ATLAS前後方ミューオントリガー検出器用オンラインシステム

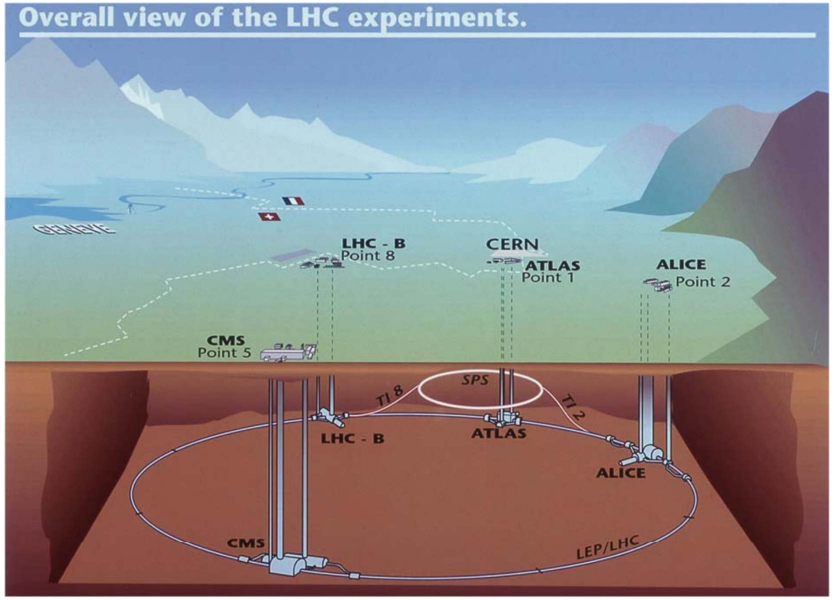
Takashi Matsushita

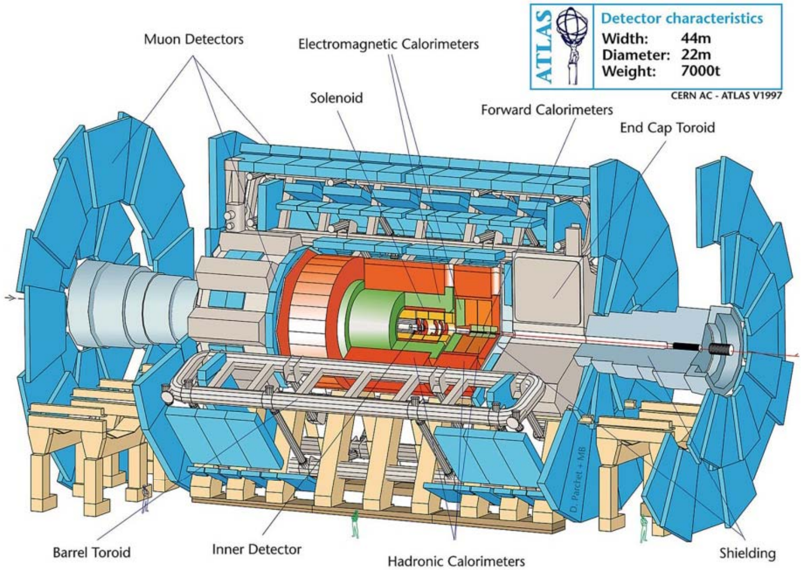
Kobe University

matsushita@phys.sci.kobe-u.ac.jp

松下崇 蔵重久弥 早川俊 中塚洋輝
佐々木修[A] 池野正弘[A] 鈴木友[A] 安芳次[A]
川本辰男[B] 石野雅也[B] 織田勧[B] 久保田隆至[B] 平山翔[B] 金賀史彦[B] 結束晃平[B]
戸本誠[C] 杉本拓也[C] 高橋悠太[C] 奥村恭幸[C] 長谷川慧[C]
菅谷頼仁[D]
福永力[E]
神戸大学 KEK[A] ICEPP[B] 名古屋大学[C] 大阪大学[D] 首都大学[E]

