The ATLAS High Level Trigger Configuration and Steering

Experience with the First 7 TeV Collisions Data

18^{th} International Conference on Computing in High Energy and Nuclear Physics

Taipei, Taiwan
18^{th} - 22^{nd} October 2010

Jörg Stelzer

DESY Hamburg, Germany

for the ATLAS Collaboration

Content of the Presentation

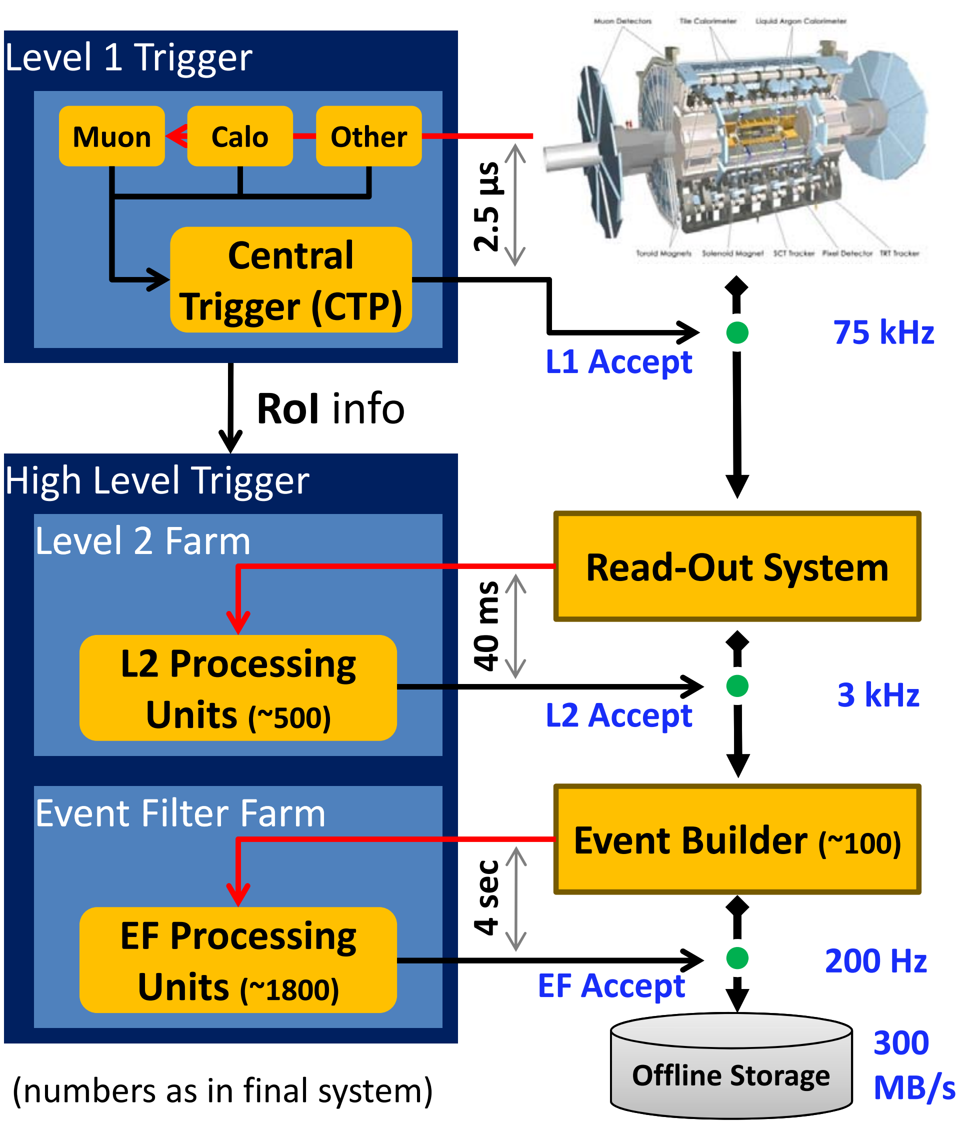
- **Requirements on the High Level Trigger (HLT) steering and configuration system**
  - Online operations, physics analysis
- **Introduction of the trigger system**
  - Menu, configuration, and steering
- **Online update of the trigger during data taking**
  - Changing the trigger menu
  - Updating the beam spot information
- **Trigger monitoring**
- **Trigger information for physics analysis**
  - Information flow and distribution
  - Trigger data access

