Parallelization of Atlas Muon Trigger Algorithm

M. Bauce, A. Messina,
S. Giagu, M. Rescigno

January 10, 2014

Sapienza
Università di Roma

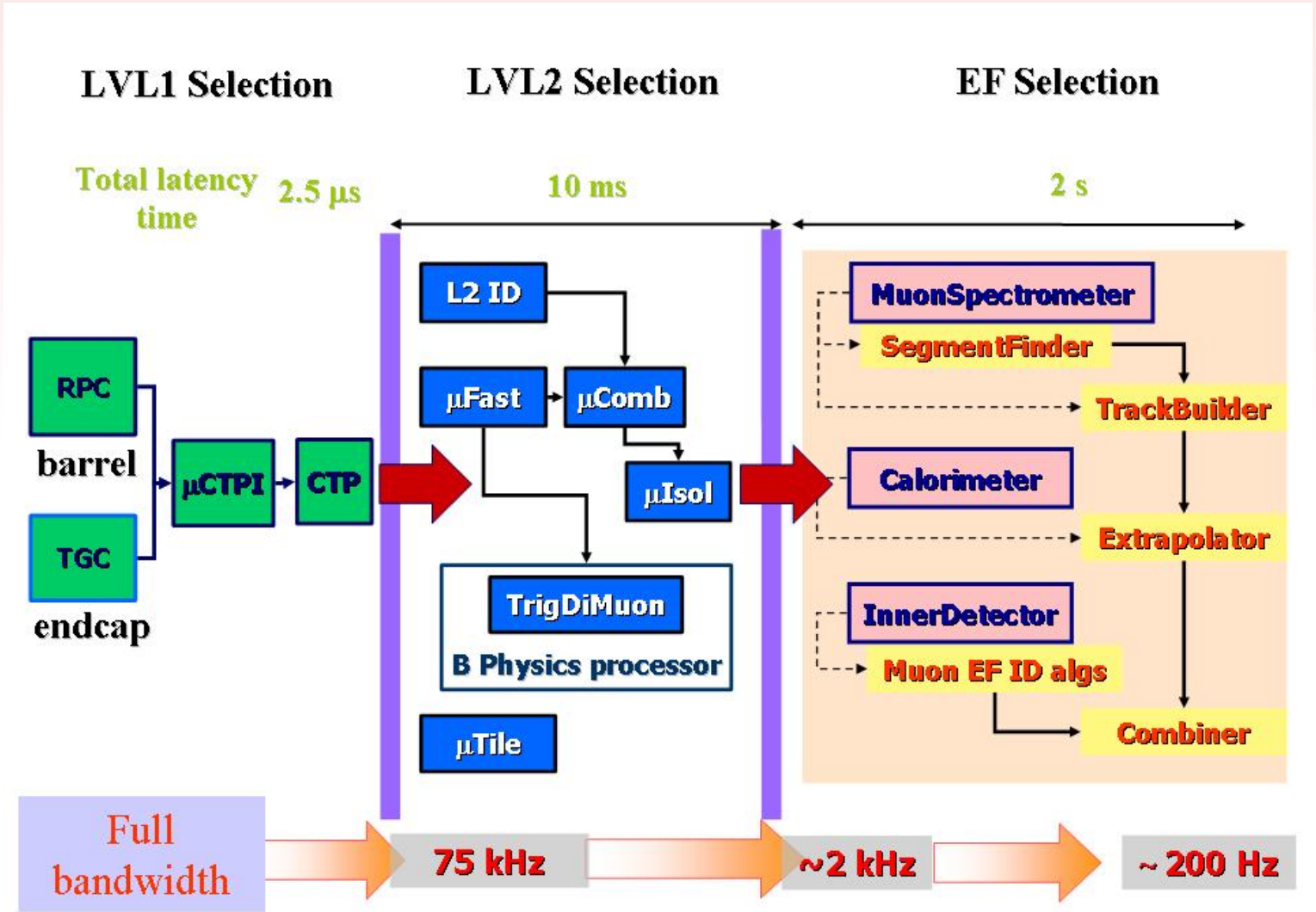
GAP RT

Atlas Trigger System

- Three stage trigger system to record data from 40 MHz *pp* interactions:
  - **L1 trigger** is completely hardware based, partial detector information used, reduce data rate to 75 kHz
  - **L2 trigger** is software based, exploits almost complete event information to reduce the rate to 3.5 kHz
  - **Event Filter** analyses the full event information giving an output rate of $\sim$200 Hz

- Software Trigger Case Study (L2/EF): **Muon Reconstruction Trigger Algorithms**
  - Parallelization of three main trigger algorithms: `muGirl`, `muComb`, `muIso`
  - Investigate the improvements from parallel computation, in particular related to the expected high-luminosity regime
  - Study the possible improvements in the parameter resolution determination, to increase efficiency/purity of the event selections

