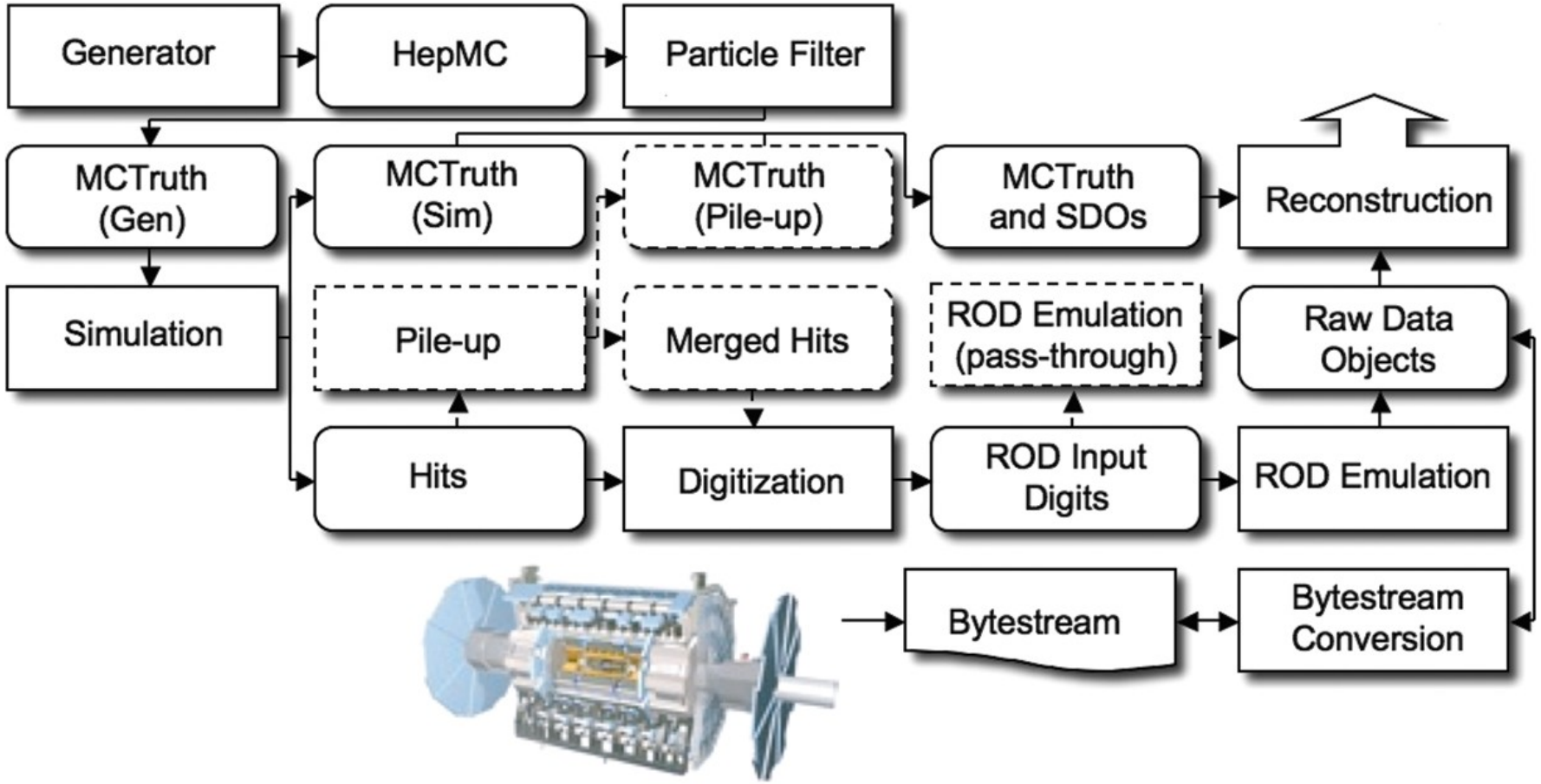
Issues with Simulating High Luminosities in ATLAS

