The Software Architecture of the LHCb High Level Trigger

M. Witek  IFJ PAN Cracow, Poland
On behalf of the LHCb Collaboration

CHEP 2012 Conference, New York
Outline

• The LHCb experiment
• Aim of the trigger
• Structure of the High Level Trigger (HLT)
• HLT evolution in 2010-2011
• Summary
The LHCb experiment

- The main goal is to study matter-antimatter asymmetry (CP symmetry breaking) and other rare phenomena in B and D meson decays with unprecedented precision*
  - Both, CP breaking and rare decays occurs in a tiny fraction of all proton-proton interactions

* The whole LHCb physics programme is wider.
Aim of LHCb trigger

- Exploit finite \textit{lifetime} and large \textit{mass} of beauty hadrons to distinguish heavy flavour from background in inelastic pp scattering
  - Remove uninteresting events as soon as possible
  - Keep high efficiency for signal events

Focus on B mesons which describe the trigger concept well

IP - Impact Parameter = distance of closest approach of the B-decay product to PV (\textit{lifetime})

PT – transverse momentum = \( \sqrt{p_x^2 + p_y^2} \) (\textit{mass}). Beam line along Z axis
Trigger constraints

• Running conditions are not stable
  – Beam parameters are likely to vary
  – Physics goals are changing with time
  – Average event execution time is limited
  – Detector simulation may not be perfect (electronic noise etc.)
    • Trigger had to be ready before first collisions in 2009

• Design assumptions
  – Flexibility – reconfiguration should be straightforward
  – Reuse the off-line software (minimize trigger bias)
  – Ability to adapt to fast increase of luminosity at the LHC startup

• Correct balance between:
  – optimization, flexibility, reconstruction quality and speed
Trigger constraints

- Trigger was designed for the number of visible p-p interactions $\mu=0.4$
  - Maximizes single p-p interaction per bunch crossing for the nominal LHC beam parameters
- It turned out that with reduced # of bunches in LHC more optimal for LHCb was to go for higher $\mu$ of $\sim1.6$
  - Value foreseen for LHCb upgrade
  - More complicated events – higher detector occupancy → more CPU needed for reconstruction

At LHCb interaction point, luminosity exceeded design value with 30% of nominal number of bunches

Peak luminosity increased within one month by a factor of 100. Valuable measurements at low luminosity with tuned less selective trigger
Structure of LHCb trigger

„On-line charm physics”
Signal/background ratio used to inspect data quality
Reconstruction tasks

- Main reconstruction tasks:
  - VELO tracking – reconstruction of VELO tracks
  - Primary Vertex reconstruction
  - Forward tracking – extending VELO track to T1-T2-T3 (CPU intensive)

- Same algorithms used for on-line and off-line whenever possible
  - Difference in tuning and quality of input

Forward tracking. Region of Interest narrowed by assuming p>6 GeV, PT>0.5 GeV
High Level Trigger

- The selections which are run as C++ application within the standard framework of LHCb, the Gaudi framework
  - http://proj-gaudi.eb.cern.ch/proj-gaudi.
- 20 k identical processes running on Event Filter Farm
- The selections are organized in so called trigger lines
- ~40 HLT1 lines and ~130 HLT2 lines
- The lines are independent and run irrespectively of the decision of any other line
  - Time consuming reconstruction algorithms are executed only once per event
- The trigger lines are coded by a number of physicists
  - A uniform framework required to control versions, reduce the code crashes and impose performance constraints (timing)
- Concentrate on evolution of HLT to adapt $B \to \text{hadrons}$ selections to unexpected running conditions for
  - HLT1 - 1 track all
  - HLT2 - Topological
High Level Trigger

- The HLT framework
  - Same as for off-line -> reuse of well tested code and configuration tools
  - Based on hybrid C++/Python environment* built from standard components
    - The C++/Python binding and intercommunications have been performed using Reflex dictionaries
  - The action of components is defined at an initialization phase using a palette of C++/Python functors (LoKi/Bender framework)
    - Very powerful mechanism to convert complicated expressions coded as strings in PYTHON into the C++ code
    - Parser of expressions, units, operators, compositions
  - It allows users to write the HLT selections in a uniform environment
    - combine and filter of particles in compact and coherent way

* Python-based Physics Analysis Environment for LHCb, Computing in High Energy Physics and Nuclear Physics 2004, Interlaken, Switzerland, 27 Sep - 1 Oct 2004
**HLT1**