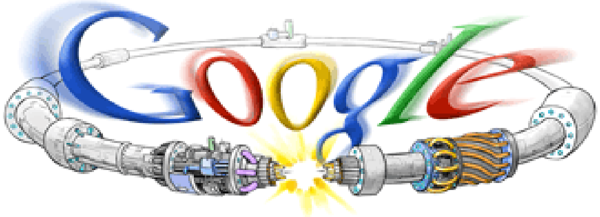
LHC/ATLAS



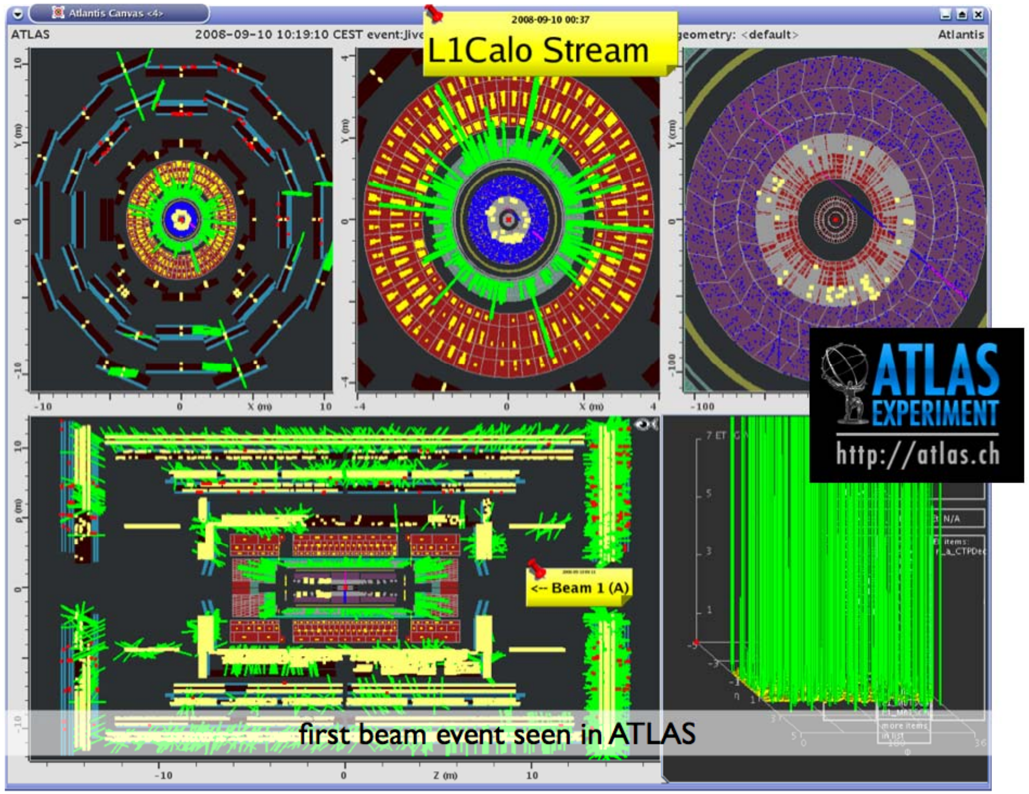


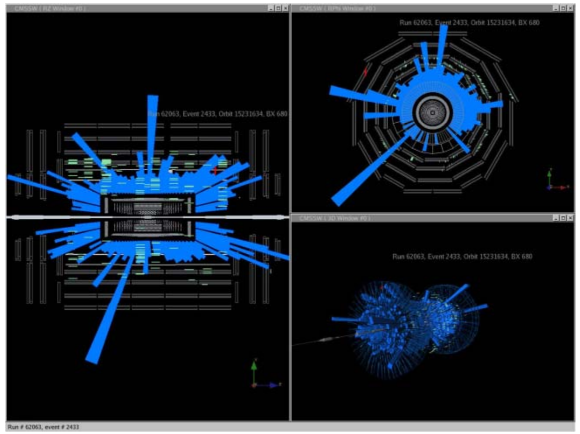
- Explores physics at TeV energy region
- Large Hadron Collider (LHC)
  - proton-proton collisions at 14 TeV
  - circumference = 27 km
  - design luminosity = 10^{34} $cm^{-2}s^{-1}$
  - bunch crossing every 25 ns
- ATLAS
  - observes 1 GHz *p-p* interaction
  - general purpose detector
  - width x diameter = 44 x 22 [m]
  - Tracking $|\eta| < 2.5$ in 2 T solenoid
    - $\sigma/p_T \sim 5\times10^{-4}\ p_T \oplus 0.01$
  - Calorimetry $|\eta| < 4.9$
    - Electro magnetic $\sim 10\%/\sqrt{E}$
    - Hadronic $\sim 50\%/\sqrt{E} \oplus 0.03$ (10 λ)