- Simulation in ATLAS
- Simulating Multiple pp Interactions
- Background Simulation:
  - Samples
  - Bunch Structure
  - Variable Luminosity
- Simulating Current Beam Conditions
- Simulating Higher Luminosities
- PileUpTools
- Where Next?
- Summary

Simulation in ATLAS (I)

- Simulating events in ATLAS is performed as a three step process within the Athena framework:
  - Event Generation
  - Sensitive Detector Simulation
  - Detector Electronics Simulation (Digitization)



Simulation in ATLAS (II): What is Digitization?

- Digitization Input:
  - The energy deposits in sensitive detector volumes (HITS) from the (GEANT4) Simulation step.
- Digitization Process:
  - Varies by sub-detector.
  - Simulates detector responses ("Digits"). Typically voltages on and/or times of pre-amplifier outputs.
  - Digit creation is followed by a simulation of the RODs (Read Out Drivers) to produce RDOs (Raw Data Outputs).
- Digitization Output:
  - RDO pool root files.
  - SDOs (Simulated Data Objects) are also written out to the pool file. SDOs provide a link between the RDOs and the Truth information.

Simulating Multiple pp Interactions (I)

- We are now in a regime where we observe multiple p-p collisions in each filled LHC bunch-crossing and also multiple filled bunch-crossings within the sensitive time window of ATLAS.

Simulating Multiple pp Interactions (II)

- Simulation 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 using the HITS from many simulated pp interactions and digitizing them all together.
    - This includes both in-time and out-of-time pp interactions within a [-800,800] ns window.
    - In addition "cavern background" events (see next slide) are also added.

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.

Background Simulation: Samples (II)

How background events are added to the signal event depends on the sample type:

- Minimum Bias:
  - Specify the Poisson mean number of events to be added per filled-bunch crossing (mu).
  - Add a random number picked from that distribution to each filled bunch-crossing.
- Cavern Background:
  - Constant background, therefore add a constant number to each bunch-crossing.
  - Separate samples for each bunch spacing to be simulated.

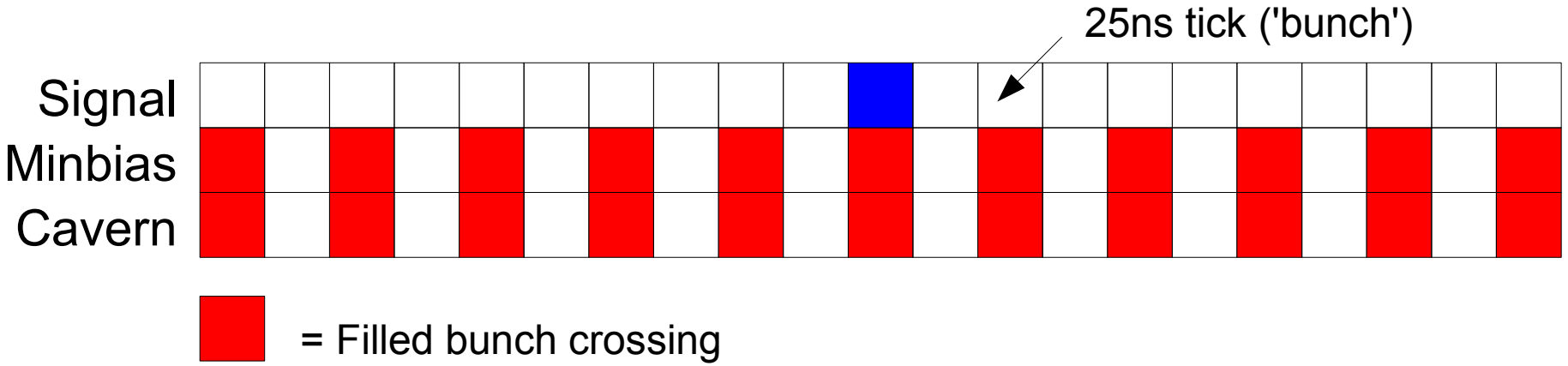
In both cases events are then offset in time depending on the particular bunch crossing they are being used for.

