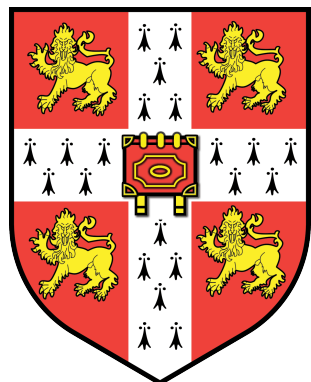


# ATLAS Simulation Computing Performance and Pile-Up Simulation in ATLAS

John Chapman

On behalf of the ATLAS Collaboration



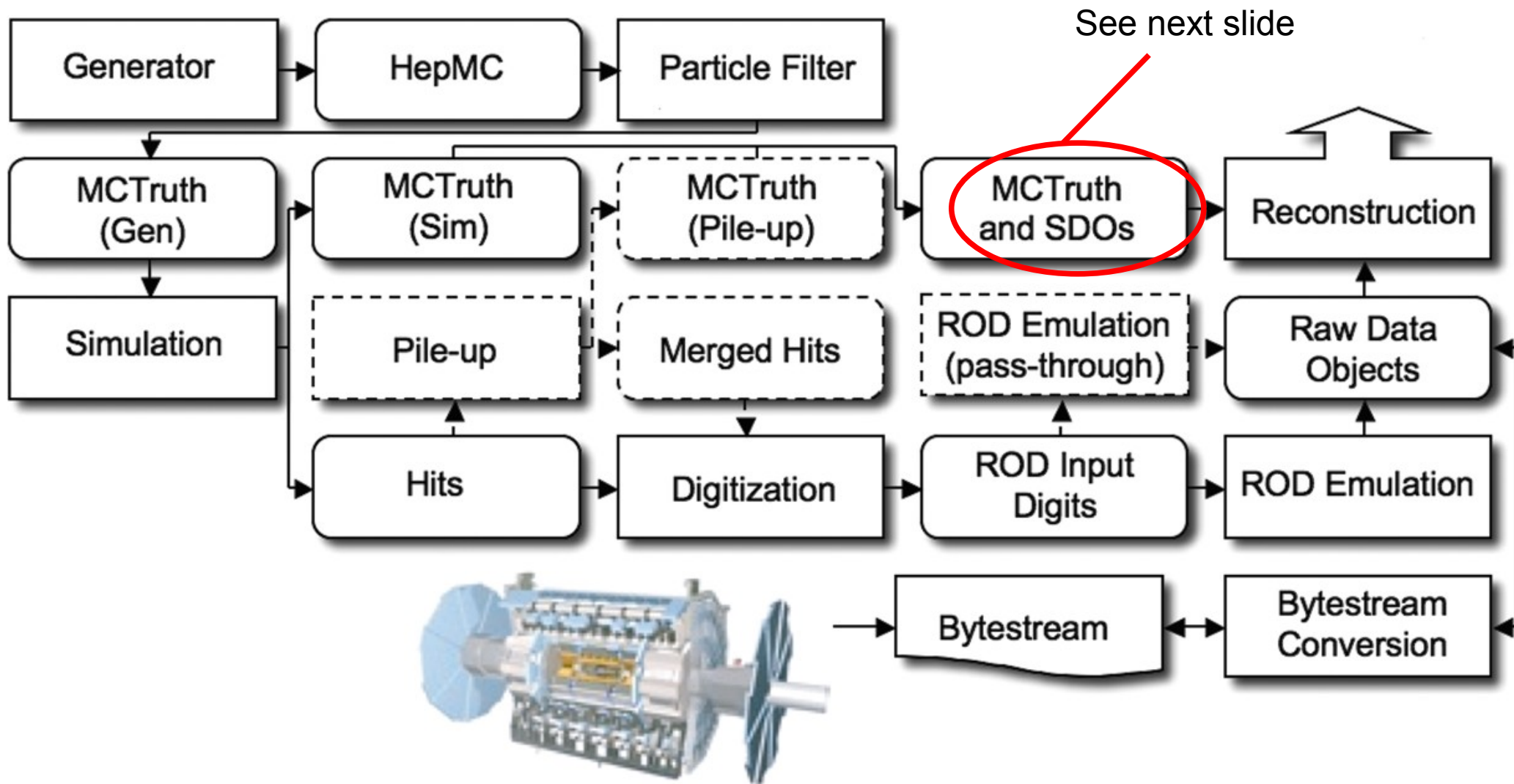
LPCC Detector Simulation Workshop  
6th-7th October 2011, CERN



# Overview

- Overview
  - Simulation Flow in ATLAS
  - Simulated Data Objects
- Simulation Performance
  - CPU
  - vmem
- Techniques for Improving Simulation Performance
  - Importance of Reproducibility
  - Frozen Showers
  - GEANT4 Magnetic Field Integration
- Pile-Up Simulation in ATLAS
  - Background Simulation:
    - Samples
    - Bunch Structure
    - Variable Luminosity
  - ATLAS Reconstruction: Impact of Pile-Up
  - Pile-up Performance
  - PileUpTools
- Summary

# Simulation Flow in ATLAS



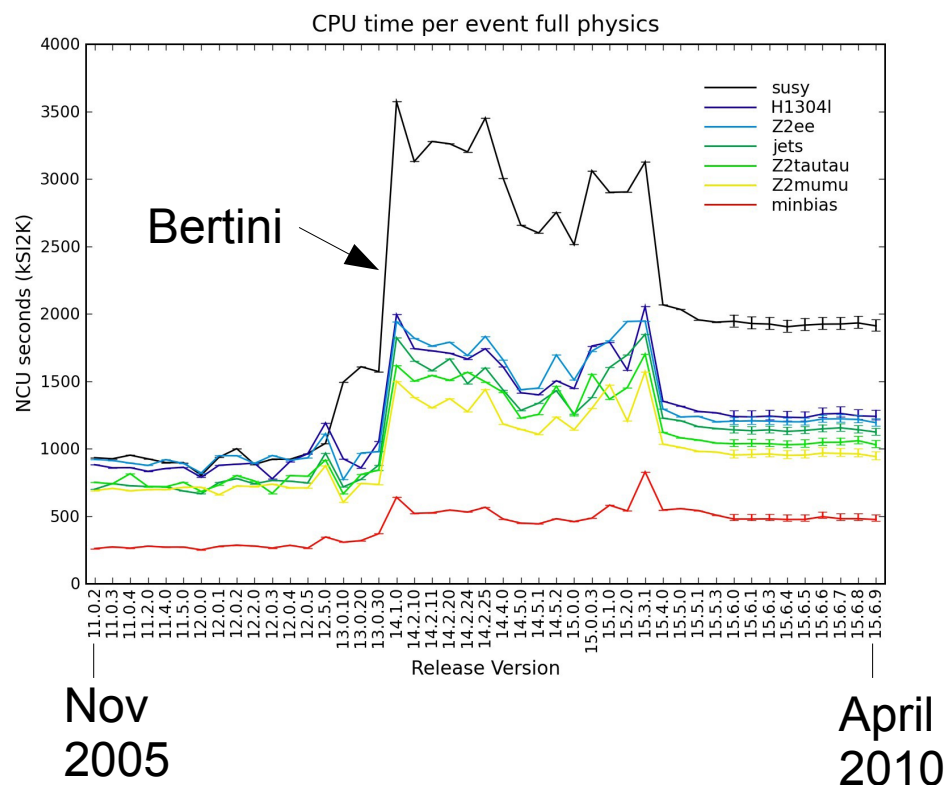
# Simulated Data Objects

- **Simulated Data Objects** (SDOs) are written out to the Raw Data Object (RDO) pool files (output of the Digitization step).
- SDOs are created by digitization algorithms for Inner Detector and Muon Spectrometer sub-detectors.
  - One SDO per hit channel.
  - Provide a link between the RDOs and the Truth information.
  - Contain a list of all the contributions to the response of that channel.
- **CalibrationHits** perform a similar function to the SDOs for the Calorimeters (see O Arnaez's talk yesterday). They are only written out for certain samples.

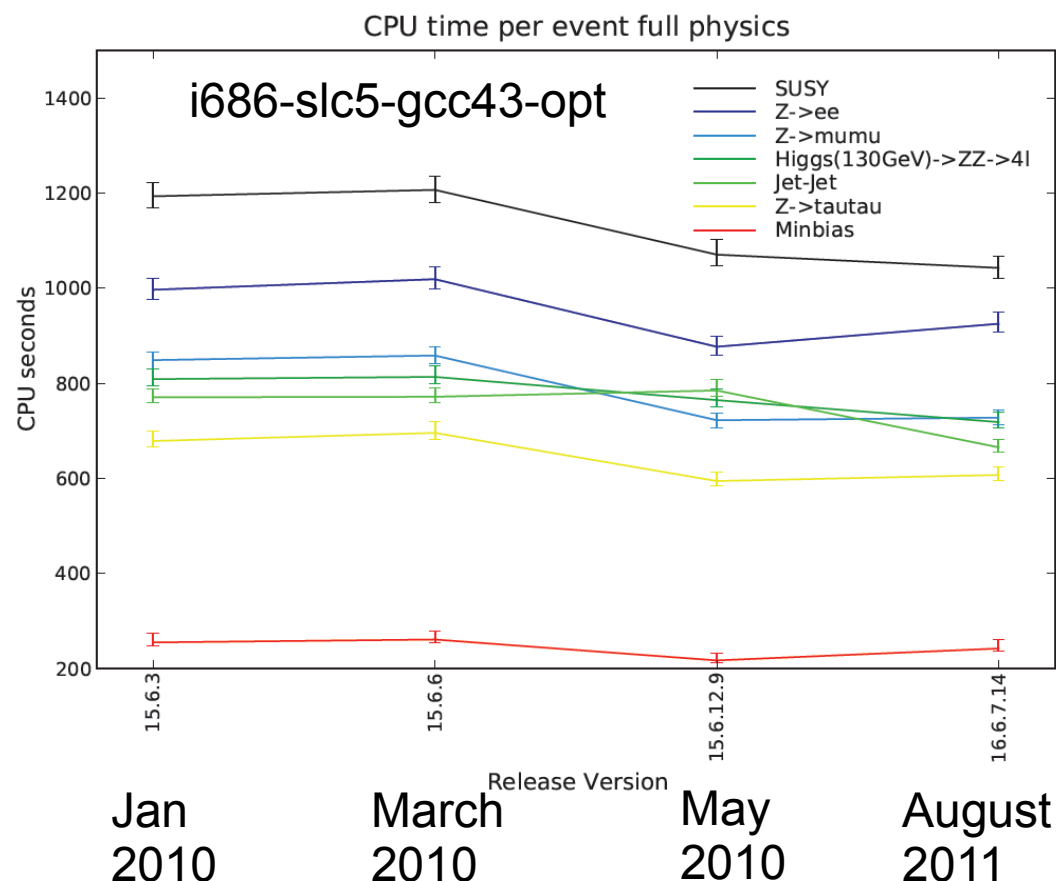
# Simulation Performance: CPU requirements

## Historical Trend

Plots by B O'Brien



CPU clock frequency 2.45GHz, scaled to 1.0GHz.