HLT Requirements

- **Reliable and fast execution**
  - Robust algorithms, failure handling mechanism
  - Early rejection to minimize data access and processing time
- **Flexible online configuration**
  - Quick deployment of different trigger configurations to adjust to LHC beam conditions and ATLAS data taking requirements
- **Independent execution of all trigger signatures**
- **Access to trigger decision, trigger objects, trigger configuration**
- **Monitoring of trigger performance**
  - Execution timing, trigger object properties

Applies for Level 1 Trigger as well !

The Trigger System



(numbers as in final system)

Level 1

- Analyses data from CALO, MUON and other detectors (e.g. minbias)
- CTP combines triggers
- Identifies Region of Interest (RoI) used to seed Level 2

Level 2 (L2)

- Mainly partial reconstruction inside RoI
- Event fragments on-demand from Read-Out System
- Fast algorithms, optimized for early rejection

Event Filter (EF)

- Full event assembled by Event Builder
- Offline algorithms in custom wrappers
- Reconstruction mainly inside RoI, some algorithm use full event

Trigger Menu, Configuration

Signature: reconstruction and selection algorithms to identify particles (leptons, jets) and event properties (total/missing transverse energy), executed in steps

- Names reflect type, energy thresholds, multiplicities, …
- A signature spans all three levels, each seeding the next
- Multiplicities are checked after each step
- Special: beam spot, monitoring, calibration

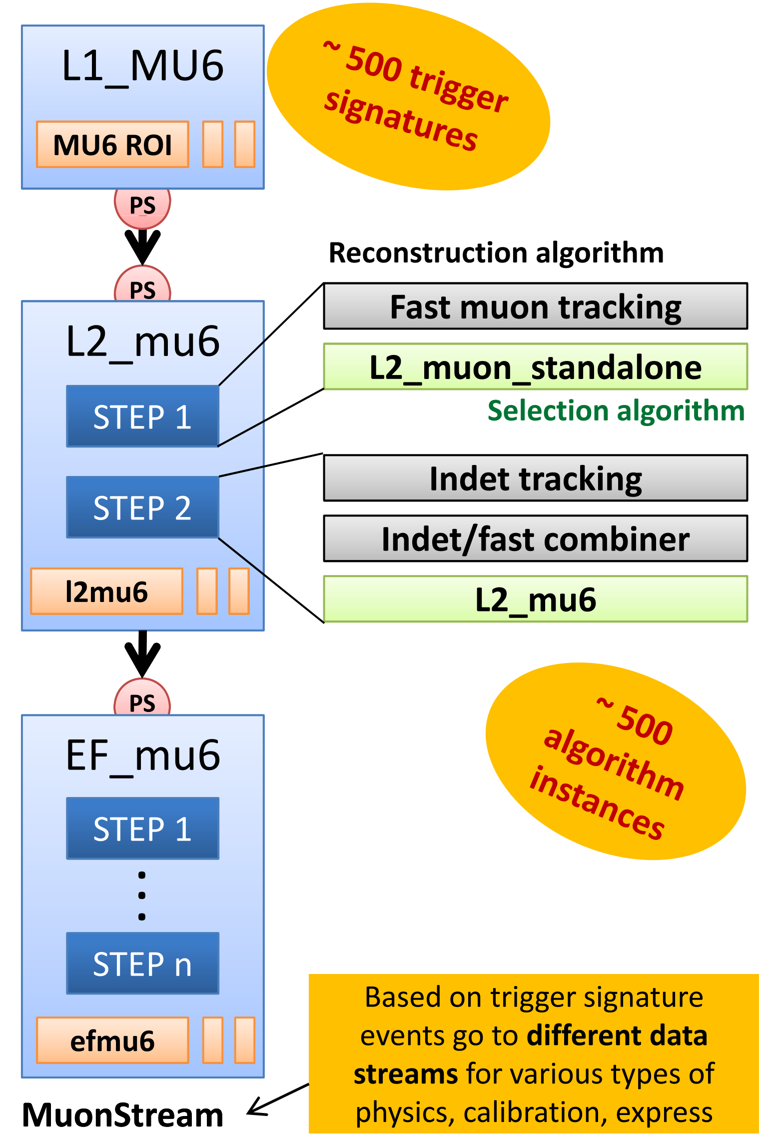
Prescaling: start execution for only a fraction of candidate events