Background Simulation: Samples (III)

- Expensive to generate huge samples of background events.
- Background events which have been used for out-of-time pile-up can safely be re-used.
  - Create a cache of background events in memory, so they can be re-used.
  - Replace events used for in-time pile-up 100% of the time.
  - Replace events used for out-of-time pile-up ~1% of the time (tunable).
  - Save memory by only reading in/caching the parts of each event which are needed.
- Cache size dominated by the size of Truth information. Will come back to the cache later....

Background Simulation: Bunch Structure

Example of a pile-up model with fixed 50ns spacing between filled bunches:



In reality the structure of filled and empty bunch crossings can be more complicated.

- Here there are clear effects on the pile-up and hence detector response depending on where in the bunch train the triggering signal event occurs.
- ATLAS includes the modelling of more complicated bunch crossing patterns in the simulation.
- Patterns can be up to 3564 elements in length and can loop around between the beginning and end of the pattern if required.
- For each signal event, the triggering bunch crossing is picked from the filled bunch crossings in the pattern, with a probability proportional to the relative luminosities of each bunch crossing.

Background Simulation: Variable Luminosity

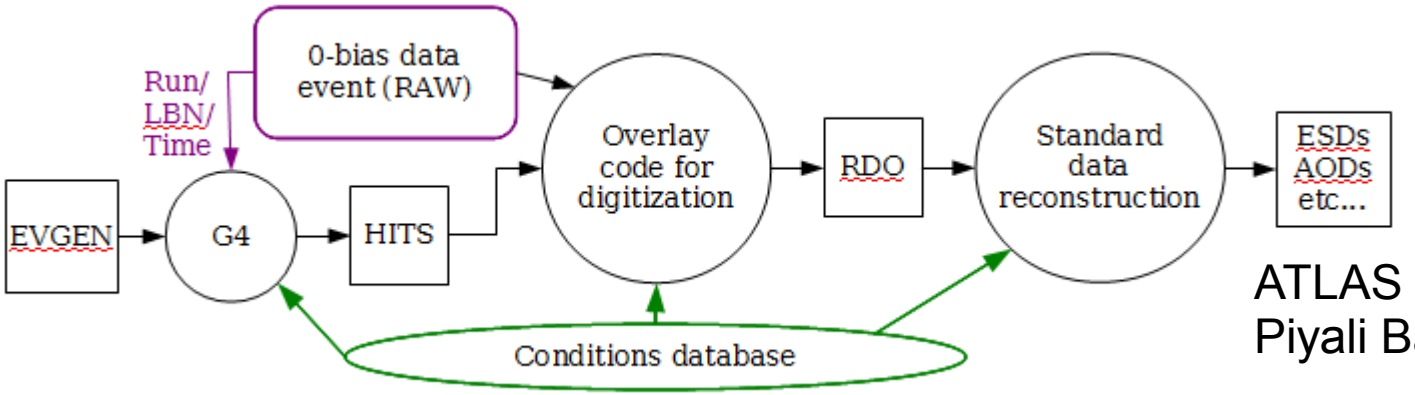
- Clearly, the bunch luminosity of the LHC varies over time.
- Both in-time and out-of-time pile-up effects are important.
- Simulating samples at a fixed mu value makes it difficult to re-weight MC to data
  - ➔ Use a range of mu values within each simulated sample.
    - The mu value used is recorded for each event.
    - This can then be to re-weight the MC sample to match a given set of data periods.

OK....

....but what about something more *data-driven*?

Simulating Current Beam Conditions

- These 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.

OK....

....but what about simulating future beam scenarios where we have no beam data?