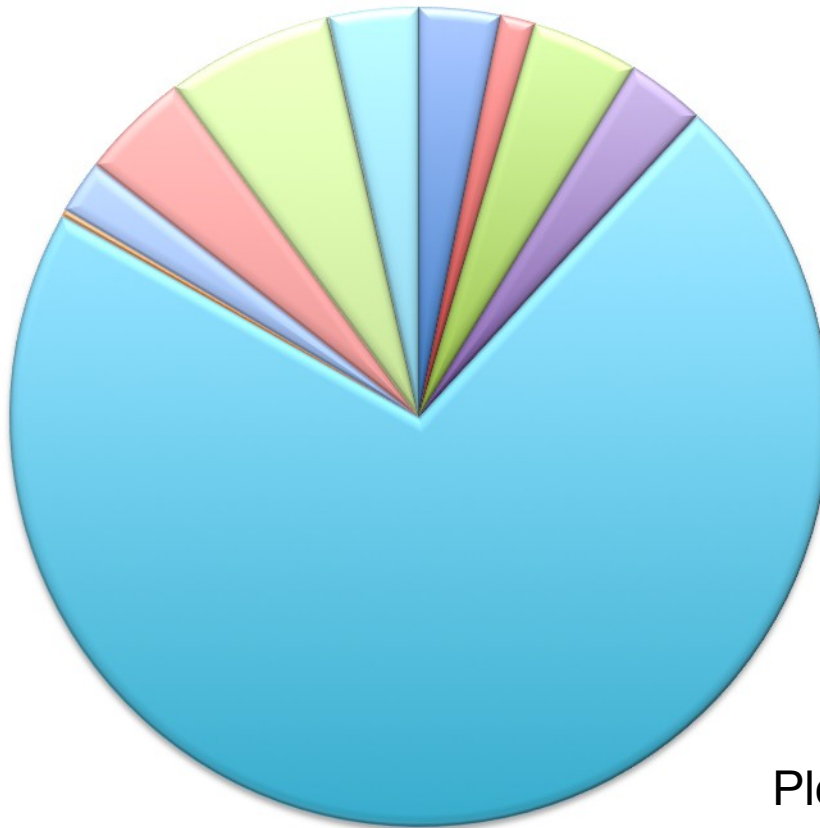


The standard simulation in ATLAS covers the region with  $|\eta| < 6.0$ .  
 Can optionally simulate far-forward detectors too.

# Simulation Performance: CPU breakdown

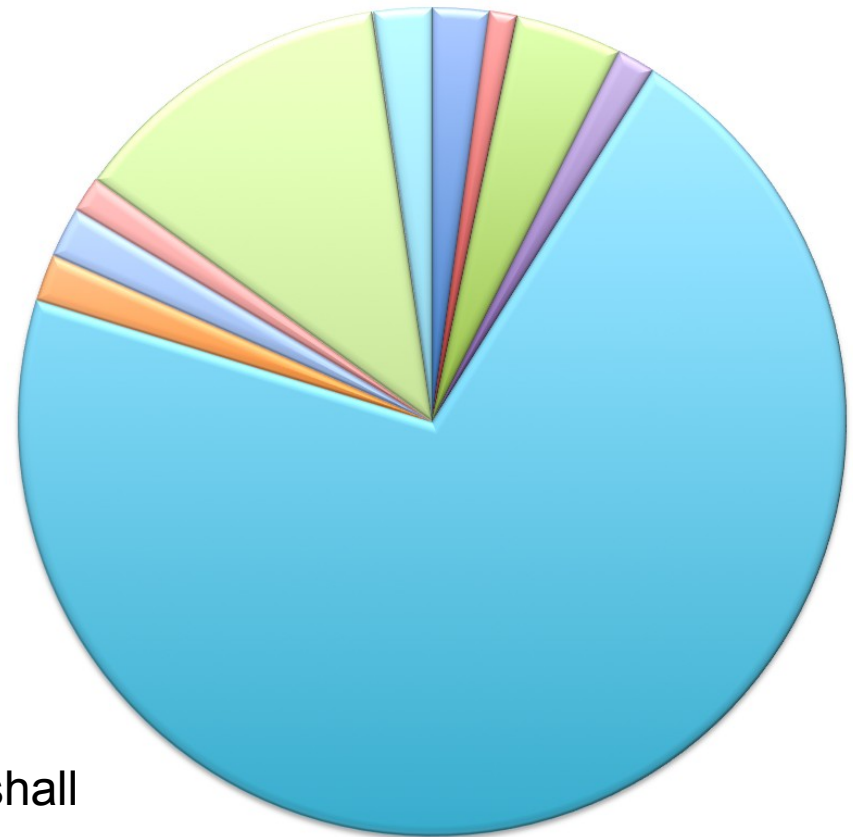
Minimum bias Simulation (with Frozen Showers)  
Total CPU per event = 71.7 s

i686-slc5-gcc43-opt



$\bar{t}\bar{t}$  Simulation (with Frozen Showers)  
Total CPU per event = 346.1 s

i686-slc5-gcc43-opt



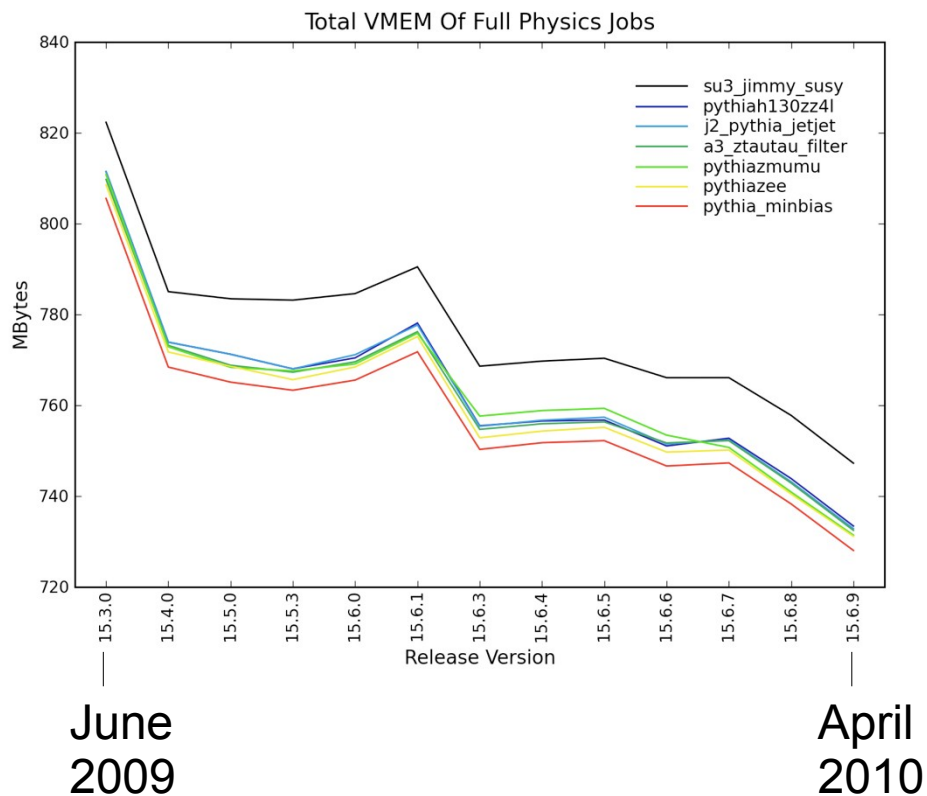
- Pixel
- SCT
- TRT
- Other ID
- LAr EM Cal
- Tile Cal
- LAr Had Cal
- FCAL
- Other Cal
- Muons
- Other

Plots by Z Marshall

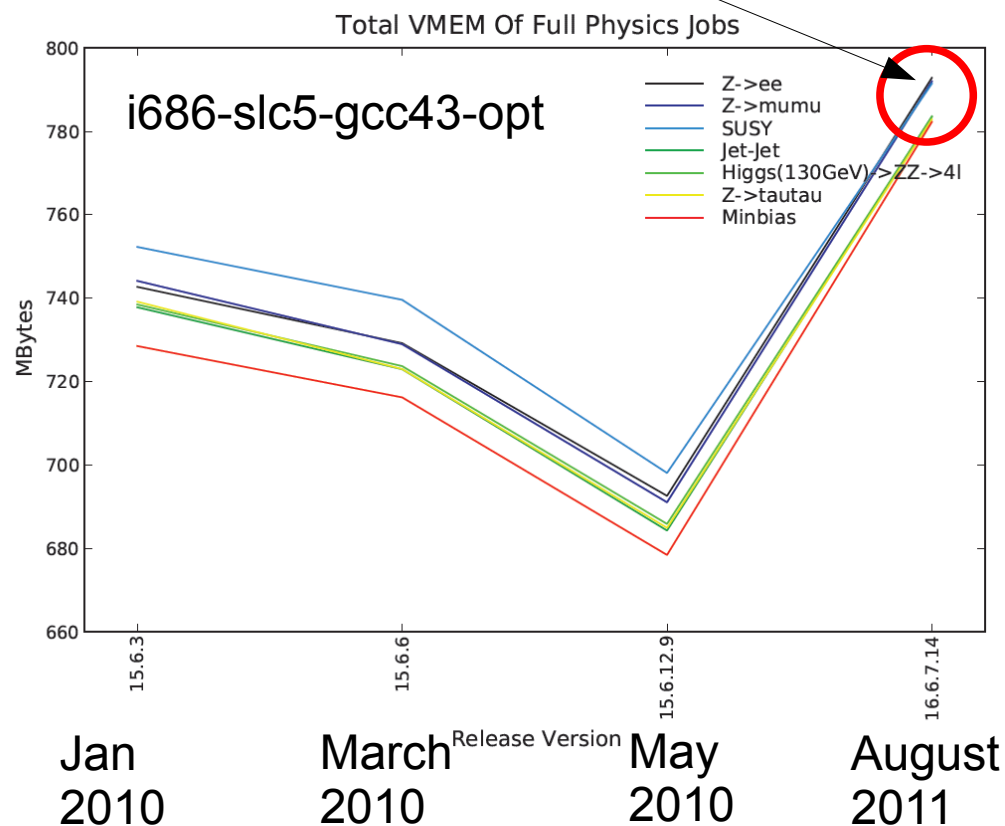
The LAr EM Cal contribution dominates... and this itself is dominated by time spent simulating the End Caps. (For Frozen Showers see later.)

# Simulation: Vmem requirements

## Historical Trend



Extra dead material in the Muon System and switch to 'detailed LAr geometry' in the newest release *may* explain this jump. (Still under investigation.)