- A HLT1 line is a chain of algorithms to process or filter trigger objects.
  - The input of an algorithm is an output of previous algorithm.
  - Each line returns its decision.
- The workflow of the HLT1 line is built using:
  - Standard off-line algorithms (configured for HLT)
  - Dedicated algorithms to quickly accomplish special tasks
- The configuration is handled using *streamer framework* (LoKi/Bender)
  - Organize C++ code workflow in readable and consistent way,
  - Safe, robust and easy to modify structure,

VeloCandidates

```c
>> MatchVeloL0Calo
>> ( ( TrIDC('isVelo') > %(Velo_NHits)s ) & ( TrNVELOMISS < %(Velo_Qcut)s ) )
>> TightForward
>> (TrTNORMIDC > %(TrNTHits)s )
>> ( ( TrPT > %(PT)s * MeV ) & ( TrP  > %(P)s  * MeV ) )
>> FitTrack
>> ( TrCHI2PDOF < %(TrChi2)s )
>> SINK( 'Hlt1%(name)sDecision' )
>> ~TC_EMPTY
```

HLT1 line within streamer framework (simplified example with parameters)
HLT1 - confirmation concept in 2010

- Confirmation of the L0 object
  - Add information of T stations and VELO
  - Search for tracks only in narrow window

- HLT1 based on L0 confirmation commissioned for 2010
  - Small number of L0 objects → reduced tracking effort
- but:
  - It was designed for nominal beam conditions
  - Improvements needed in particular for 2011 running conditions
    - Reduction of ghost tracks at higher detector occupancy
HLT1 – One Track trigger in 2011

• New inclusive approach based on observation that:
  – In every signal decay, there is always at least one good quality track with sufficiently large IP and PT

• The method:
  – VELO reconstruction (no momentum information)
  – Select tracks with high IP
  – Forward reconstruction with minimum momentum and PT enforced ($\Delta p/p \sim 1\%$)
  – Track fit – fast Kalman filter with simplified detector geometry
  – For good quality tracks ($\chi^2/\text{ndf}$) check incompatibility of coming from primary vertex
    • Quality of track parameters compatible with that of off-line at first step of HLT!

• Performance
  – HLT1 efficiency 70-90% wrt to L0
  – Extremely robust inclusive method
  – Timing ~12 ms/event
• Upper level configuration by means of dedicated PYTHON classes (Hlt2Line)
  – Automatically coupling output/input of subsequent algorithms in a chain
  – Use the same selection framework as for off-line selections

• HLT2 initially meant as a set of exclusive trigger lines
  – Evolved towards inclusive selections + some dedicated exclusive selections

• Almost off-line quality reconstruction
  – Global track reconstruction
    • Forward tracking of all VELO tracks
    • Other reconstruction on-demand
HLT2 – topological selections 2010

- Topological trigger is designed to select decays of any b-hadron decays with at least two charged secondaries
- Cut based approach was used in 2010
  - Track $\chi^2$/ndf, IP$\chi^2$, PT, $\Sigma$PT, flight distance $\chi^2$, $\Sigma$ IP$\chi^2$, DOCA, corrected invariant mass
- 2-, 3- and 4-body versions
- Main trigger for hadronic B decays
- 80-90% efficiency

\[ m_{\text{cor}} = \sqrt{m^2 + (p_T^{\text{miss}})^2 + |p_T^{\text{miss}}|} \]

$P_T^{\text{miss}}$ is missing momentum transverse to the flight direction from primary to secondary vertex.
In 2011 beam conditions required further reduction of background by a factor of 3, possibly keeping the same signal efficiency.

- MVA used instead of cut based selection – same input variables.

A Boosted Decision Tree:
- Protect the BDT from using small regions of phase space
- Ensure common properties of B-decays are used
- Minimize on-line vs off-line differences.
- Create a sufficiently fast implementation.

HLT2 – topological selections 2011
Use a BDT with discretised variables: Bonsai Boosted Decision Tree.

- Choose discretisation using physics, resolutions and common B-decay properties.
  - Start with large number of discretization values, decrease this number without losing much in performance
- Allows the semi-infinite number of BDT if statements to be converted to a 1-D array of response values with fast look-up.

- Performance:
  - Efficiencies at least as good as in 2010
  - Background reduced by a factor 3.
  - 1 kHz combined output.

MC signal purity

| MC signal purity | L0 | 3% | HLT1 | 10% | TOP0 | 100% |
|------------------|----|----|------|-----|------|------|
| 1.5%             |    |    |      |     |      |      |
Summary

- The flexibility of the HLT software structure allowed the LHCb to adapt to unexpected running conditions in 2010 and 2011.
- Excellent performance of the trigger allowed LHCb to perform many valuable measurements.