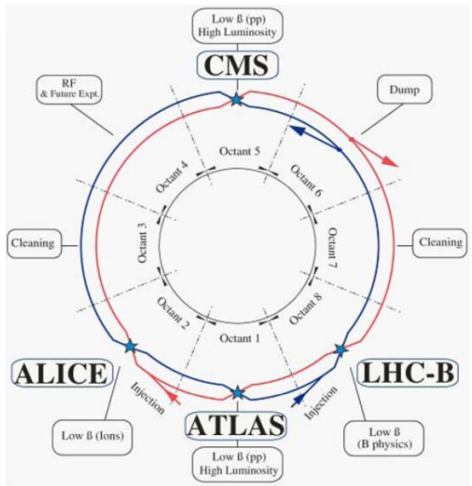
LHC started operation on 10/09/2008



- Man made events observed at both CMS/ATLAS



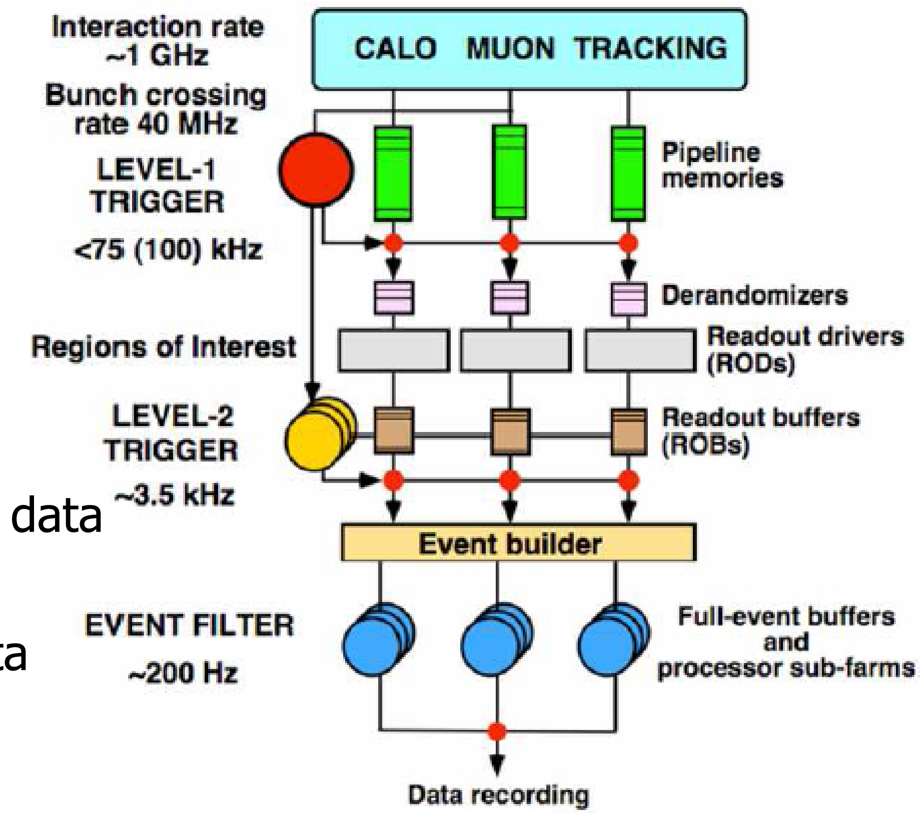




ATLAS trigger system – overview

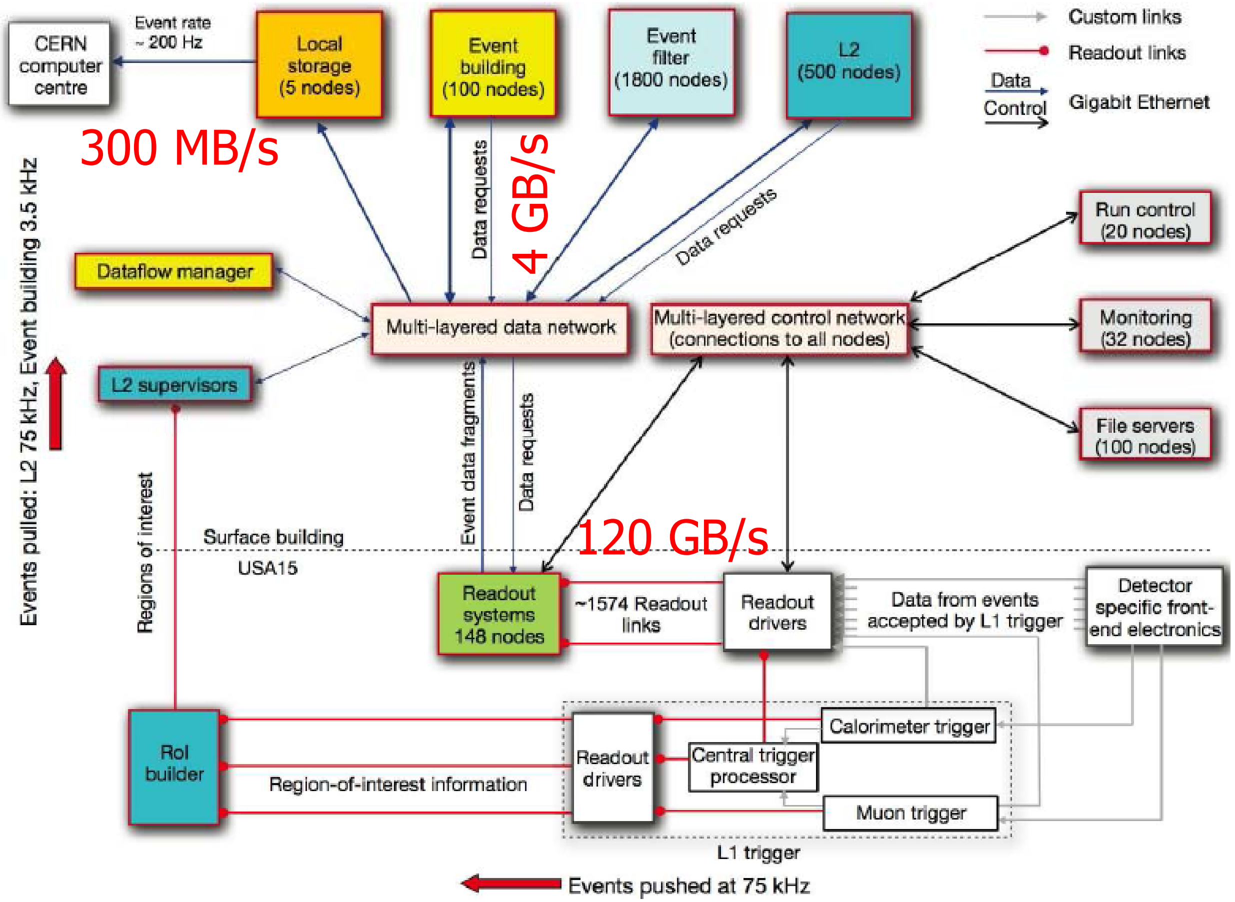
- Reduce event rate from 40 MHz to recordable rate
  - 300 MB/s , event size ~1.5 MB → 200Hz
- Three level trigger with region-of-interest (ROI) based 2nd level trigger