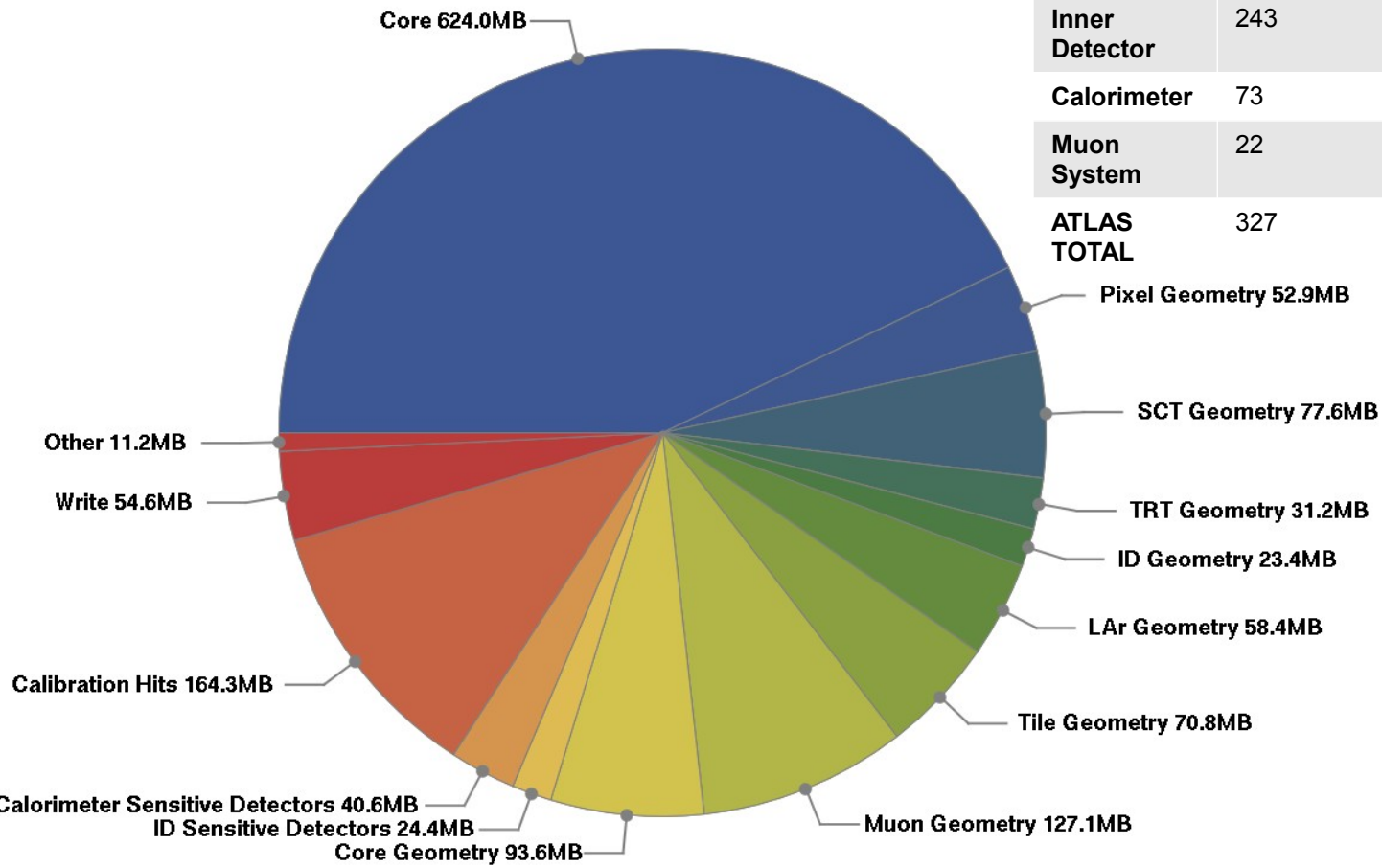
Plots by B O'Brien

# Simulation vmem breakdown

Contribution to vmem [total 1454.1MB]  
 x86\_64-slc5-gcc43-opt

sub-system	Materials	Solids	Logical Volumes	Physical Volumes	Total Volumes
Beampipe	43	195	152	514	514
BCM	40	131	91	453	453
Inner Detector	243	12,501	18,440	56,838	1,824,616
Calorimeter	73	52,366	35,864	182,282	1,557,459
Muon System	22	33,594	9,467	76,945	1,424,768
<b>ATLAS TOTAL</b>	<b>327</b>	<b>98,459</b>	<b>63,379</b>	<b>316,043</b>	<b>4,086,839</b>

(The above numbers are from 2010. Calorimeter and Muon System numbers will have increased recently.)





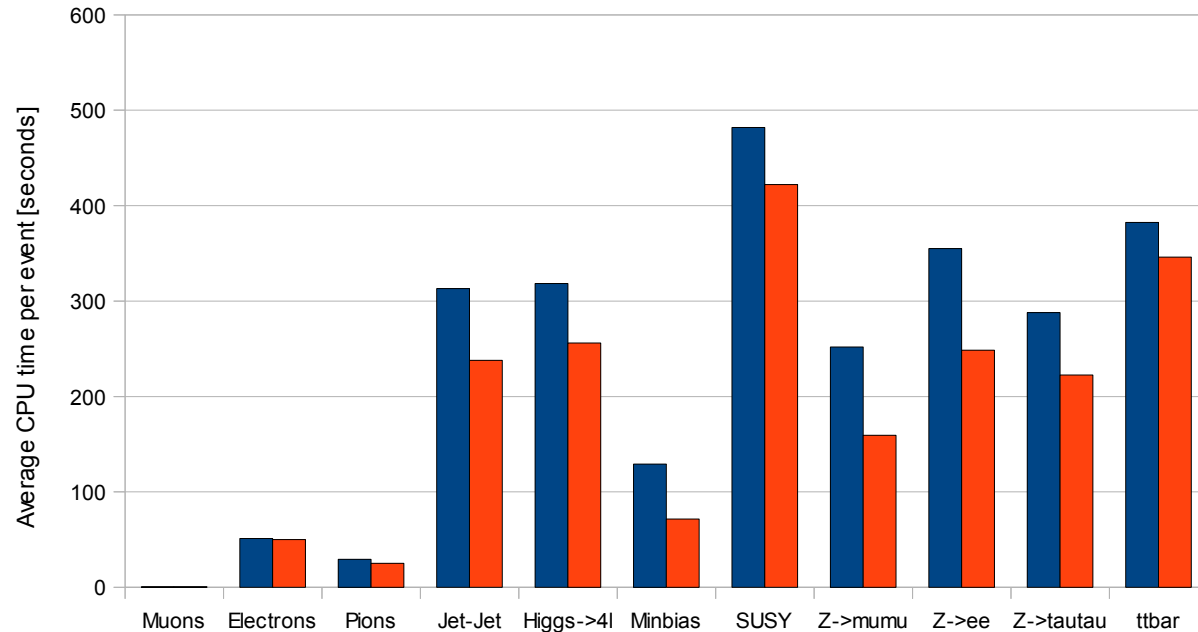
# Importance of Reproducibility

- Ability to reproduce individual events is crucial when debugging.
- Current **crash rate per event simulated is  $\sim 5E-6$**  on the grid.
- **Really** want to avoid waiting for 999 events to be simulated just to observe a crash on event 1000...
- Recent improvements mean the simulation of a single event is completely reproducible in ATLAS:
  - At the start of each event:
    - Random Number streams are re-seeded based on the run and event numbers and an offset specified on the command-line.
    - Discard any cached random numbers in CLHEP::RandGauss.
  - Ensured all random number calls use the right random number engine.

# Frozen Showers

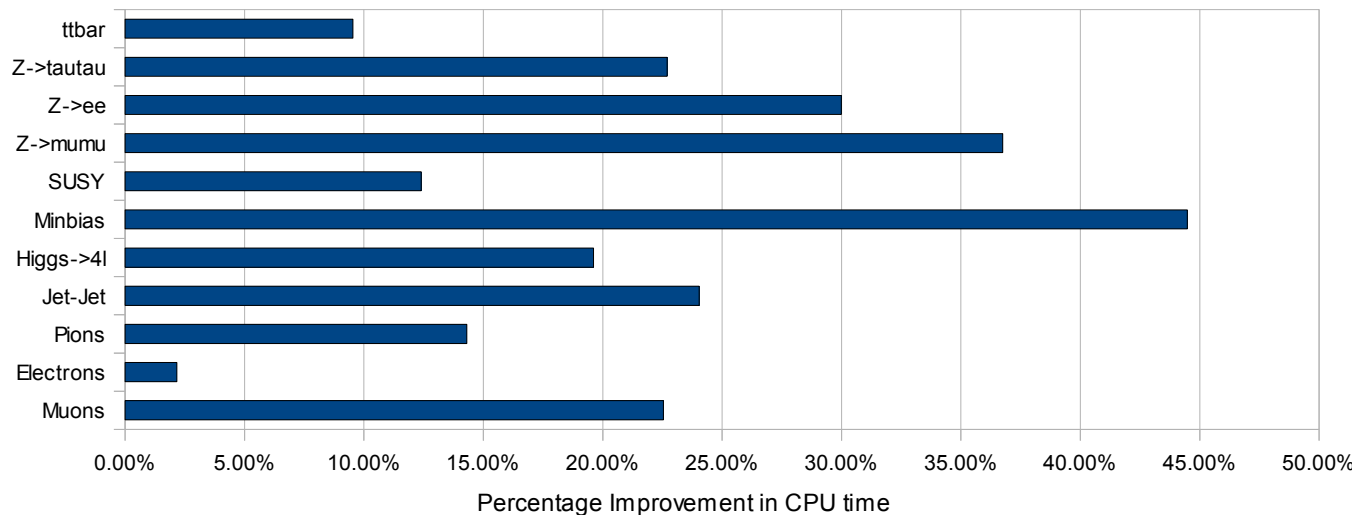
- Method of fast shower simulation:
  - Create a library of pre-simulated showers, using the full simulation (Frozen Showers).
  - Particles with  $E \geq 1$  GeV:
    - Use Full simulation.
  - EM particles ( $e^\pm, \gamma$ ) with  $E < 1$  GeV and  $T_n < 150$  MeV neutrons:
    - Substitute each particle with a 'Frozen Shower'.
- The transverse and longitudinal size of Frozen Showers can be easily modified: try to tune directly to the data.
- Output format identical to full simulation.
- Frozen Showers are implemented for all LAr calorimeters and **deployed for the Forward Calorimeter in ATLAS Production.**

# Frozen Showers Performance



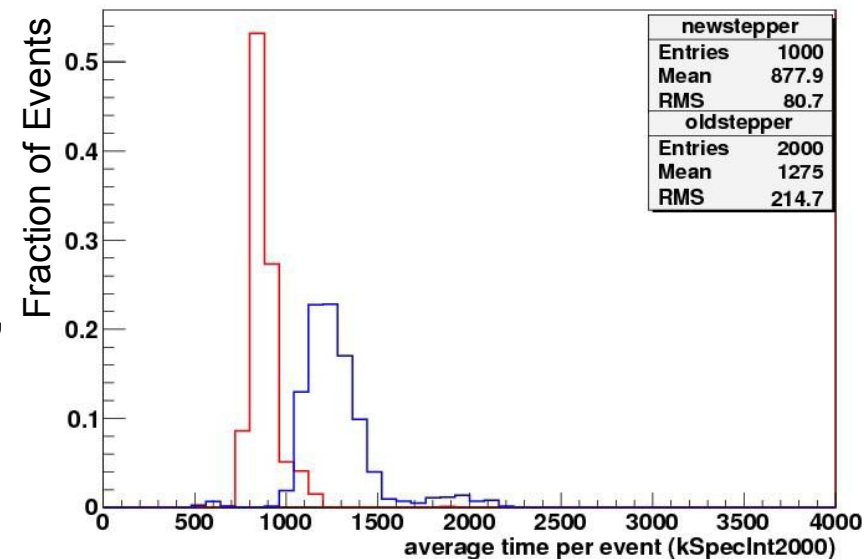
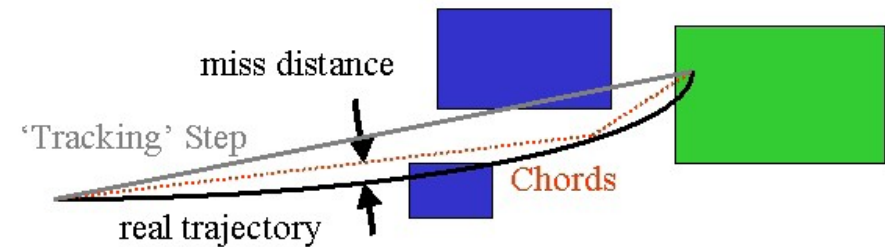
In 2011 Simulation production, frozen showers are used to simulate the FCAL only, but the reductions in CPU time are significant. ~45% for minimum bias.