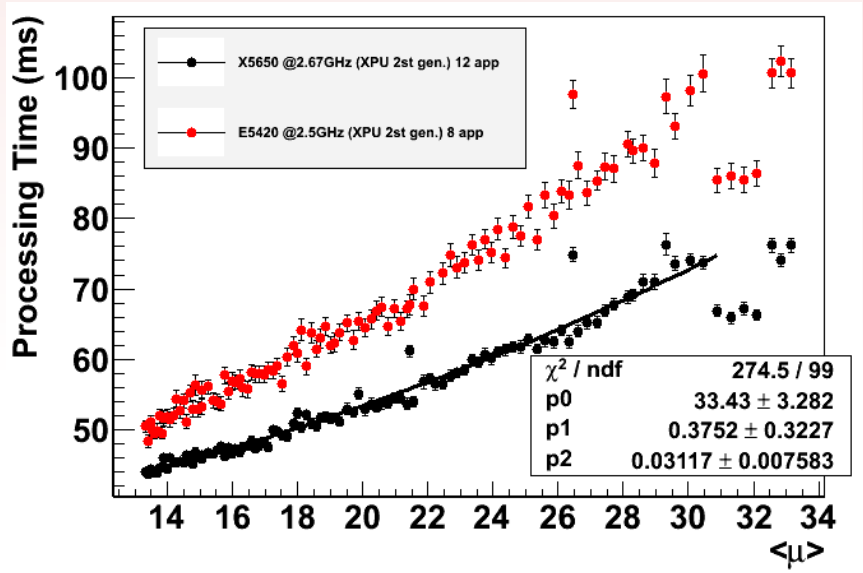
Muon Trigger Flow



► Starting from next run L2 and EF will be merged in a High Level Trigger (HLT), combining various algorithms.

Atlas Software Muon Trigger

- Level 2 Muon identification:
  - track reconstruction in the muon spectrometer: $\sim$2.5 ms
  - combination with a tracks from ID: $<$3 ms
  - track isolation in a given cone based on tracks and calorimetry: $\sim$10 ms
- For track and energy reconstruction:
  - Algorithm execution time grows linearily with the pileup
  - Purity dependent on the width of the cone



- **Find the tasks that are naturally parallelizable**, aim to a stable behaviour as pileup increase.

muComb: Track matching and FTK

- Current implementation
  - candidate reconstructed by SA-Muon as seed, extrapolation to IP using Look-Up Tables
  - combine with a track in the Inner Detector: retrieve p_T, η, ϕ
  - constraint on using loose matching criteria lead to 10% fake rates

Improvements from parallelization

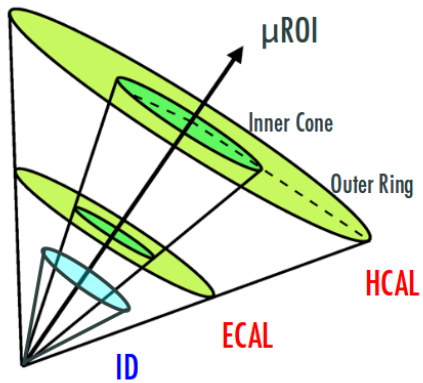
- Natural parallelization of the muon-track matching (pileup-dependent critical issue)
  - Additional tracks to consider from FTK detector
- Can implement a fast and precise Geant4 based extrapolation, that improves the LUT method

MuIso: FullScan and event-based corrections

- Current implementation
  - muon as seed (muFast, muComb)
  - Track-based isolation: exploits tracks from the silicon inner detector, discriminates based on $\sum p_T/p_T^{\mu}$ in $\Delta R < 0.2$ cone
  - Calorimeter-based isolation: evaluate the energy deposit in a cone with ΔR_{outer} <0.4 (EM+HAD), then discriminate based on $\sum E_T(outer) - \sum E_T(inner)$
    - muon energy loss subtracted through the calorimeter using a back-extrapolation of the trajectory

Improvements from parallelization

- Parallel processing of **full scan** information from FTK:
  - improves track isolation
  - pileup-dependent corrections to reconstructed parameters (already in offline)