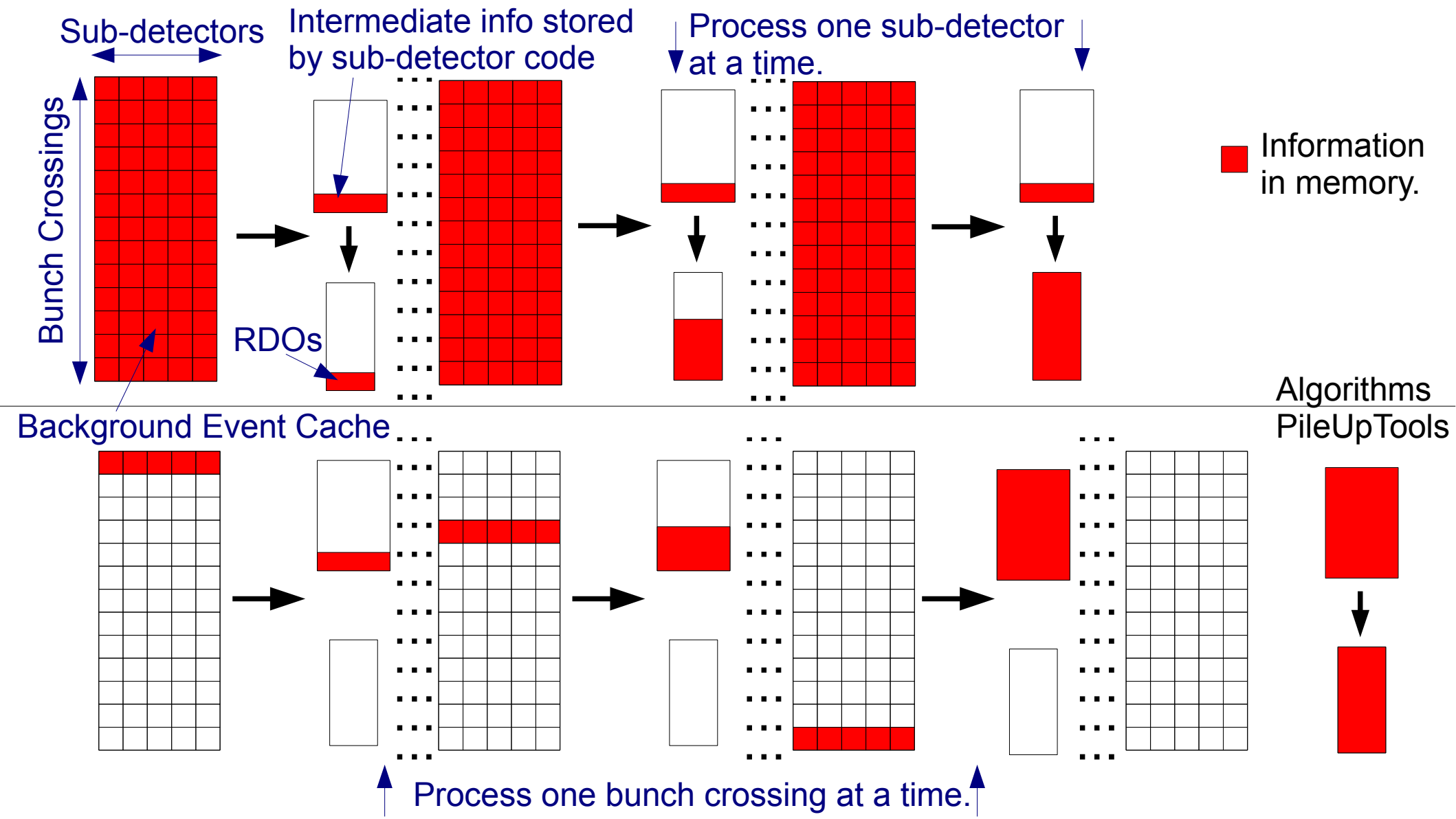
Simulating Higher Luminosities

- For luminosities at which data has not yet been taken, then event overlay is not an option...
- The previous pile-up approach has issues too...
- Consider a typical upgrade scenario:
  - 200 pp-collisions per filled bunch crossing
  - fixed 50ns spacing between filled bunch crossings
  - → ATLAS would be sensitive to 33 filled bunch-crossings
    - → 33 x ~200 x 2 = **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.
  - A single pile-up Algorithm calls an 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.

