Methods and computing challenges of the realistic simulation of physics events in the presence of pile-up in the ATLAS experiment

John Chapman
On behalf of the ATLAS Collaboration

IEEE NSS,
28th October – 3rd November 2012,
Anaheim, California, USA
Overview

- Overview
  - ATLAS Experiment at the LHC
  - Simulation Flow in ATLAS
- Pile-Up Simulation in ATLAS
  - Pile-up Digitization (Original Approach)
  - Pile-up Digitization (Revised Approach)
- Overlay
  - Overlay Workflow
  - Overlay Details
  - Overlay Production
  - Overlay Future
- Summary
Z→mumu event with 23 reconstructed vertices.
ATLAS Experiment at the LHC

• Definitions:
  – The two beams in the LHC are divided up into 25 ns buckets, some of which will contain a bunch of protons. When the beams cross within ATLAS, if both buckets contain protons, this is known as a filled bunch-crossing.
  – \(\mu\) is defined as the mean number of interactions per bunch-crossing averaged over all filled bunch-crossings seen by ATLAS. \(\mu\) falls during the course of a run.
  – Bunch spacing is the time between filled bunch-crossings.
  – A bunch train is a group of filled bunches separated by a common bunch spacing.
ATLAS Experiment at the LHC

• We are now in a regime where we observe multiple p-p interactions in each filled LHC bunch-crossing and multiple filled bunch-crossings within the [-800,800] ns sensitive time window of the ATLAS detector.
• The effects of these additional interactions must be included in any simulation of the detector response.
  
  – **Current**: Beam energy 8 TeV, Peak \(<\mu>=40\), bunch spacing in train= 50 ns, maximum of 33 filled bunch-crossings within sensitive window of ATLAS.
  
  – **Design**: Beam energy 14 TeV, Peak \(<\mu>=27\), bunch spacing in train = 25 ns, maximum of 65 filled bunch-crossings within sensitive window of ATLAS.
  
  – **Future scenario relevant for ATLAS upgrade**: Beam energy 14 TeV, Peak \(<\mu>=140\), bunch spacing in train = 25 ns, maximum of 65 filled bunch-crossings within sensitive window of ATLAS.
Simulation Flow in ATLAS

- Generator
- HepMC
- Particle Filter

MCTruth (Gen)
MCTruth (Sim)
MCTruth (Pile-up)

Simulation
Pile-up
Hits
Merged Hits
Digitization

MCTruth and SDOs
ROD Emulation (pass-through)
ROD Input Digits

Reconstruction
Raw Data Objects
ROD Emulation
Bytestream Conversion
Bytestream

15/10/2012
John Chapman, University of Cambridge, ATLAS Collaboration
Simulation Flow in ATLAS

Conversion of generated particle four-vectors into energy deposits in detector material. Performed using Geant4 9.4.

15/10/2012
John Chapman, University of Cambridge, ATLAS Collaboration
Simulation Flow in ATLAS

Simulation of Detector Electronics and Readout. Performed using internal ATLAS code.
Simulation Flow in ATLAS

Pile-up is ATLAS terminology for the multiple pp-interaction with the sensitive time window of the detector.
Pile-Up Simulation (Original Approach)

- Reduce required size of simulated background-dataset by re-using background events.
- Job maintains a cache of background events in memory. Size > requirement for 1 signal event.
- Pick background events at random from this cache, so they are not always used in the same bunch-crossings.
- Process all in-time and out-of-time events at once.
- Background information is only retrieved/cached for events used within sub-detector specific time windows - e.g. [-25,25]ns for the Pixel Detector, but [-800,800]ns for the Muon Drift Tubes.
- Replace events in the cache at a configurable rate.
- \(<\mu>\) value simulated varies for each signal event to reflect variations seen in data.
- **Issues:**
  - Memory requirements increase rapidly with \(<\mu>\).
  - High energy tail of background can dominate triggers.
Pile-Up Simulation (Revised Approach)

- Split background dataset up according to energy. Increase statistics for high energy portion to reduce re-use (can be a problem for physics analyses).

- Reduce memory requirements by:
  - **Removing background cache**: Read-in, use, then discard background events. (Same files may still be read in multiple times by a job, order of event usage within a file is still random.)
  - **Process one bunch-crossing at a time**: Background information can be discarded sooner.

- Pick background events at random from this cache, so they are not always used in the same bunch-crossings.

- **Issues**:
  - Heavy I/O load.

![Leading Jet $p_T$ vs Jet $p_T$](chart.png)

<table>
<thead>
<tr>
<th>Jet $p_T$ [GeV]</th>
<th>Low $p_T$ Sample</th>
<th>High $p_T$ Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jets / pb / GeV</td>
<td>( \times 10^{-8} )</td>
<td>( \times 10^{-8} )</td>
</tr>
</tbody>
</table>
OK....
....but what about something more data-driven?
Overlay Workflow

Generator → HepMC → Particle Filter

MC Truth (Gen) → MC Truth (Sim) → MC Truth And SDOs

Simulation → Digitization → ROD Input Digits

Run No. / LBN / TimeStamp

hits → ROD Emulation (pass through)