- Level1 trigger – custom built hardware based
  - coarse glanurality
    - calorimeter
    - muon
  - Trigger decision in ~ 100 bunch crossings [2.5 μs]
- Region of interest builder
- High level trigger – software based
  - 2nd level trigger with partial event data
    - Trigger decision in ~ 40 msec
  - 3rd level trigger with full event data
    - Trigger decision in ~ 4 sec



ATLAS read-out system – overview

- Detector
  - front-end electronics with pipeline memories to cope with ~2.5 μs trigger latency
  - read-out drivers
- PC farms
  - read-out system
    - custom built buffers in PC farm
  - event building
    - more PC farm on data network
- DAQ software
  - control, configuration, monitoring on control network



ATLAS Trigger & DAQ framework

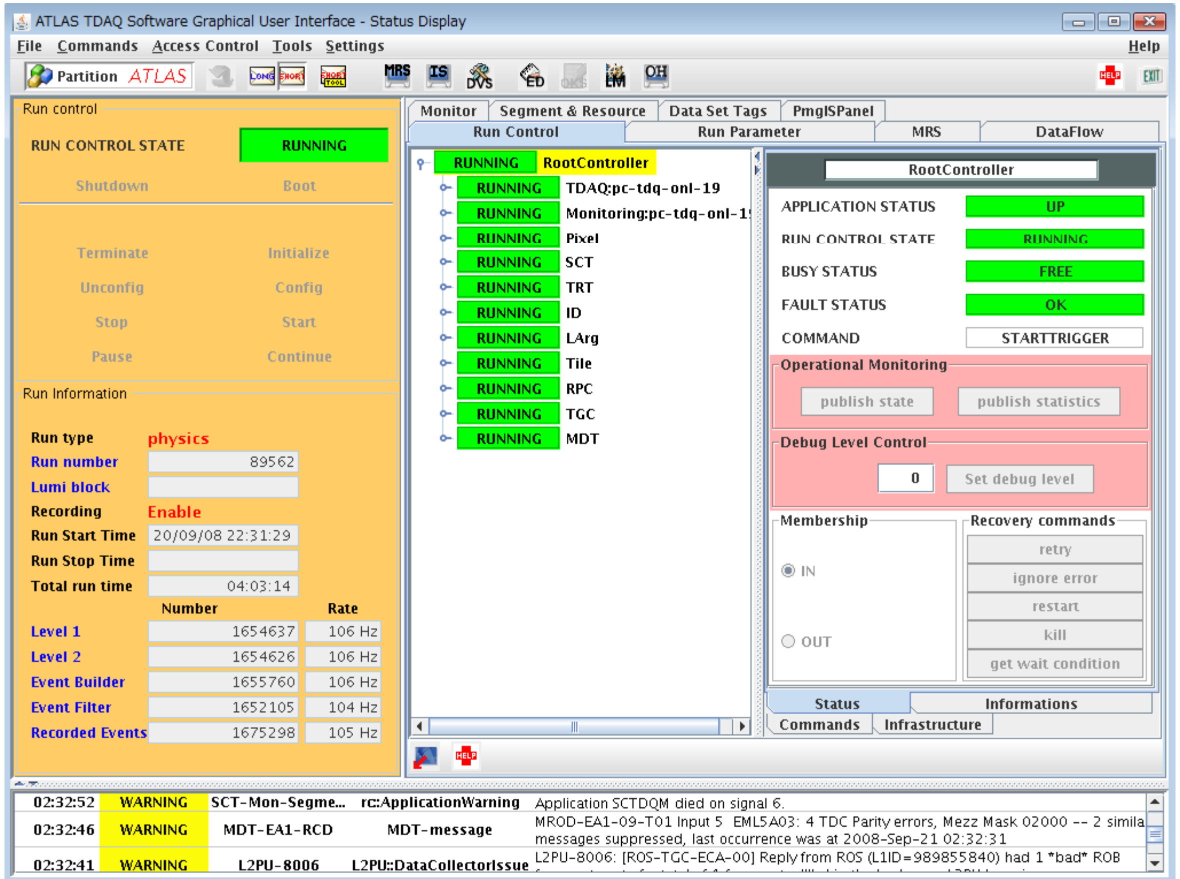
- ATLAS provides framework for developing online software dedicated to each detector component