- Control of **trigger rate** (mostly low energy thresholds)
- Prescale (PS) for each signature at each level
- **Enable/disable** signatures during the run

Menu: a list of **signatures** and their **prescales**

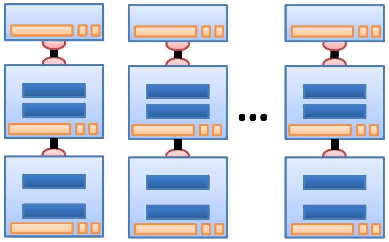
Menu stored in **Configuration Database**

- Identified by three integer **keys** (Menu, L1-, HLT-prescale-set). Immutable, provides history of the trigger configuration.



The Steering – the Heart of the HLT

Framework to execute all trigger signatures independent of each other and form the event accept decision.



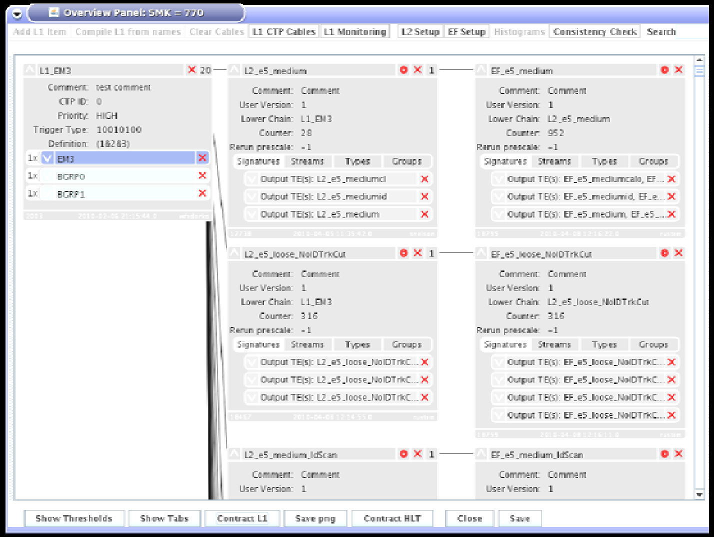
- For each signature:
  - Checks trigger decision of previous level (must be positive)
  - Applies prescale decision *before* algorithm execution
    - **If prescale value is less than 0, do not execute this signature**
  - Executes trigger algorithms to build and select trigger objects
  - Checks multiplicity of selected trigger objects → marks trigger signature as passed
- At least one signature passed ? → event accepted
  - Based on passed trigger signatures assigns event to various *data streams*

- Other Features of the Steering
  - **Caching** of algorithm **execution** (if same input and algorithm configuration)
  - Stores **mapping** of trigger signatures to trigger objects → physics analysis
  - Execute signatures after event is accepted → increase statistics in correlation matrix
  - **Monitors** execution time, provides online trigger rates