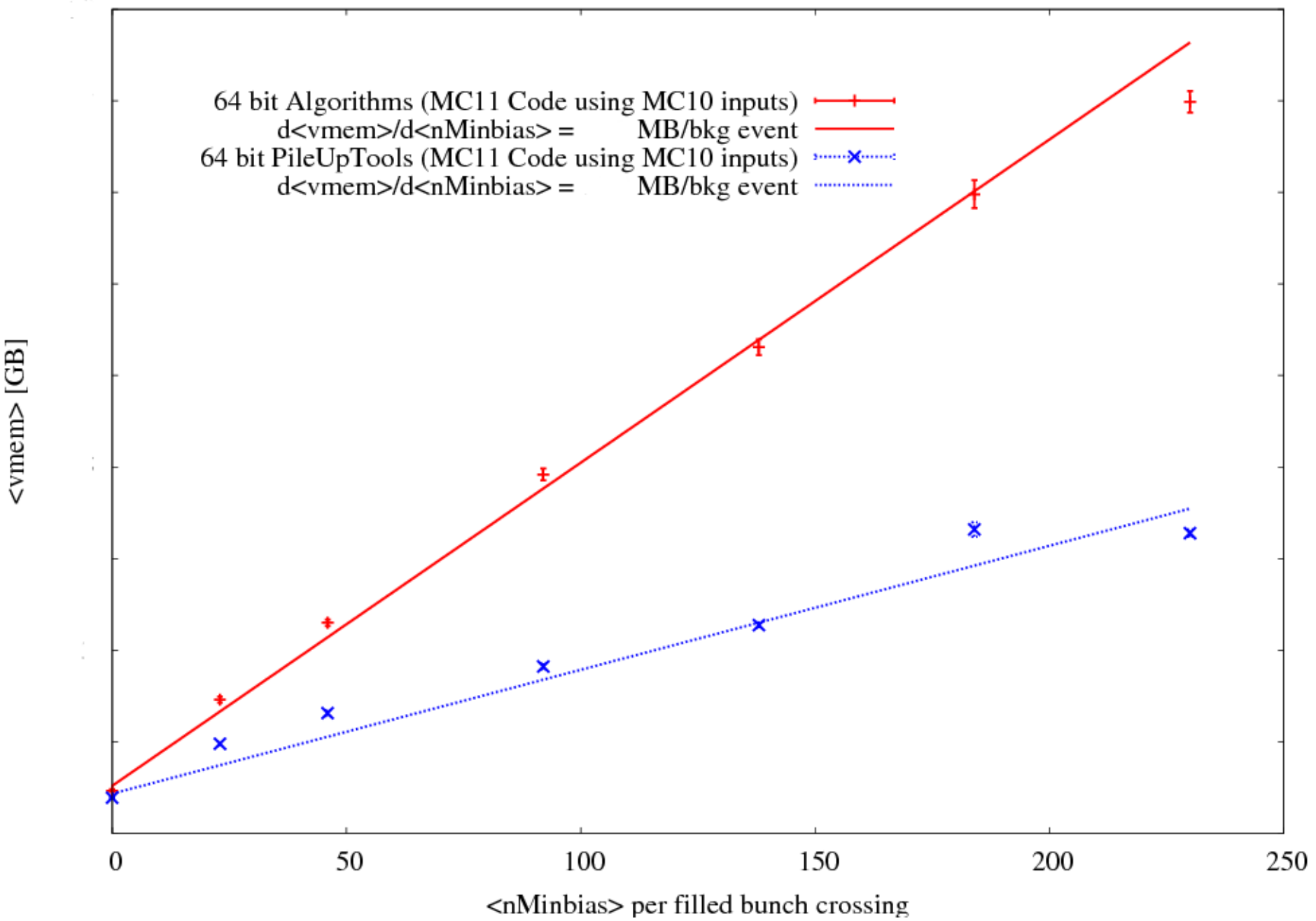
Algorithm and PileUpTools Approaches to Pile-up Digitization