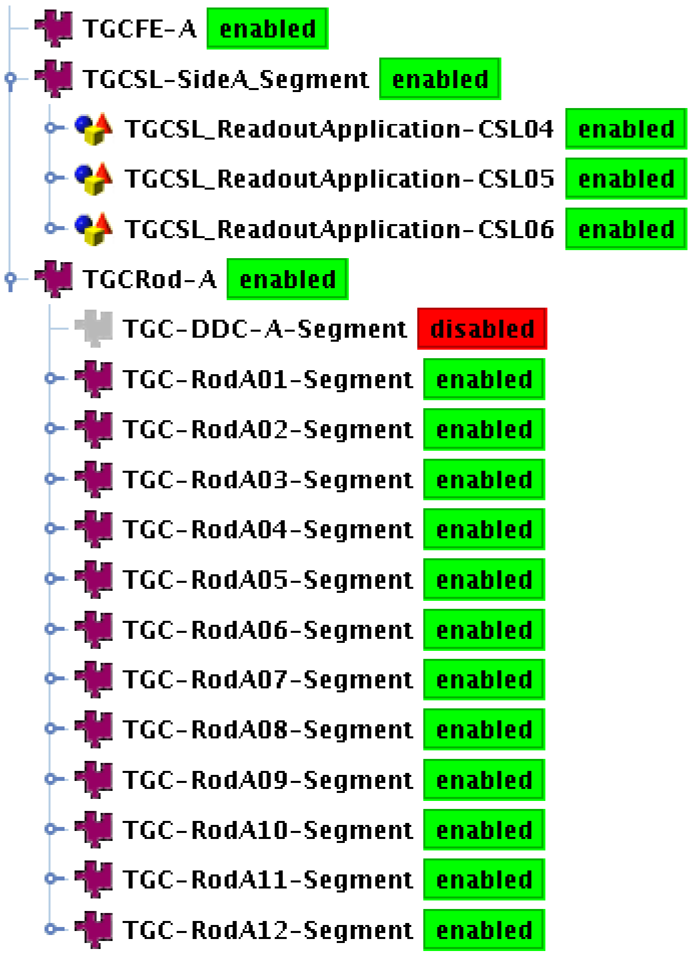
- As the system is large and complex, it is of paramount importance to have mechanism for early detection of problem, quick diagnosis and fixing the problem
  - Network based message logging system – important information for system operation passed to central system operator, categorised in warning, error, fatal etc.
  - File based message logging system [per process] – detailed running record of each component for investigation of problems

- Xml based database for system configuration
- Monitoring
- State machine
- VME access driver and library for supported hardware/OS

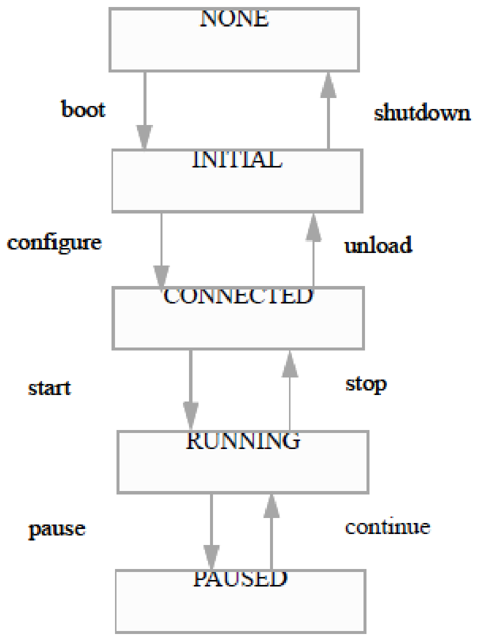
ATLAS Trigger & DAQ control

- Detector specific software is integrated and controlled by GUI application
- Hierarchy of run controller handles all the components in a run
  - Each detector component also has multi layer structure, configurable with xml based database

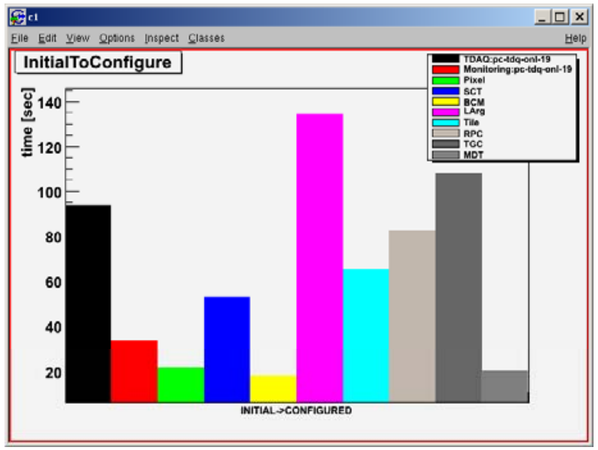


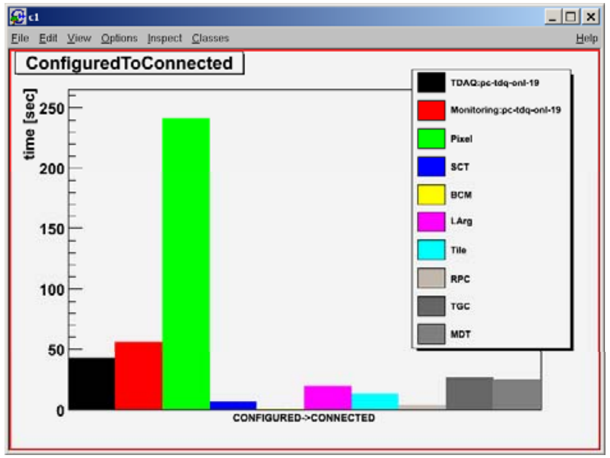


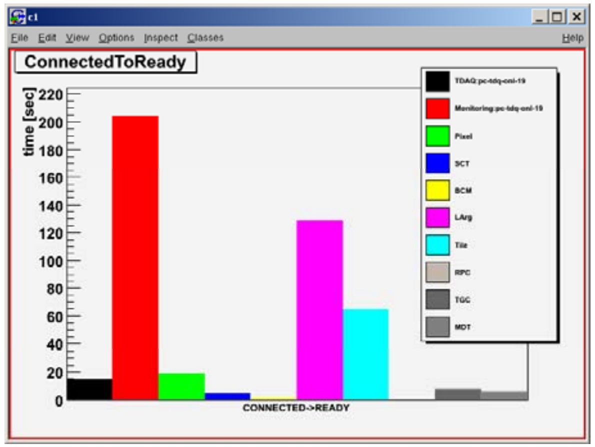
ATLAS TDAQ state machine



- State machine to synchronise activity of each detector
- ATLAS State machine as seen by an operator
  - Boot – initialise run controller tree
  - Configure – start application for control/read-out, set-up registers, download FPGA firmware etc.
  - Start – release BUSY and be ready for trigger
- Typical transition time for recent runs
  - ~ 10 min.