Trigger Configuration – Database Work

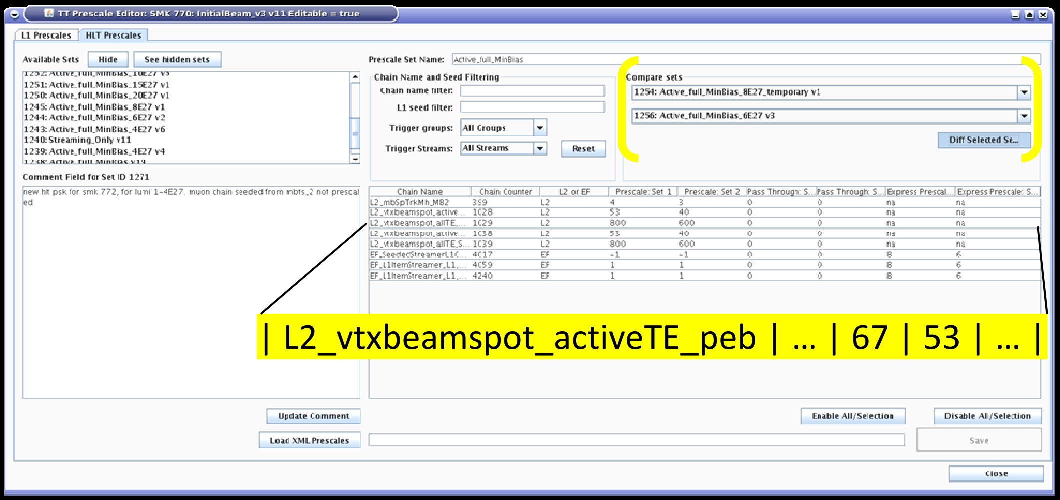
- Entire L1 and HLT configuration in the trigger configuration database (TriggerDB)
- Trigger menu and operation experts need a flexible tool to
  - Browse/search the TriggerDB for information
  - Import new configurations into the TriggerDB
  - Modify existing configurations
  - Copy configurations between TriggerDB’s
- GUI: TriggerTool (TT)
  - Java application
  - Command-line version for repetitive tasks
  - Also used for Test- and Monte Carlo- TriggerDB

Multiple prescale sets prepared for use during a run to optimally use available bandwidth

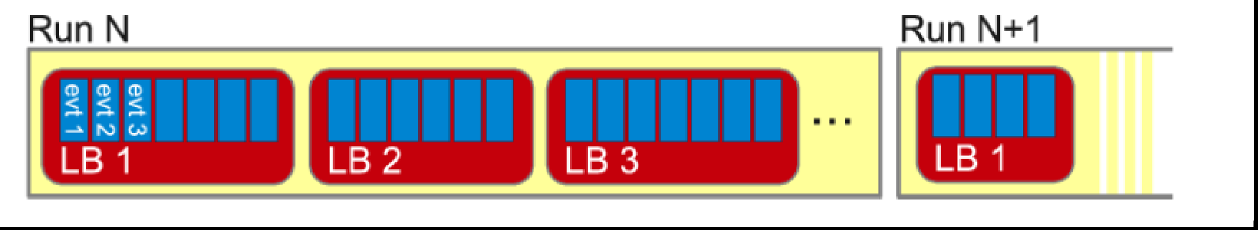


E.g. trigger menu editor in the TT

E.g. viewing differences of prescale sets with the TT



ATLAS Run Structure



- Run
  - Period of continuous data taking
  - Typically lasts hours, usually corresponds to an LHC fill
- Luminosity Block (LB)
  - Time interval (~2 min) within a run
  - Luminosity and conditions are considered to be approximately constant
  - Smallest unit of data considered by data quality,
    - **Data quality of a LB is criteria for use in physics**
- Prescales and beam spot can be updated between LB

