algorithms, such as *b*-tagging
- Our engagement in the **CSC ROD** helped the system out of a crisis, enabled commissioning, and culminated in a redesign from which the system should emerge more performant and robust
- Our group has TDAQ expert status and privileges, are part of daily **Operations** and on-call response, and we will be there for first data!

Backup Material

Trigger / DAQ Architecture



Event data **pulled**: partial events @ ≤ 100 kHz, full events @ ~ 3 kHz

Event data **pushed** @ ≤ 100 kHz, 1600 fragments of ~ 1 kByte each

CORAL Server Project

* Time line and Milestones
  - Weekly meetings have been held between the SLAC TDAQ team and the CERN CORAL team from the beginning
  - SLAC led the definition of the new protocol, making sure it meets the requirements of a proxy for the ATLAS online system
  - In spring 2008, the project reached official status within the LCG, gaining endorsements from LHCb and ALICE
  - CERN has implemented the server and the client plugin, SLAC has contributed the new CORAL proxy
  - The project has now entered the debugging phase, and intense testing is now underway with the prototype of the ATLAS HLT
* Impact
  - While this new technology will need time to mature and reach production quality, it addresses problems that are both current and common; we can expect widespread applications, which may well reach beyond the ATLAS/LHC community

Partial Event Building

* L2 Scheme
  - A dedicated L2 algorithm fills a pre-defined list of ROBs
    - May be seeded by an L1 RoI or any given L1 trigger
  - Event Builder sees non-empty list and builds event partially
  - Partial events are not processed by EF but written directly to stream
* EF Scheme
  - Calibration events can also be selected in the EF
    - In this case they must have passed L2 as physics
  - If the EF rejects it as physics, the event is *stripped* and written to the calibration stream
* Overlaps
  - If an event is both physics and calibration in L2, it is fully built once, sent to the EF, and if it is still both, one copy is written to the physics stream, another is stripped and written to the calibration stream

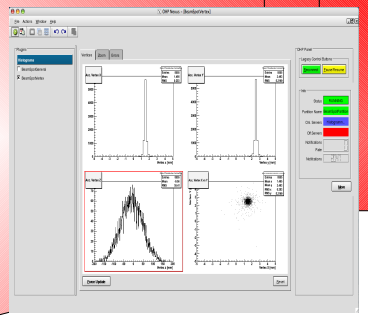
Beamspot Measurement Schema

L2 Trigger Farm

Beamspot Algorithm

Histogrammed beam position parameters

* *Published every luminosity block*
* *Displayed in ACR*



DQM-F

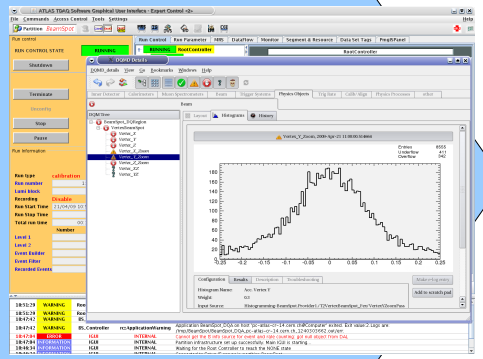
Data Quality Monitoring ***Framework***

* Analyzes histograms
* Publishes results

ATLAS LHC Communication Applications (PVSS)

DQM-D

DQM ***Display*** *in ATLAS Control Room*



LHC Control Room

Beam position parameters Delivered to LHC