■ Full Simulation  
 ■ Frozen Showers



# GEANT4 Magnetic Field Integration

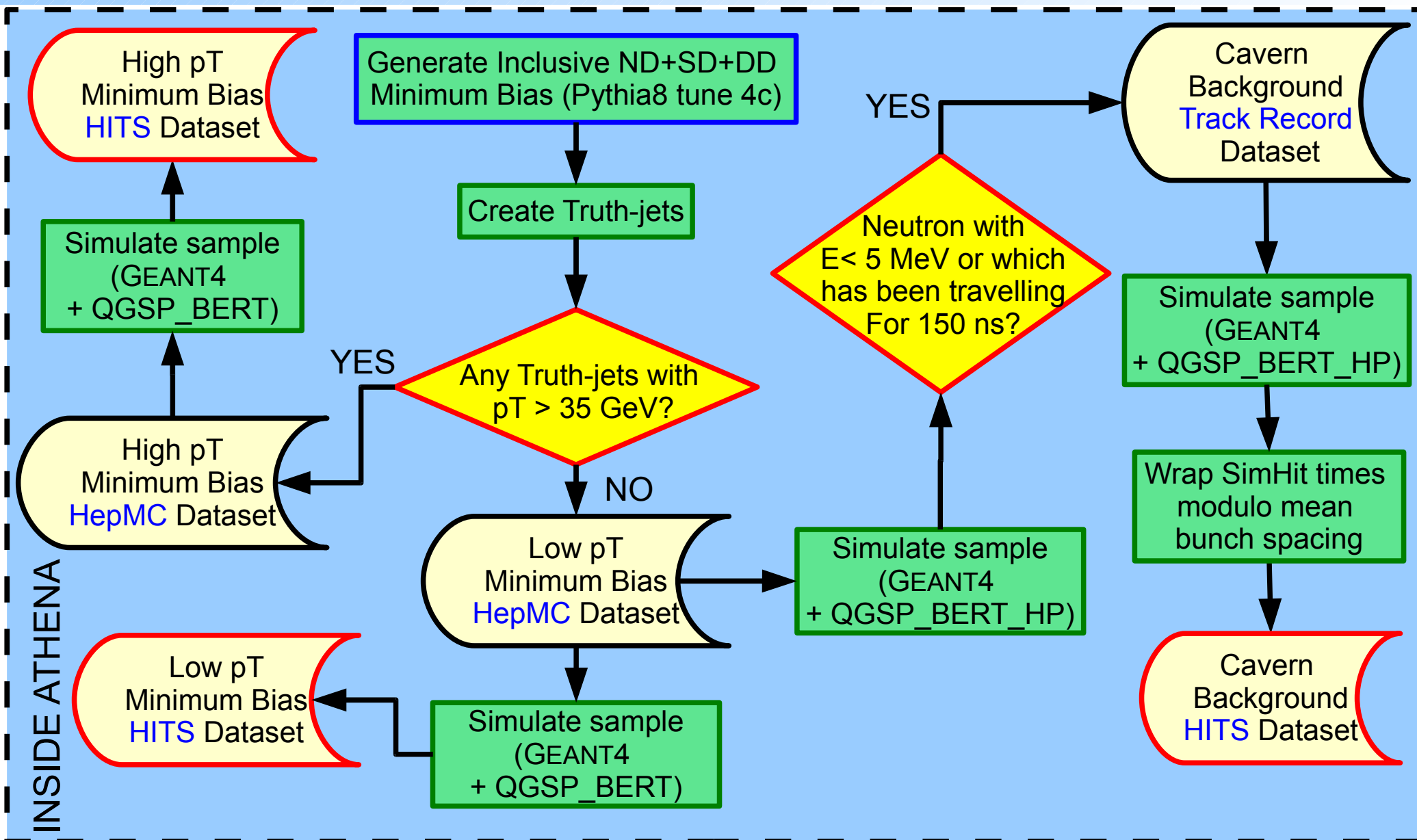
- Lots of simulation time spent on EM field integration steps...
- Switched from using **G4ClassicalRK4** Stepper to the **AtlasRK4** Stepper in Simulation.
- **AtlasRK4** (20-30% CPU improvement!)
  - based on work done by the ATLAS tracking group.  
[E Lund et al 2009 JINST 4 P04001](http://dx.doi.org/10.1088/1748-0221/4/04/P04001)  
<http://dx.doi.org/10.1088/1748-0221/4/04/P04001>
  - uses adaptive RK-Nystrom integration, intermediate calc. steps cached and adaptive local error estimation.
- This technique has now been integrated into GEANT4 as the **G4Nystrom** Stepper.



# Simulating Pile-Up in ATLAS (I)

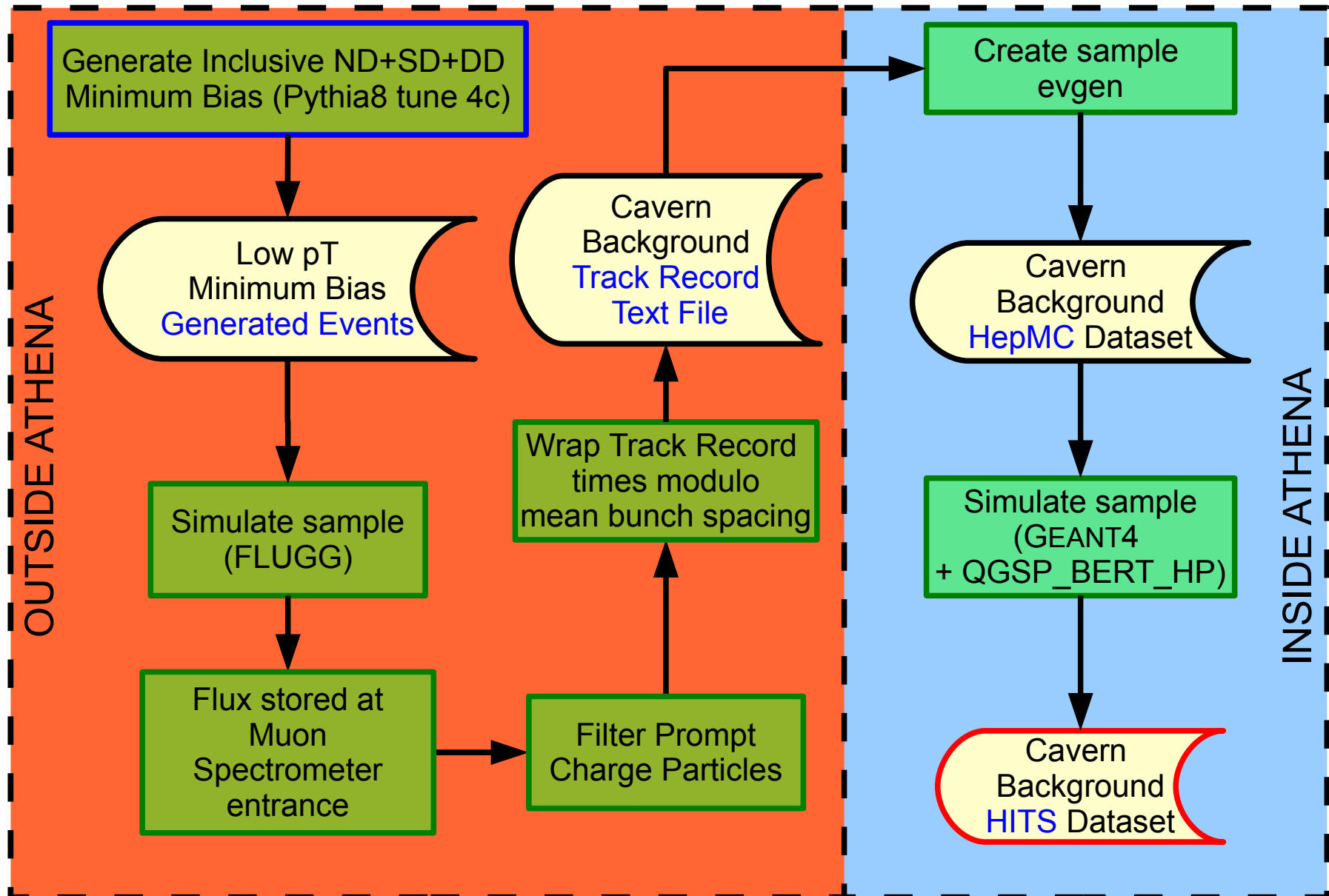
- We are now in a regime where we observe **multiple pp collisions** in each filled LHC bunch-crossing and multiple filled bunch-crossings within the **[-800,800] ns sensitive time window** of ATLAS.
- Simulation in the Athena framework proceeds as follows:
  - Run the event generation and (GEANT4) simulation steps for single pp interactions.
  - Combine multiple simulated pp interactions during the digitization step (“**Pile-up Digitization**”).
    - Attempts to reproduce this situation by digitizing the HITS from many simulated pp interactions all together.
    - This includes both in-time and out-of-time pp interactions.
    - “cavern background” events are also added (see next few slides and L Jeanty's talk yesterday).