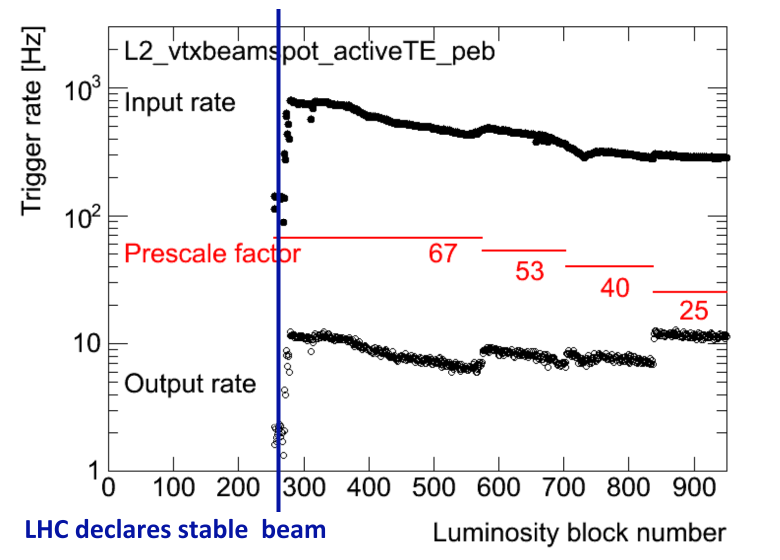
Online Prescale Update

Motivation for changing prescales

- Enable/disable signatures
  - Switch on all physics triggers when LHC declares stable beam and ATLAS detector is ready (HV up)
- Adjust prescales size
  - Optimize bandwidth use as luminosity falls during fill

Requirements

- No reduction in data taking efficiency
- Apply consistently across all HLT nodes
- Reproducible offline



Example trigger with adjusted output rate. The prescales values/sets are calculated in advance for all lumi ranges.

Solution

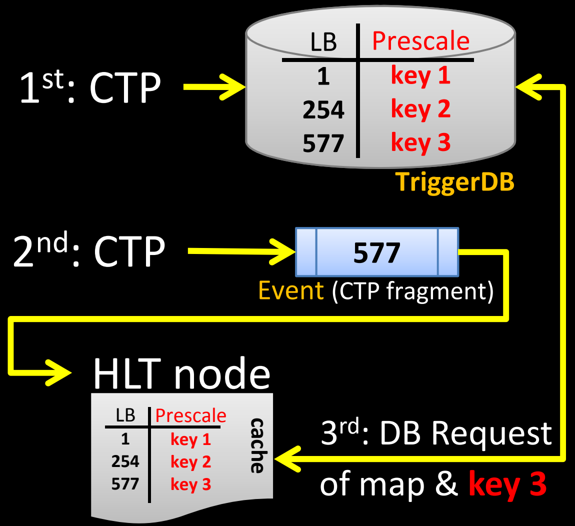
1st CTP stores *map of LB to HLT prescale key* in TriggerDB

2nd CTP flags change request as part of event data

- Ensures the request reaches all HLT nodes

3rd HLT nodes extend internal cache: *map*, *new prescale set*

- Apply prescales according to the LB of the event
- Archive prescales to conditions database for physics analysis



Process initiated by shift leader, trigger and run control shifter !

Online Beam Spot Update

Motivation

- A number of signatures are sensitive to the beam spot position (B-triggers)
- Fast determination of beam spot and feed back to trigger algorithms