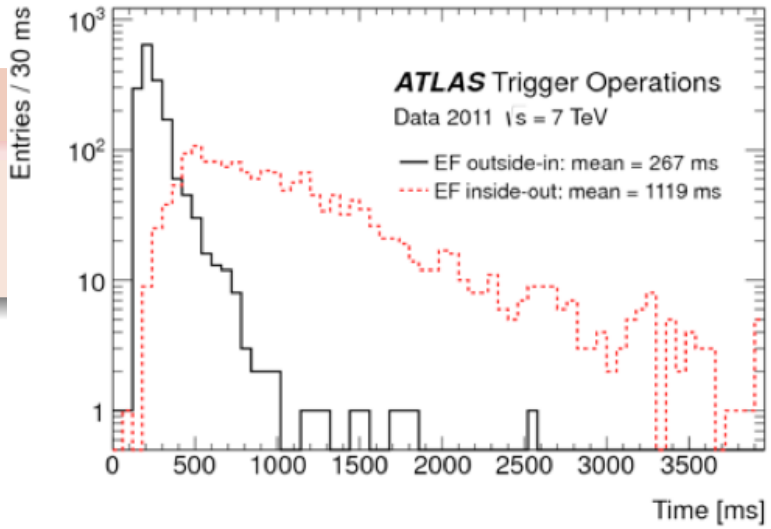


MuGirl: EF Neural Network-based algorithm

- Inside-out tracking algorithm, called `muGirl`:
  - NN-based identification of muon candidate starting from an ID track, look for a segment match in the muon spectrometer
  - Better efficiency than an outside-in algorithm in reconstruction of low-p_T muons (not crossing all of the spectrometer): interesting for B-physics analysis.

Possible improvements

- NN implementation on GPU provide huge speedup factors



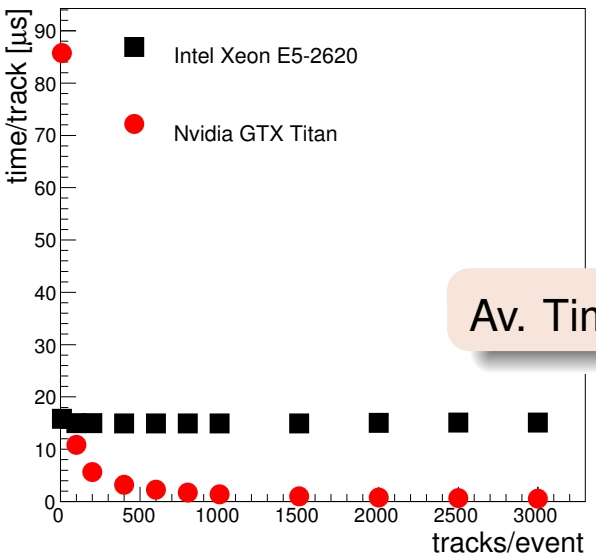
Summary and Conclusions

- Currently working for trigger algorithm parallelization
- Bottleneck might be the implementation in the Atlas framework
  - developing a standalone test environment

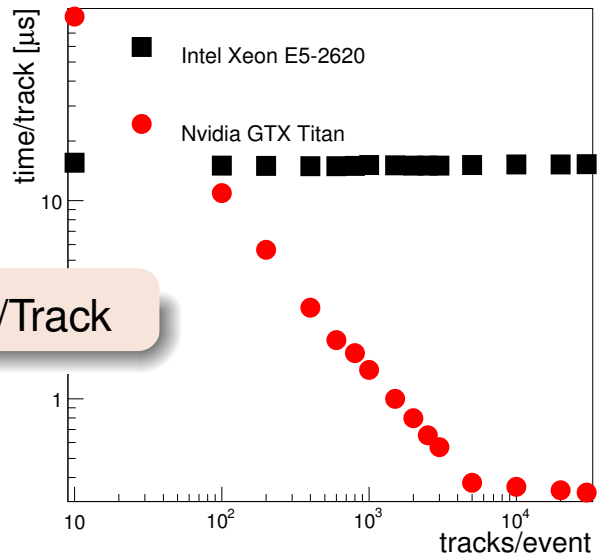
BACKUP

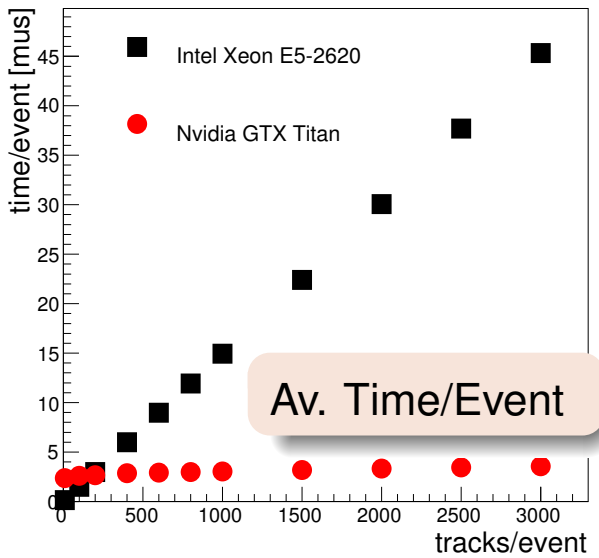
Standalone preliminary studies

- Track fitting on GPU - first test on the GAP server in Rome:
  - Input: raw track candidates from Inner Detector RoI
  - **Parallel fit of an increase number of track**



Av. Time/Track





Av. Time/Event

- GPU implementation less dependent on the track multiplicity:
  - promising for upcoming data taking conditions, with increased luminosity.