PileUpTools Memory Savings (I)

- Considering again the previously discussed scenario:
  - 200 pp-collisions per filled bunch crossing
  - fixed 50ns spacing between filled bunch crossings
  - For 64-bit Athena builds:
    - Both peak and average memory requirements are reduced by over 50%!
  - For 32-bit Athena builds:
    - Memory requirements are well within the limit of addressable memory.

PileUpTools Memory Savings (II)



Where Next? (I)

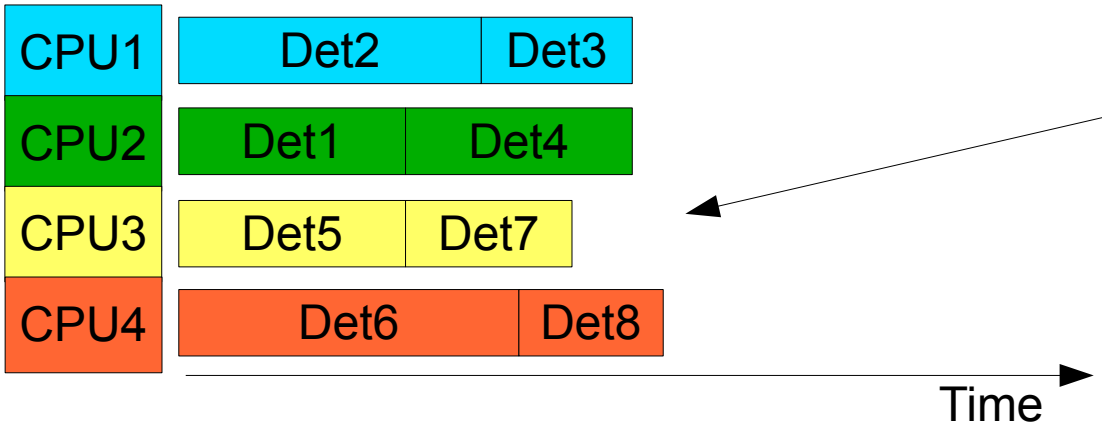
- Control the number of concurrently open background files more carefully.
  - Events are used in a random order.
    - Can end-up with many background files still open with just a single unused event remaining.
- Another alternate approach pile-up:
  - overlaying GenEvents.
  - run the ATLAS fast simulation over the combined set of GenEvents.
- Application of "AthenaMP" to pile-up jobs.
  - Is there any mileage in sharing the background cache or at least part of it between multiple threads each simulating a different signal event?

Where Next? (II)

- PileUpTools are very successful at reducing memory requirements, but do so at the cost of increased run time.
  - Need extra time for additional I/O. Look for savings elsewhere...
- Could look into parallelizing the jobs by sub-detector (or even at the detector element level)?
  - The digitization of each sub-detector is independent, so this should not be an issue.

- With a single core:

- With four cores:



Assign sub-detectors to the available cores such that the real time taken to process each bunch-crossing is minimised.

Summary

- The Simulation of multiple in-time and out-of-time pp interactions (pile-up) within ATLAS is done during the Digitization stage of the Simulation.
  - The current Digitization code can successfully do this for luminosities up to the LHC design luminosity
- For current beam conditions Event Overlay allows us to use data-driven backgrounds.
- For future higher luminosity beam conditions, the revised PileUpTools approach to pile-up simulation allows us to simulate typical SLHC beam scenarios and beyond.
- Other strategies are being considered to further reduce memory requirements and speed up pile-up simulation.