Requirements

- As for prescale updates

Solution

(1) Event vertex fitting at Level 2

- Histogram vertex information, collect and sum over time

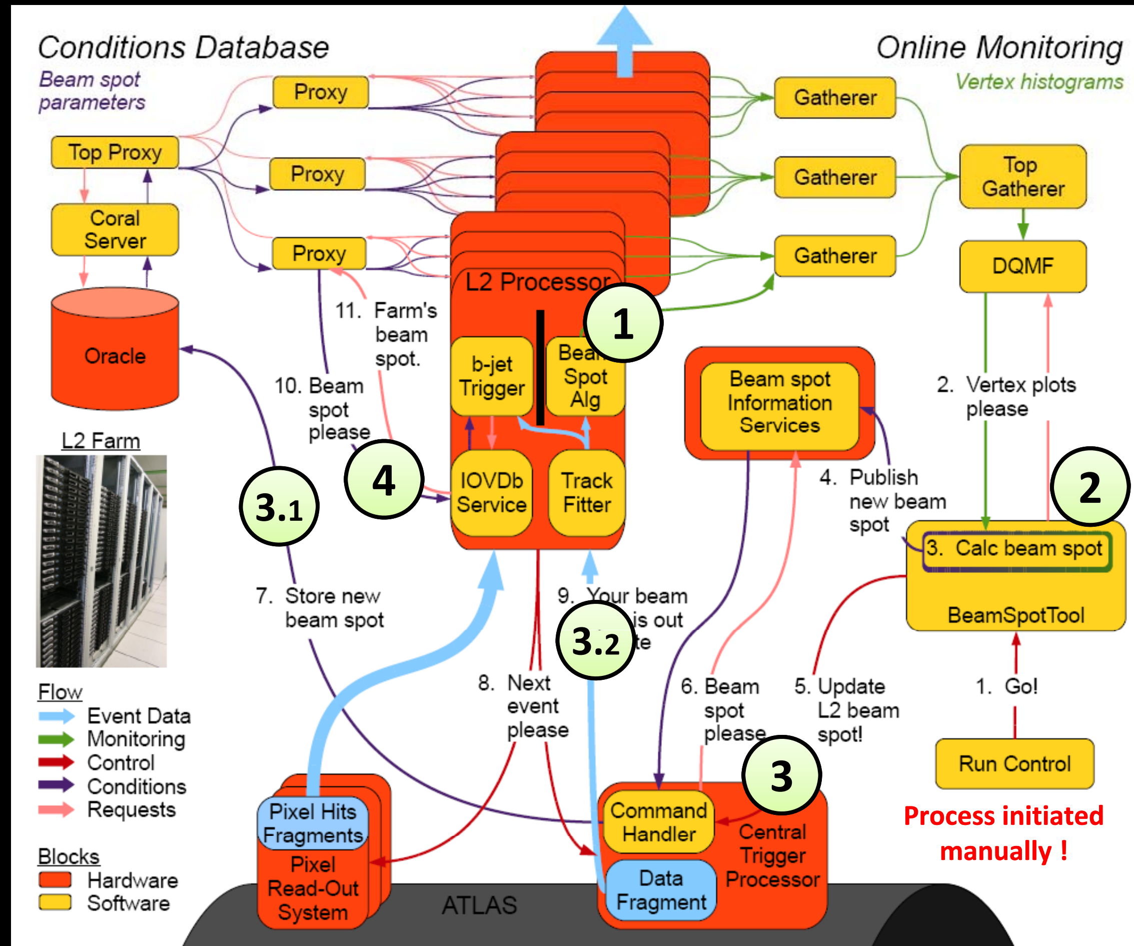
(2) Fit position, tilt angles of interaction region

- Check if update needed

(3) Send parameters to CTP

- (3.1) Update conditions database
- (3.2) Use event (CTP fragment) to notify HLT nodes to re-read conditions DB

(4) Update beam spot parameters



More details on the Poster about the ALTLAS beam spot online update mechanism on Wednesday.

Monitoring HLT Resource Usage

Motivation

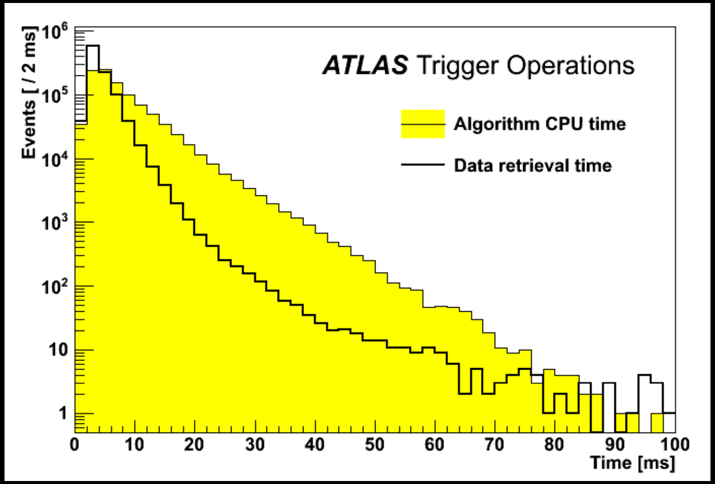
- Understand the online CPU usage and data access, identify cause of any problems
- Provide detailed input to predict resource usage at higher luminosities

Requirements

- Monitor online CPU usage and data access rates
- Rejected events must be included
- Handle large number of events, algorithms, and HLT applications.
- Minimal impact on DAQ system

Solution

- For every event record trigger decisions
- Sample timing data from the instrumented HLT Steering and data access interface
  - Buffer timing data from multiple events
- Data written out at low rate like calibration events into event data stream