# Background Sim: Pythia8 + GEANT4



INSIDE ATHENA

# Background Sim: FLUGG + GEANT4



# Simulating Pile-Up in ATLAS (II)

## Terminology

- $\mu$  = interactions per crossing averaged over a specific lumi block for a specific BCID (bunch crossing ID)
- $\langle\mu\rangle$  = interactions per crossing averaged over a specific lumi block and over all colliding BCIDs

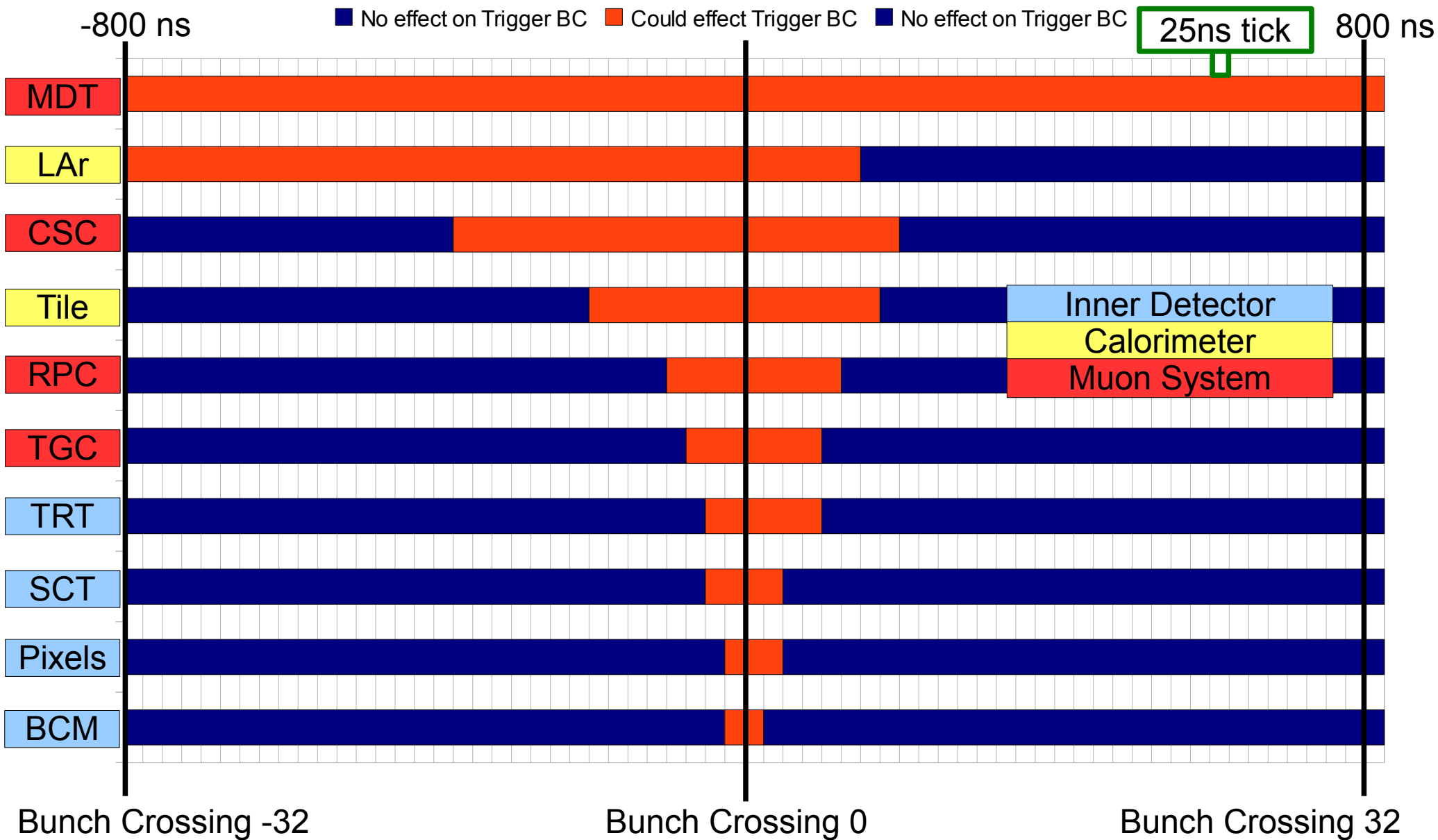
Background type determines how events are added to the signal event:

- **Minimum Bias**: Add a random number of events picked from a Poisson distribution with mean  $\langle\mu\rangle$  to each colliding BCID.
- **Cavern Background**: Add a constant number of events to each BCID. Rate depends on  $\langle\mu\rangle \times$  fraction of colliding BCIDs.

Offset event times according to the BCID they are used.



# Simulating Pile-Up in ATLAS (III)



# Simulating Pile-Up In ATLAS (IV)

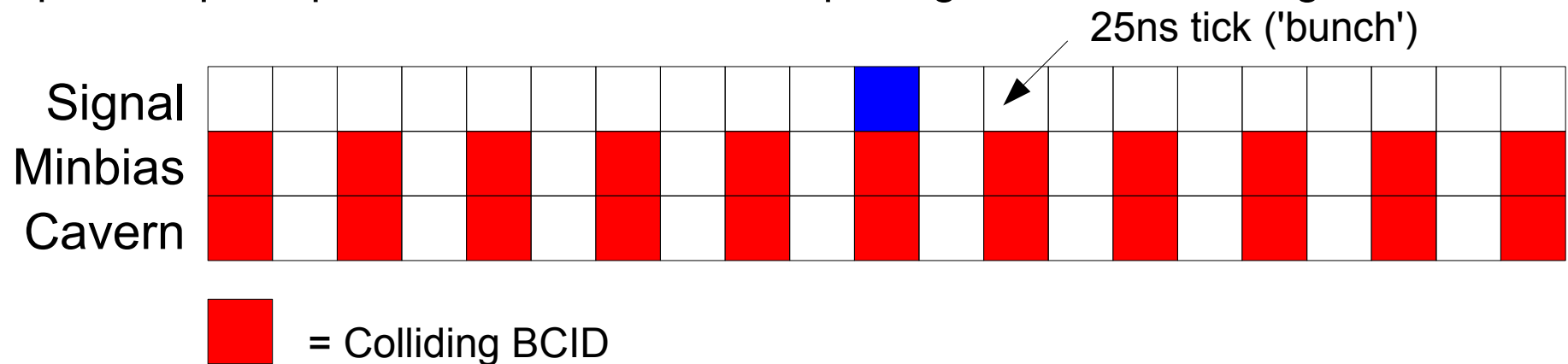
- Generating huge samples of background events = Expensive!
- Disk constraints limit the maximum sample size.
- Create a cache of background events in memory, so they can be re-used.
- Save memory by only reading in/caching the parts of each event which are needed.
- After a cached event is used, it may be replaced by a fresh event.

Background Type	In-time/Out-of-time	Replacement Probability
High pT Minimum Bias	Both	100%
All	In-time	100%
Low pT Minimum Bias	Out-of-time	~1% (tunable)
Cavern Background	Out-of-time	~1% (tunable)

- Cache size dominated by the size of Truth information.

# Background Simulation: Bunch Structure

Example of a pile-up model with fixed 50ns spacing between colliding BCIDs:



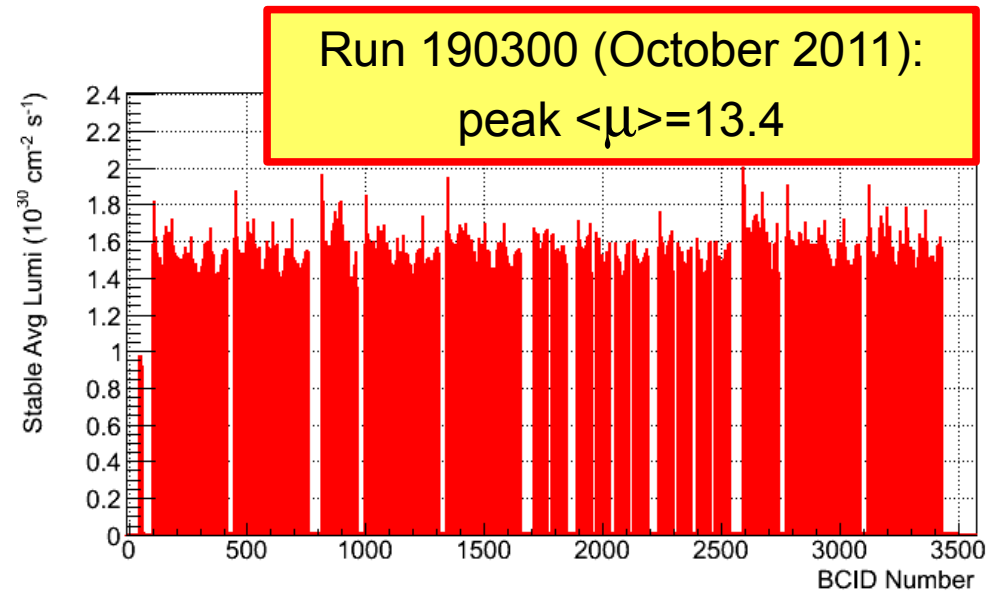
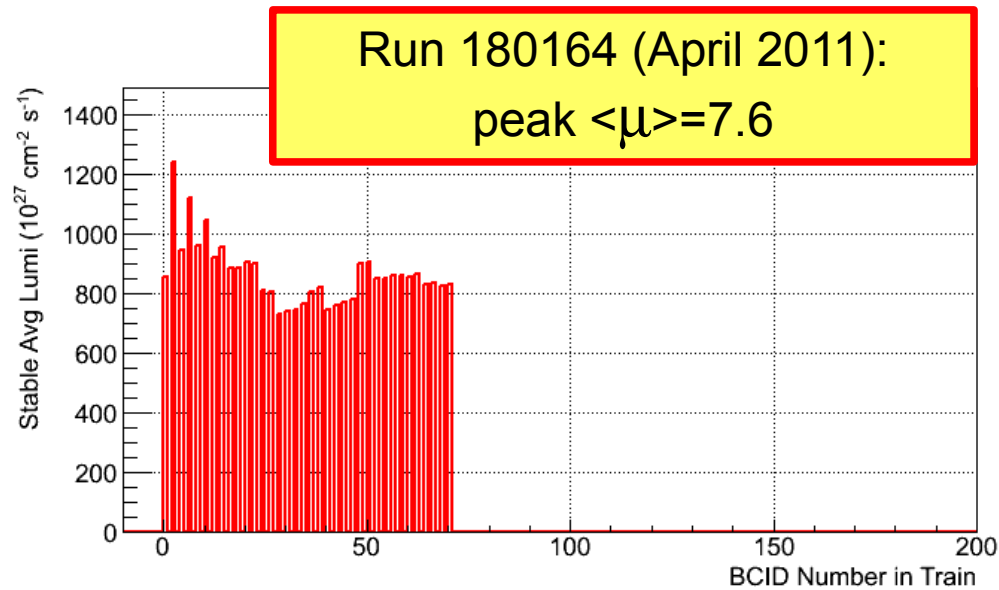
In reality the structure of colliding and non-colliding BCIDs can be more complicated.



- The pile-up/detector response is affected by the position of the triggering BCID in the bunch train (see later).
- Bunch structure modelling is included in the pile-up simulation.
  - Patterns can be up to 3564 elements in length and wrap-around if required.
  - Each triggering BCID is picked from the colliding BCIDs in the pattern, with a probability proportional to the relative luminosities of each bunch crossing.

# Background Simulation: Variable Lumi. (I)

- Well known that  $\langle\mu\rangle$  varies over time.