ATLAS TDAQ recent status

- Some numbers from a recent run;
  - number of computer nodes used for Trigger & DAQ (TDAQ) system: ~ 1600
  - Xml configuration database size: ~ 40 MB
  - Average event size: ~ 3 MB
  - Throughput to disk: ~ 350 MB/s
  - Trigger rate: O(100) with cosmic trigger, 20kHz random L1 trigger

Partition ATLAS

Run State	RUNNING	**Error State**	No error
Run Type	physics	**Run Tag**	data08_cos
Run Number	89335	**Run Time**	00:05:40
LuminosityValue	2	**Changes every**	1000 SECONDS

L1 Accepts	16090	**Average Event Size [MB]**	3.16738
L2 Accepts	16090	**Recorded Events**	14851
EF Accepts	14834	**Throughput to Disk [MB/s]**	356.325

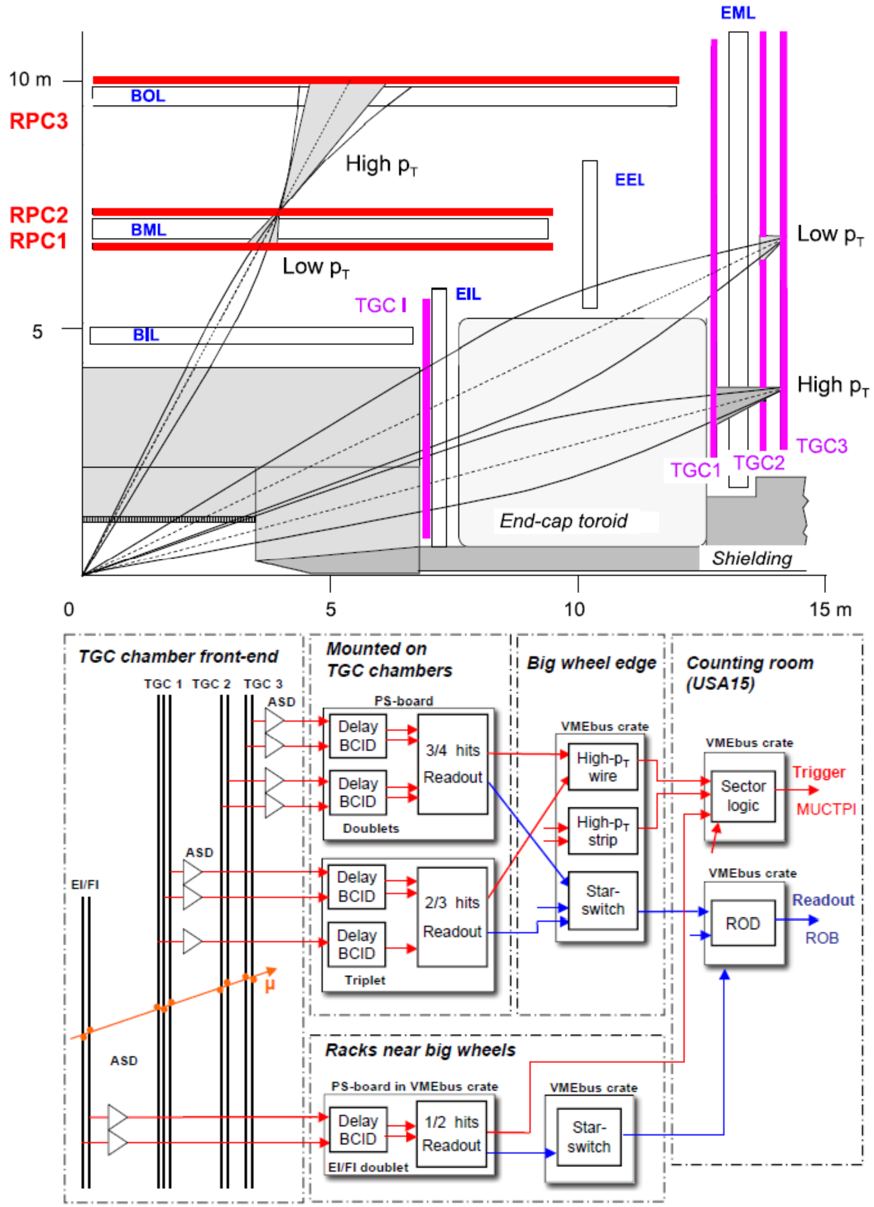
Busy Monitoring

CTPMI		**CTPCORE**		**CTPOUT 12**		**CTPOUT 13**		**CTPOUT 14**		**CTPOUT 15**	
VME	0%	**Backplane**	0.624699%	**TTC2LAN**	0%	**BCM**	0%	**LHCf**	OUT	**CSC**	OUT
ECR	0%	**Result**	0.624699%	**Pixel**	0.419204%	**LAr H/F-C**	OUT	**MDT B**	0%	**TGC-C**	0%
Veto0	0.202101%			**SCT**	0%	**LAr H/F-A**	0%	**MDT EC**	0%	**TGC-A**	0%
Veto1	0%			**TRT**	0%	**LAr EMEC**	0%	**Tile EB**	0%	**RPC**	0%
Backplane	0.621744%			**L1Calo**	0%	**LAr EMB**	0%	**Tile LB**	0%	**MUCTPI**	0%