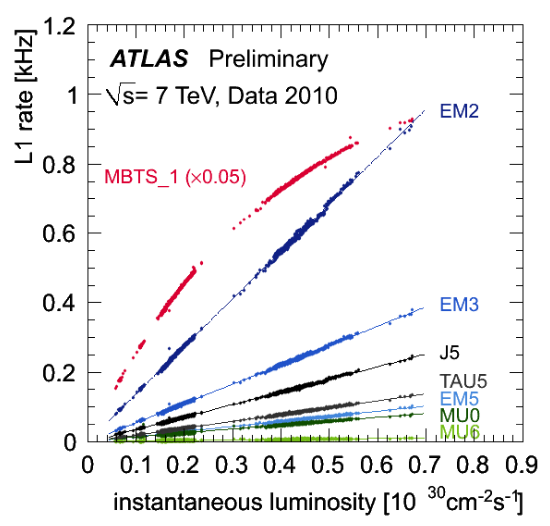


Example of execution time monitoring for a typical L2 reconstruction algorithm
Algorithm time (CPU) and data access.

Monitoring Data for Resource Use Prediction

Rates of physics trigger signatures scale with luminosity and are easy to predict

- Exception: some signatures are sensitive to pile-up (#interactions/event), e.g. total energy triggers, others saturate



L1 rates scaling with luminosity

Prediction of:

a) Events-to-tape rate – include correlation between signatures and stream overlap

b) Trigger execution time – caching of execution and raw data from detector read-out

c) Data access rate and bandwidth usage – caching of raw data from detector read-out

Steering and data access interface are instrumented to record execution and data access times and cache requests.

- Offline analysis of enhanced bias data (realistic L1 trigger distribution) predicts events-to-tape rate for higher luminosity and new menus. Data access rates can be estimated.
- Invaluable for menu design as it guaranties stable operations

Trigger Information for Physics Analysis

Data flow:

- Trigger decision, trigger object: per event data
- Trigger configuration: meta data in conditions database and offline data files