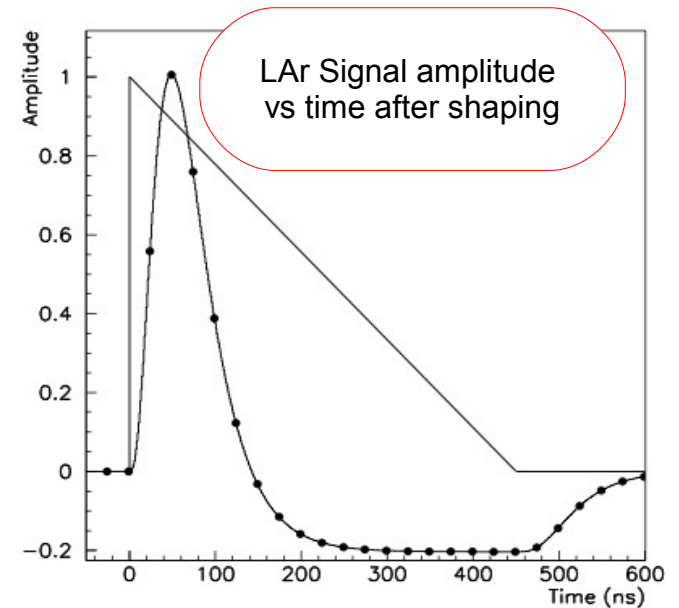
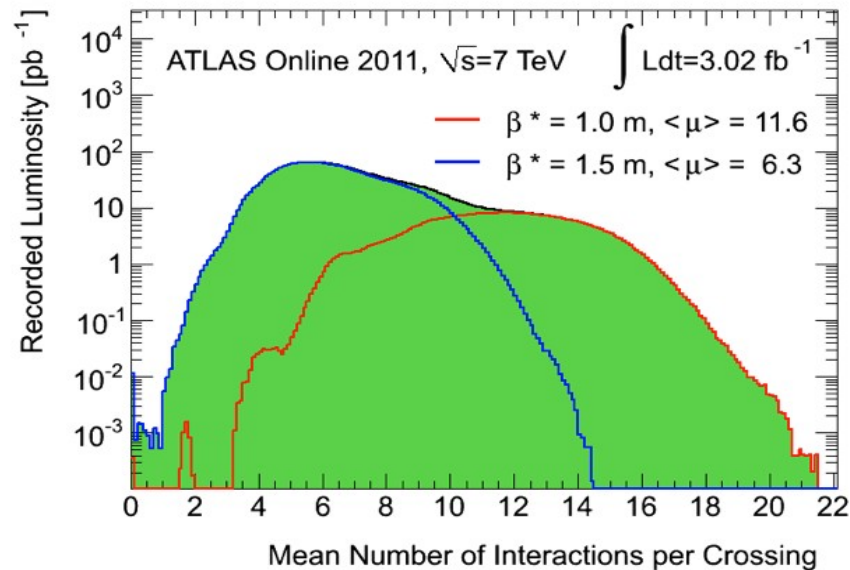


- $\mu$  can also vary greatly from BCID to BCID in data, as the plots above show.
- Both in-time and out-of-time pile-up effects are important.
- Problem:**
- Simulating samples at a fixed  $\langle\mu\rangle$  value makes it difficult to re-weight MC to data...

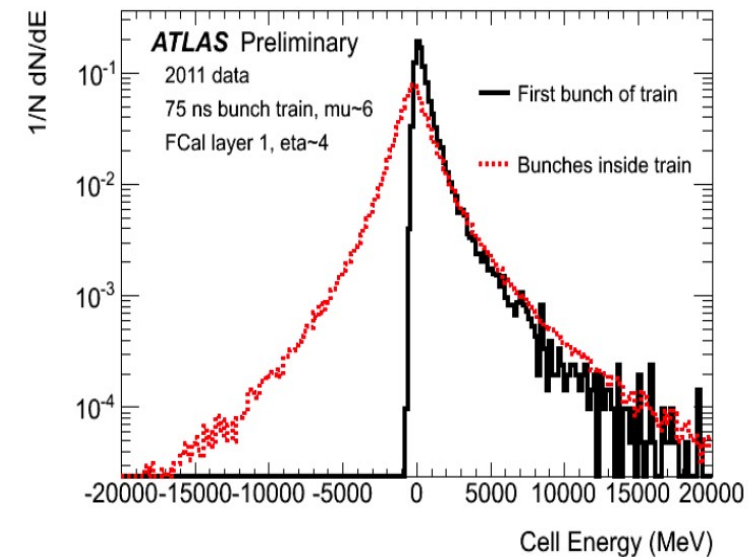
# Background Simulation: Variable Lumi. (II)

- **Solution:**
  - Use a range of  $\langle\mu\rangle$  values within each simulated sample.
    - The  $\mu$  and  $\langle\mu\rangle$  value used are recorded for each event.
    - This can then be used to re-weight the MC sample to match a given set of data periods.
  - So far have only used configurations where  $\mu=\langle\mu\rangle$  or  $\mu=0$  in production.
  - Will start using bunch pattern configurations with variable mu values (i.e.  $\mu\neq\langle\mu\rangle$  or 0) in the next round of MC production to improve our understanding of out of time pile-up.

# ATLAS Reconstruction: Impact of Pile-Up (I)

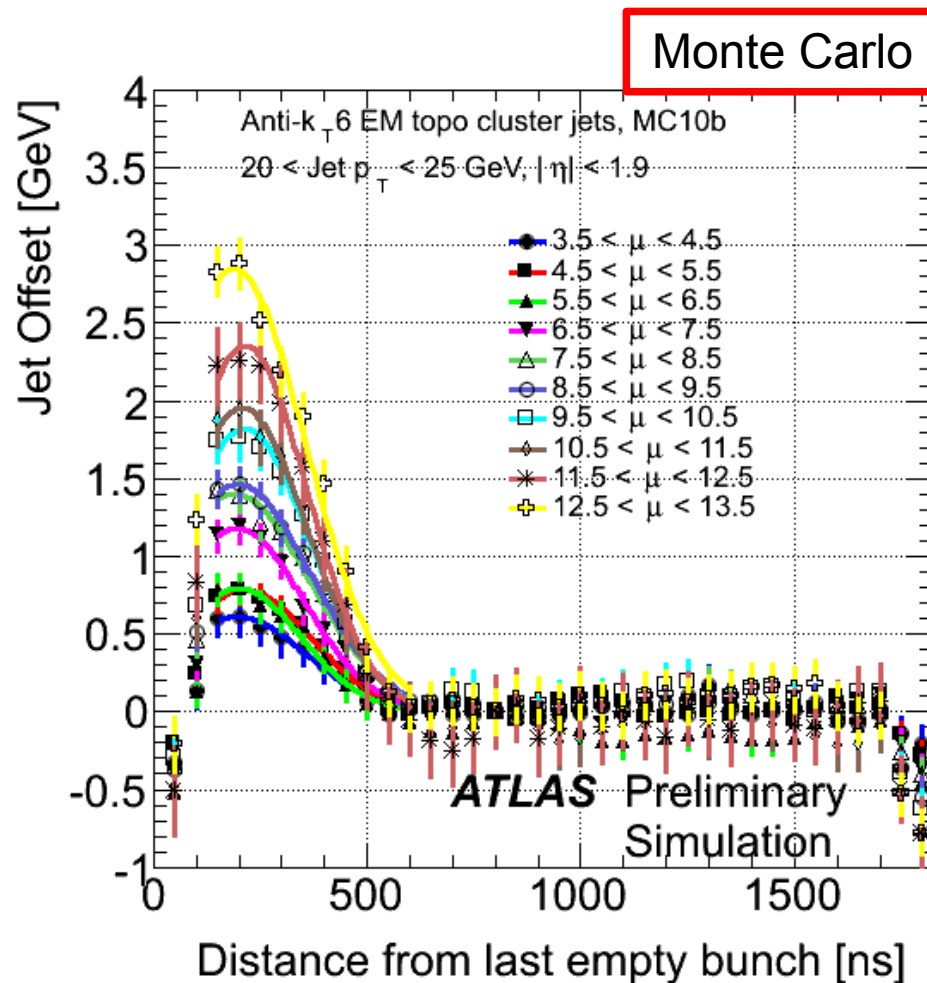
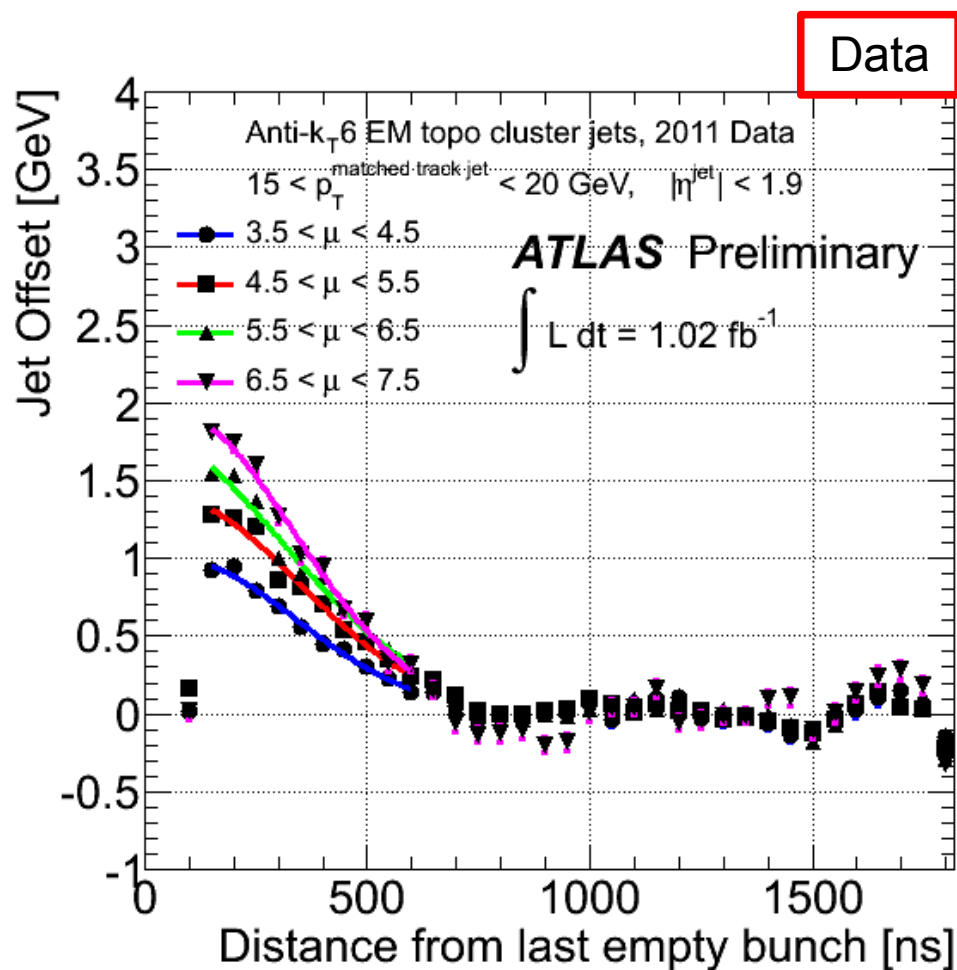


- Do not expect a significant impact on tracking, nor muons, nor even electrons and photons.
- But sizeable impact on jets (+ETmiss) and  $\tau$ .
- LAr drift-time is  $\sim 500$  ns and out-of-time bunches have impact on measurement.
- Bipolar pulse shaping designed so that  $\langle ET \rangle \sim 0$  for 25 ns bunch-spacing and uniform intensity per BX.
- Optimal performance will require correction per cell type in  $\eta$ -bins and as a function of luminosity to set average measured ET to  $\sim 0$ .