- ATLAS TDAQ system ready for data taking

ATLAS TGC

- Part of ATLAS level1 muon trigger system
- Provides muon trigger tagged with p_T information
  - p_T estimated with curvature in magnetic field
- Custom built trigger and read-out electronics, ~ 300k channels
- One read-out and trigger unit is a sector; 1/12 in phi
  - timing/coincidence/read-out ASICs ~ 5000 registers via JTAG (max 160 bits)
  - 6 FPGAs for trigger
  - one read-out driver
- ROD: Israel
- Others [Front-End]: Japan



ATLAS TGC Front-End system – I

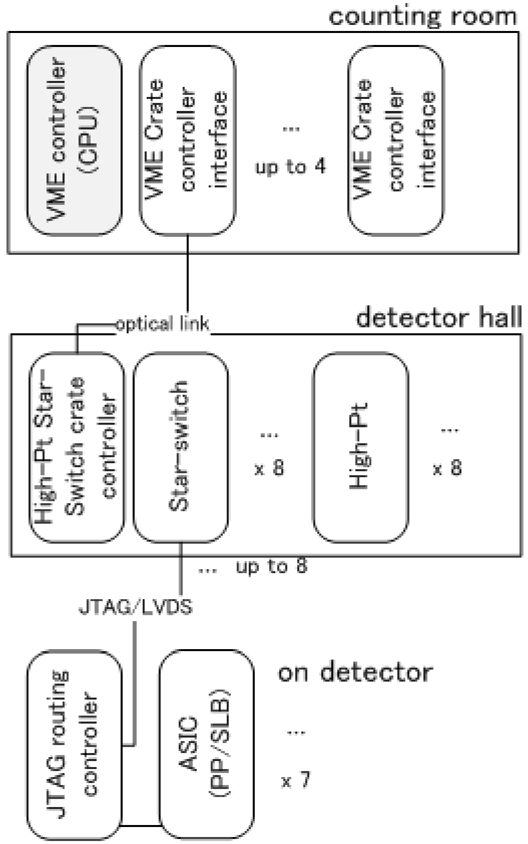
- As of last October
  - Hardware had been designed, implemented, tested and being installed
  - On the other hand, software had been just implemented
    - Took ~ 10 min to set-up one sector
    - Frequent failure on register setting with JTAG ~ 1 error per sector
      - → unreliable, inefficient

- Urgent need to implement software that works correctly, reliably and efficiently before start of data taking

- Strategy
  - Use ATLAS TDAQ framework as much as possible
    - cost of software maintenance is much higher than developing; never re-invent the wheel
  - Make software system as simple as possible for maintenance
  - Implement error check at every step to prevent operation in faulty state
  - Produce a lot of useful logging messages for easy debugging/diagnosis

ATLAS TGC Front-End system – II

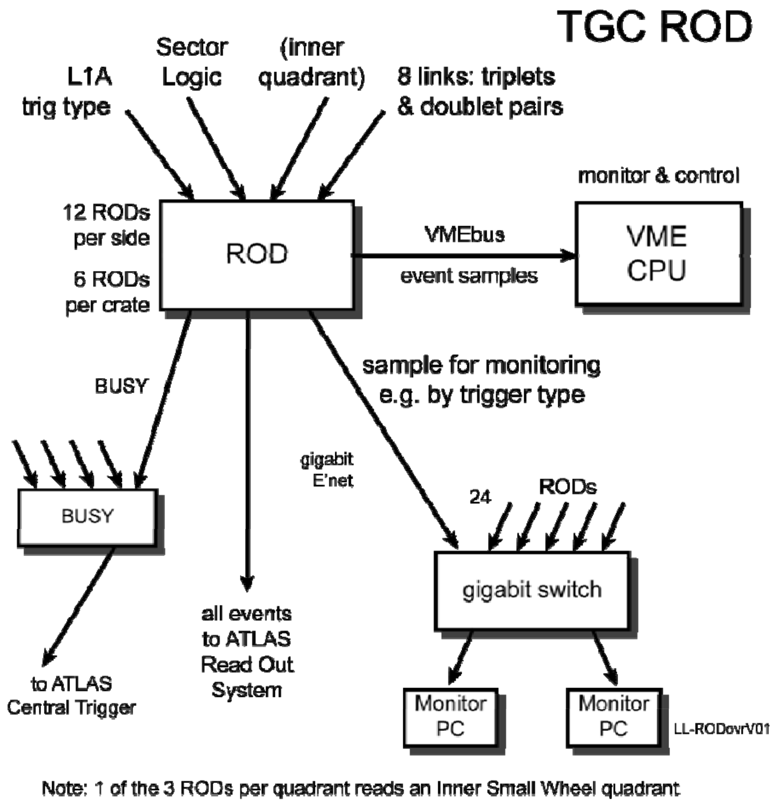
- Reliable configuration of timing/coincidence/read-out ASIC with JTAG critical ~ 5000 registers per sector
- JTAG access is not so simple
  - Control software needs to handle VME → CCI/HSC boards (VME) → JTAG chain to read/write a register
- Resource handling is critical
  - use semaphore for exclusive access
- Data integrity check is critical
  - use the simplest way; write to register then read it back, repeat until we get consistent read back value