Trigger DB unique source of trigger configuration

Data taking

Reconstruction

Physics analysis using the trigger

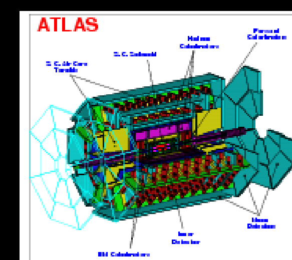
TriggerDB
All configuration data

WEB

Stores Trigger Menu

Configures

Conditions Database



Trigger menu

Encoded trigger result from all 3 levels

Data File

- Trigger Decision
- Trigger Objects
- Trigger Configuration

Single tool to access

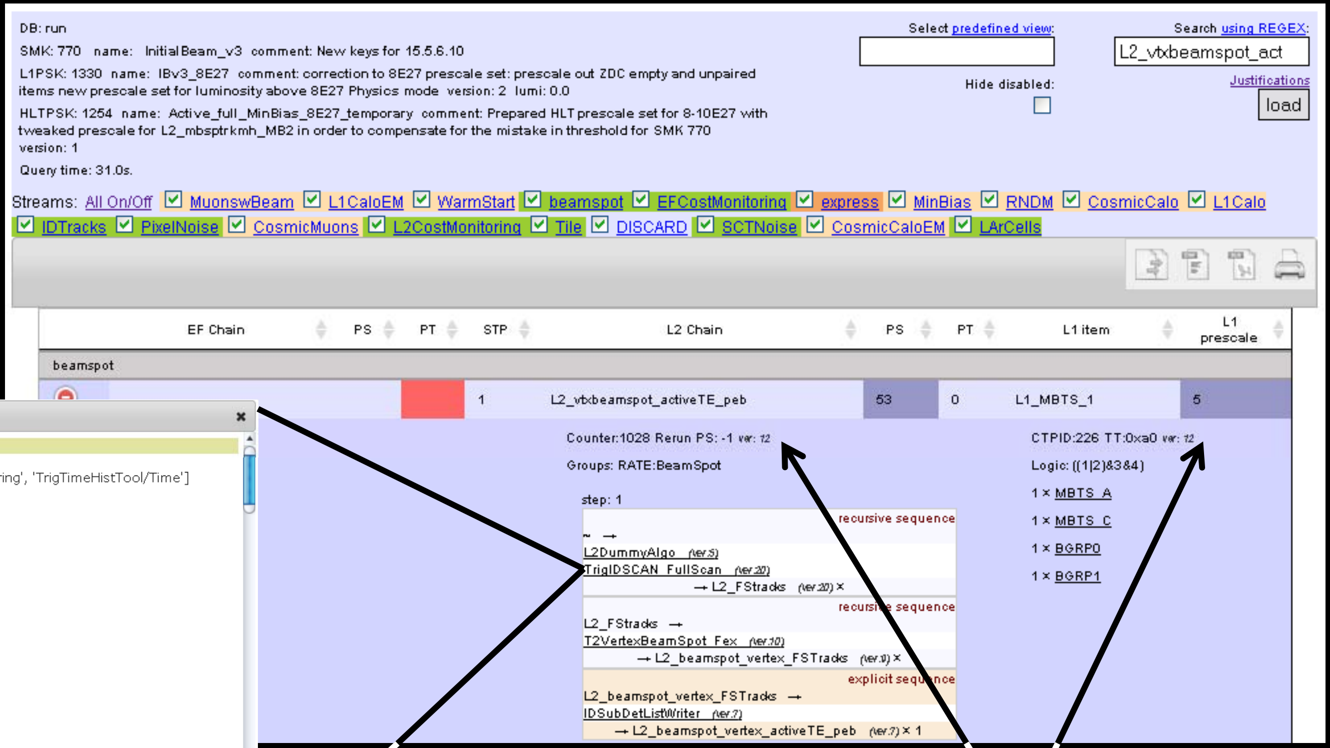
ESD | AOD | DPD | TAG

ATLAS data formats

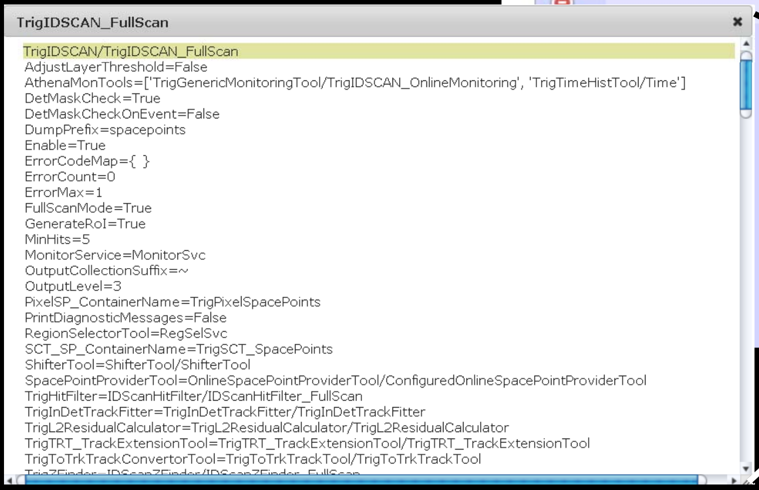
Web-Based Trigger Configuration Browser

Browse trigger configuration for all ATLAS data:

- Data streams, trigger signatures, algorithms, selection cuts, versions …



Properties of each reconstruction and selection algorithm



Triggers version numbers automatically assigned when configuration changes

Under development:

Trigger evolution browser – important to understand trigger performance over time

Conclusion

- **Trigger configuration tools to browse and modify configurations**
- **HLT prescales and online beam-spot can be updated during a run**
  - Adjust the trigger to changing beam and data taking conditions
- **HLT resource monitoring gives a detailed picture of CPU usage and data access size/rate**
  - Quickly spot unusual behavior and pinpoint problems
- **All information to analyze the effect of the trigger in physics analysis is recorded and available for remote data analysis**

ATLAS has a reliable and flexible trigger that has proven successful in taking 7 TeV LHC data. Support analysis of trigger data in every level of detail.