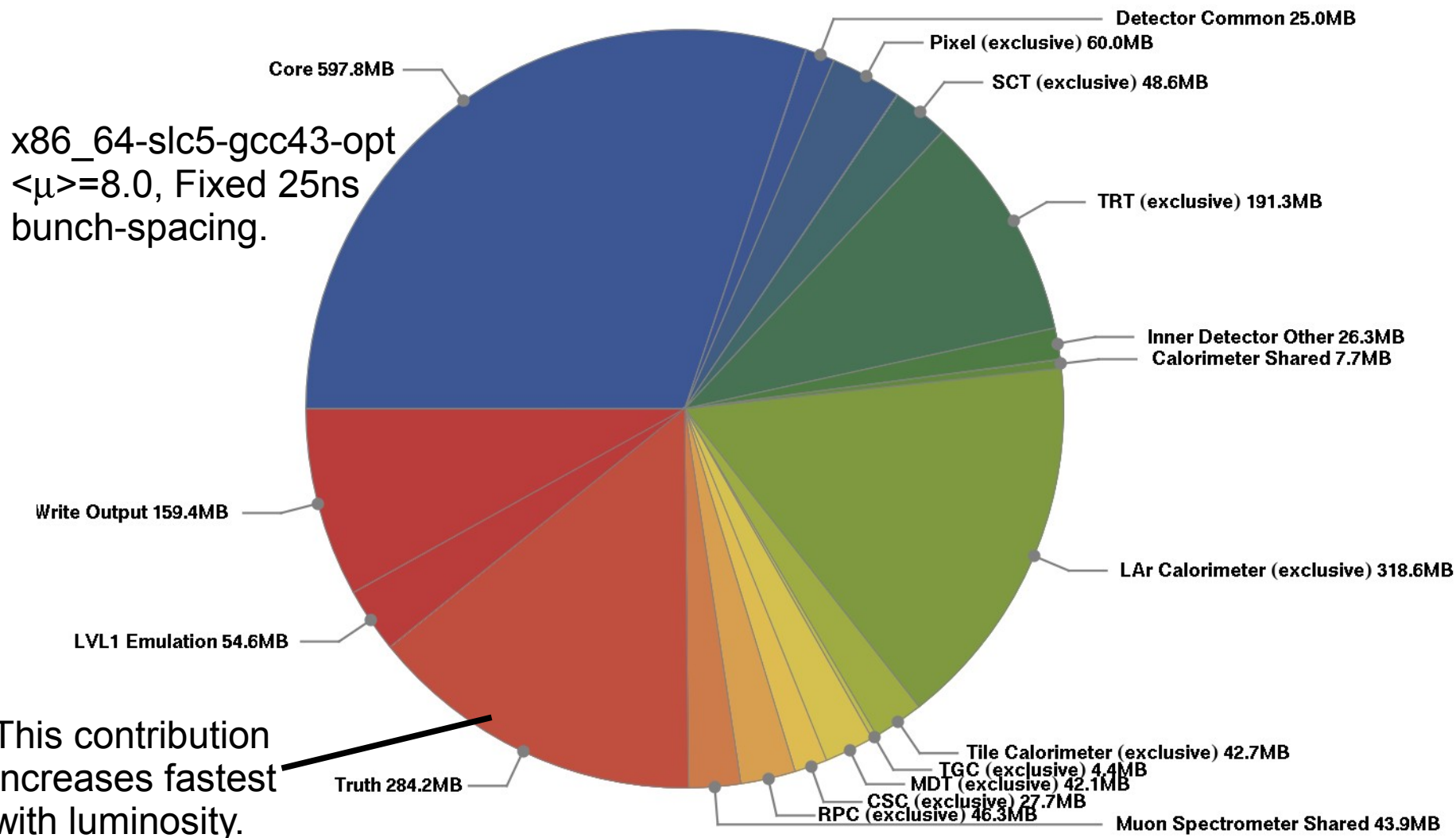
# ATLAS Reconstruction: Impact of Pile-Up (II)

- Jet offsets from pile-up are modelled to <50%
- Remaining differences from **BCID-to-BCID beam current variation** were not modelled in MC10b.



# Pile-up Digitization: vmem breakdown

Contribution to vmem [total 1980.8MB]



x86\_64-slc5-gcc43-opt  
 $\langle \mu \rangle = 8.0$ , Fixed 25ns  
 bunch-spacing.

This contribution  
 increases fastest  
 with luminosity.

One approach to save memory under validation is to filter truth info in the background HITS files.



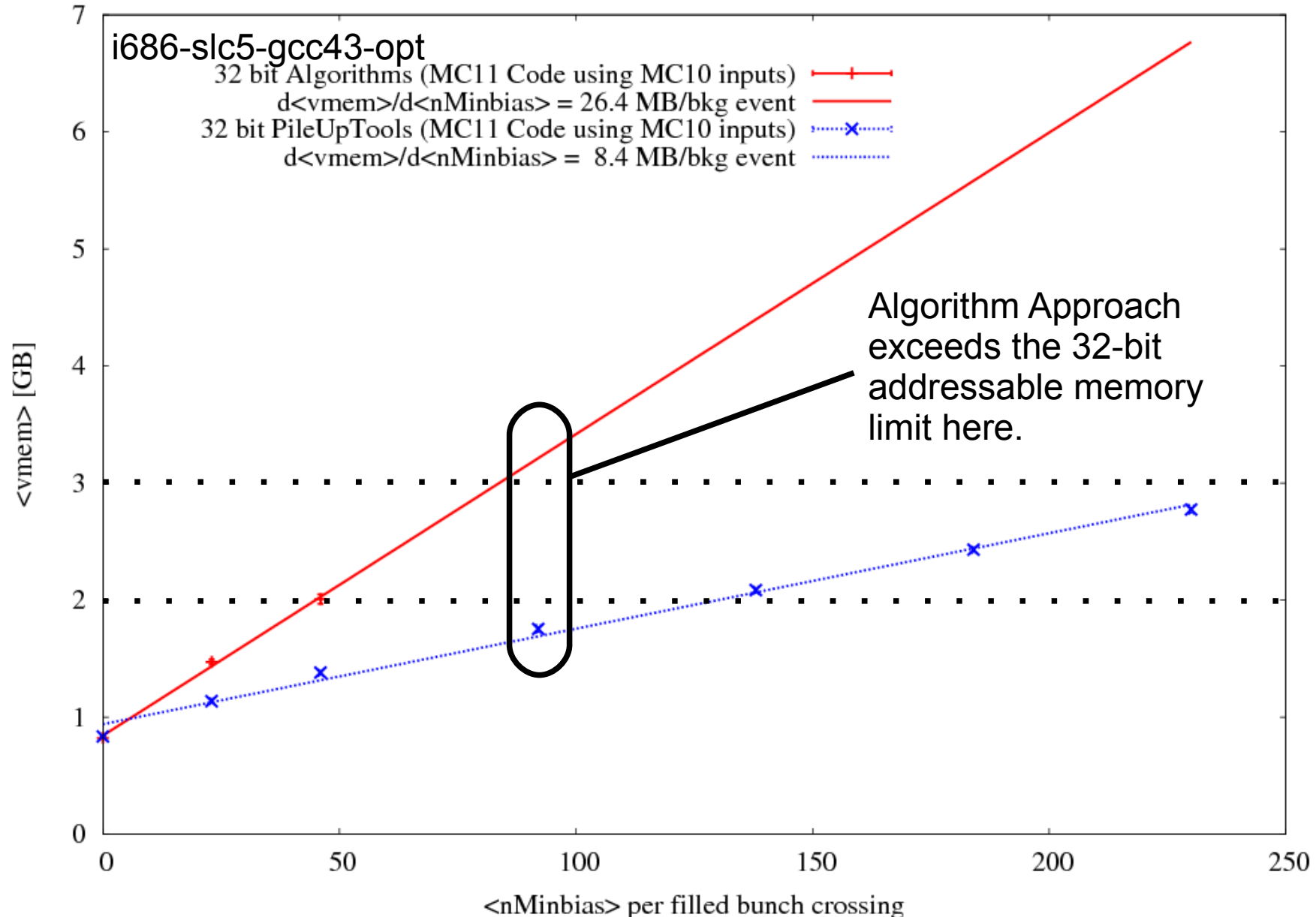
# Simulating Higher Luminosities

- For High luminosities **previous pile-up approach has issues...**
- Consider a typical upgrade scenario:
  - 200 pp-collisions per colliding BCID
  - fixed 50ns spacing between colliding BCIDs
  - ATLAS would be sensitive to 33 colliding BCIDs
    - $33 \times \sim 200 \times 2 = \mathbf{O(13200)}$  background events (**minimum bias+cavern**) required per single signal event!
- Having this many simulated events in memory at once is not feasible, so an alternative must be found...