- After renovation of the software
  - Reliable register setting ~ negligible error rate
  - Takes ~ 2 min to set-up the whole system

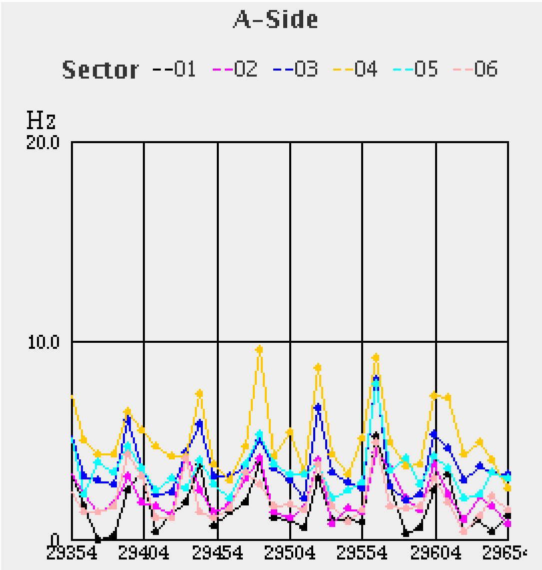
ATLAS TGC ROD

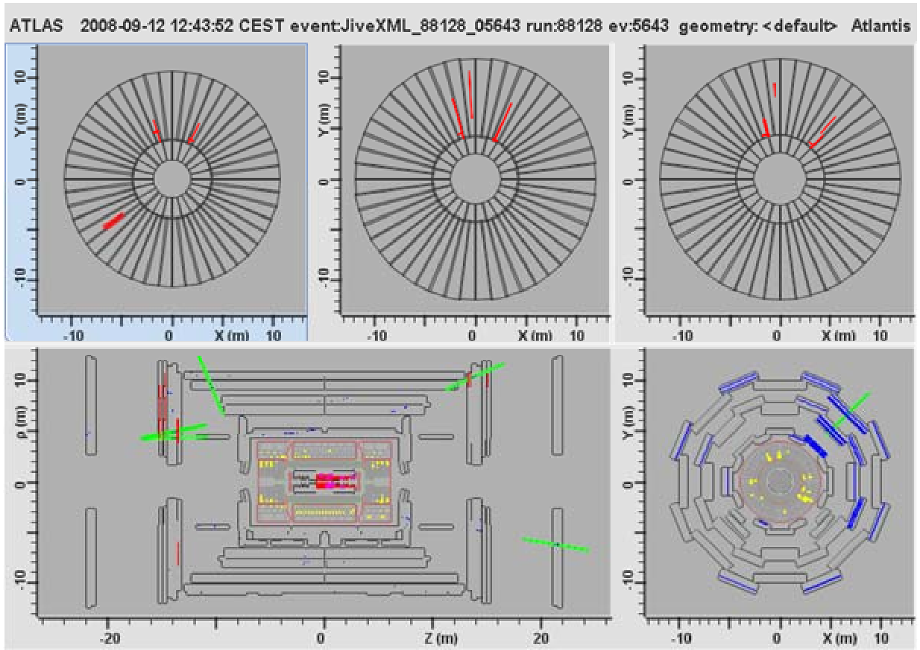
- TGC uses 24 read-out drivers (ROD); one ROD per sector [13k channels]
- Input
  - 12 optical fibres for data input
  - one optical fibre for trigger
- Output
  - one optical fibre for ATLAS read-out system
- ROD functionality
  - merges and checks input data
  - decodes and formats the data
  - verifies data integrity
  - sends the data to read-out system
  - generate BUSY if necessary
  - samples data for online monitoring and recording



ATLAS TGC recent status

- TGC Trigger and DAQ system working
  - Trigger
    - providing stable trigger with cosmic ~50 Hz
    - observed changes of trigger rate coincide with LHC beam injection
  - Read-out
    - current limit on read-out rate is ~25 kHz
    - observed halo-muon event successfully





Summary

- ATLAS uses three level trigger system
  - 1st custom built hardware based trigger: 40 MHz → 75 (100) kHz
  - 2nd software trigger with region-of-interest information: → 3.5 kHz
  - 3rd software trigger with event reconstruction → 200 Hz

- ATLAS DAQ system
  - detector specific part – custom built hardware
  - other part uses commodity – PC farm on gigabit ethernet
  - stores events at a rate of ~300MB/s

- ATLAS Trigger and DAQ system is ready for data taking
- ATLAS TGC Trigger and DAQ system working as well;
  - reliable configuration of system ~ 130k registers in 2 minutes
  - reliable read-out system ~ 300k channels at ~ 25 kHz

- Awaits for collision data
  - To move on to calibration and physics analysis