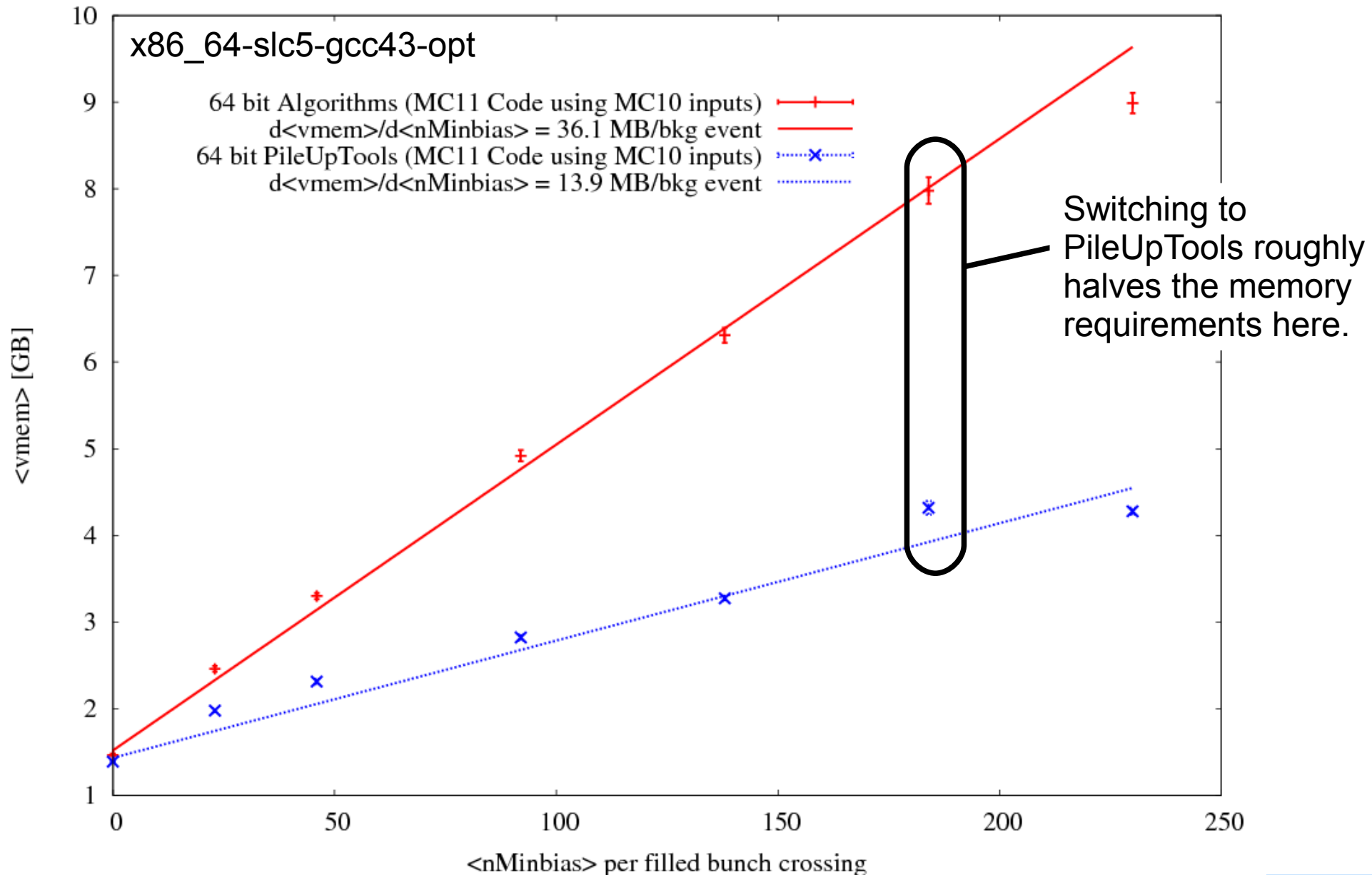
# PileUpTools: BC by BC Pile-Up

- The previous pile-up approach (AKA the “Algorithm” approach) :
  - digitizes the information from all required bunch crossings for a given sub-detector before moving on to the next sub-detector.
  - Background event info cached to allow re-use.
- The “PileUpTools” approach:
  - provides one filled bunch crossing at a time to all sensitive sub-detectors.
  - Background events are read as required and discarded from memory after each filled bunch crossing is processed.
    - Sacrifice caching of background to save memory.
    - Resulting increase in I/O Time means an increased wall-clock time.
  - A single pile-up Athena Algorithm calls an Athena AlgTool for each sub-detector. The AlgTools know the time window for which they are sensitive to bunch crossings.
  - Digits/RDOs are produced from **intermediate information cached locally by the sub-detector tools**, after all filled bunch-crossings have been processed.

# PileUpTools Memory Savings (32-bit)



# PileUpTools Memory Savings (64-bit)

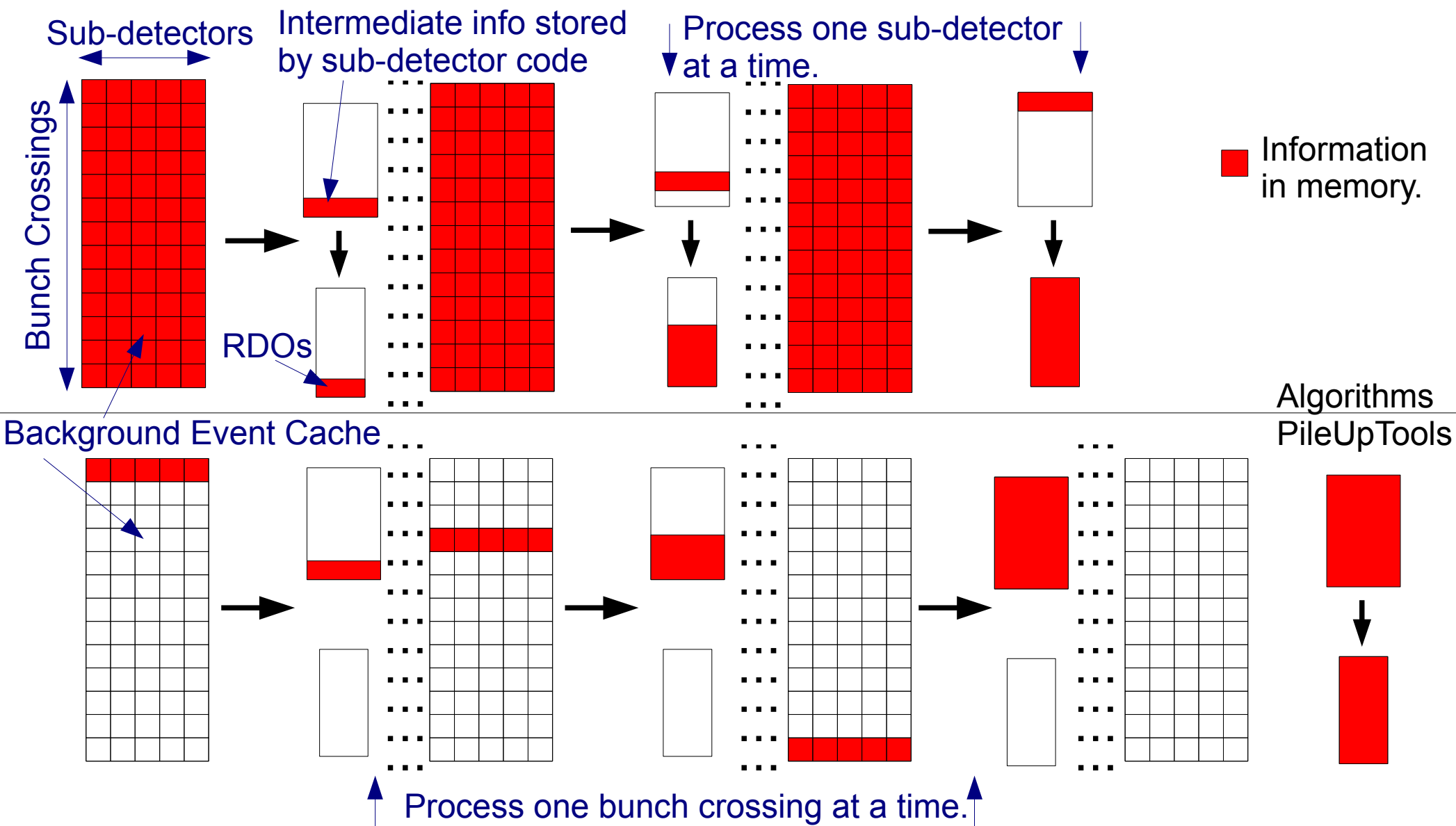


# Summary

- Simulation CPU requirements have been improving over time. The highly detailed nature of the full simulation means that it is still quite slow, so there are still good reasons to use fast simulations in some cases.
- Techniques such as frozen showers and improved EM steppers have significantly improved simulation time.
- Pile-up has a significant effect on the reconstruction and so it is important that it is simulated correctly.
- Care is taken to balance the competing demands of minimising job size, repetition of background events and background sample size on disk.
- Simulation of variable  $\mu$  and  $\langle\mu\rangle$  values is also important to include.
- For intermediate luminosities filtering the truth information in background HITS files should pile-up simulation without sacrificing CPU performance.
- Simulation of of higher luminosities requires a new approach. The PileUpTools approach allows  $\langle\mu\rangle=200$  pile-up to be simulated without exceeding the limit of 32-bit addressable memory.

# Backup

# Algorithm and PileUpTools Approaches to Pile-up Digitization



# Background Simulation: Samples (I)

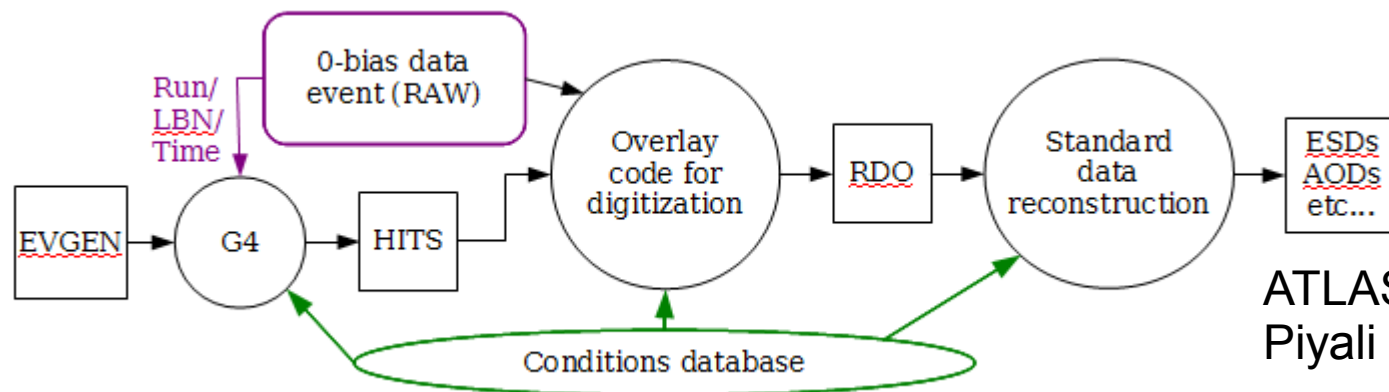
The prompt signal from pp collisions in the ATLAS detector is collected over only a few hundred nanoseconds. However, long after the collisions, a gas of low energy neutrons and photons is still present in the cavern. This gas is generally referred to as “**cavern background**.” This type of background is notoriously difficult to properly simulate, mostly due to the difficulties in correctly describing low energy neutron physics.

- ATLAS divides the particles from background pp-collisions into two parts:
  - The prompt signal from single background pp collision is simulated as a “**minimum bias**” event.
  - The low energy/long lived particles from this sample are dropped from the minimum bias sample simulation and simulated in a separate “**cavern background**” sample.
    - Assumed to be asynchronous, so the times of simulated hits are wrapped around modulo the mean spacing between filled bunches.
    - Muon detectors are most affected by high cavern-background rates.



# Using Data-Driven Background

- Standard Pile-Up simulation methods have allowed ATLAS to simulate conditions in the detector during beam running up until now.
- A new approach is under development “**Event Overlay**”. This approach **allows events to be combined at the RDO level**.
  - Allows MC events to be “overlaid” on Data “Zero-bias” triggered events.
  - Zero-bias triggers are read out one revolution later than a triggering BC.
  - Data driven background modelling, will automatically follow changing beam luminosity and detector conditions (including noise).
  - Includes beam gas, beam halo etc. automatically
  - Must be careful to use correct data conditions for simulation and digitization.



ATLAS Overlay Group  
Piyali Banerjee et al.