Reconstruction → Raw Data Objects → Overlay

Bytestream → Bytestream Conversion → Raw Data Objects
“Zero-bias” triggers are read out one LHC turn later than a triggering BC. Random trigger, but rate proportional to luminosity. The rate at which we can readout zero-bias events is limited by trigger/DAQ considerations.
Read out un-sparsified calorimeter data for this trigger only. Needed to merge simulated signal with data overlay correctly in digitization step. Consequently, RAW event size ~50% bigger than standard readout.
Use information from Zero-bias event to be overlaid to ensure that simulated ‘signal’ event has conditions (beam-spot, magnetic field and detector alignment) consistent with overlaid event.
Overlay Workflow

RDO level MC events are “overlaid” on Zero-bias triggered Data events.
Overlay Details (I)

- Overlay successfully used at previous experiments (e.g. D0, BaBar).
- Procedure used for ATLAS has a number of improvements:
  - **Very realistic use of conditions** - i.e. beam-spot measured every ~5 min, sampled according to the time of the event. Use data alignments and magnetic field at time of data event for simulation and reconstruction.
  - **Accurate** (offline) sampling of zero-bias events according to integrated luminosity in each luminosity block from luminosity DB
    - exact average luminosity profile!
  - **Accurate** (online) sampling of zero-bias events according to instantaneous luminosity in each pp bunch, via +1 turn EM14 trigger.
    - exact per-bunch luminosity profile!
  - **Grouping of zero-bias events** (offline) into datasets of (50k) randomly sampled (luminosity weighted) events.
    - Full data period luminosity distribution used, even for small (50k) simulated (sub-)samples.
    - c.f. Standard approach of taking N random files of zero-bias chosen from the period. Luminosity distribution correct on average, but subject to fluctuations depending on the exact files used.
    - By distributing these zero-bias datasets to different grid sites, we automatically balance the production of large samples across the grid, and minimise storage requirements, but retain correct luminosity distribution in sub-samples.
Overlay Details (II)

Pros:

- p-p overlay will get the details of the pile-up correct.
  - Analyses sensitive to pile-up modelling should benefit.
  - Should also help detector performance groups where pile-up modelling is a problem. E.g. for Jet calibration at high pile-up.
- Conditions will track data more closely.
- Lower job memory requirements.

Cons:

- Can not simulate imaginary situations (e.g. Future high luminosity, different energy events, or upgrades to the detector).
- Less accurate when combining overlapping background and signal on the same channel for some sub-detectors (e.g. silicon).
  - RDO contains less information than background HITS.
- Probably can not have as many overlay events as simulated events.
  - Some re-use will need to happen, need to avoid bias from this.
- Don't have the background MC truth information
  - It's data!
Real data $Z \rightarrow \mu \mu$ event in a central Pb-Pb collision. Simulating this with overlay is much more efficient (and gets the underlying event more correct!)
Overlay Production

• Work has been done integrating overlay workflow into the production system
  – Needs access to large (>100TB) Zero-bias RAW datasets
  – Smaller Zero-bias datasets need to be prepared before use:
    • currently done ‘by hand’.
    • Must now automate the merging of O(10k) files with ~5 events each into O(500) files with 100 events each! (Files must first be ordered by LumiBlock, to minimise alignments and conditions changes during simulation.)
  – Overlay simulation has been integrated into the production system.
    • Still need to automate sampling from the 50k zero-bias datasets and use different ones for each MC sample in some optimal way to make use of all zero-bias stats.

• For large scale production, workflow in production system may need to be optimized.
  – Overlay Simulation needs to access data conditions, which change more frequently with time than the standard Monte Carlo conditions.
    • Each 100 event overlay job has events from a few different runs, so requires the loading of O(200MB) of database information (similarly for the reconstruction job).
    • Requires sophisticated caching at local sites, and optimized flow of database info:
      – Frontier squid project: Oracle database access is cached via a Frontier server at each Tier1 and a squid cache at the local site.
Summary

• Must account for the effects of multiple proton-proton collisions within the sensitive time window of the ATLAS detector.
  – Effects will increase with increased luminosity in the near future.
• ATLAS has two methods of simulating the effect of these “pile-up” interactions.
• Pile-Up Digitization: simulate single p-p collisions separately using Geant4, then combining them before they are processed by the Digitization step.
  – Memory usage is minimised by only storing background events being used for one LHC bunch-crossing in memory at a time.
  – Minimise the effect of reusing background events by boosting the statistics of the high pT tail of the background events, so they are re-used less often.
  – Can be used for future scenarios.
• Overlay: detector backgrounds are sampled from raw data using a special zero-bias trigger. Simulated physics events are overlaid at the raw data level.
  – This gives a very realistic simulation of the detector response to physics events.
  – Runtime proportional the number of events in the sample and memory requirements proportional to the size of a single event, with a small overhead.
  – New approach is more robust than the standard approach to overlay.
• Many of the computational and production issues associated with simulation on the grid using Overlay have already been addressed.
  – Database access requirements are being addressed.
  – Still some work to do on automation of initial background dataset production.
The transverse isolation energy is computed as the sum of electromagnetic and hadronic calorimeter cell energies within a cone of radius 0.4 in the eta-phi space around electromagnetic objects (electron and photon candidates). The energy from the core of the cone in the electromagnetic calorimeter (5x7 cells around the object barycenter) is excluded from the sum. Corrections to the isolation energy from the electromagnetic object out-of-core energy leakage and from the underlying event and in-time pileup (interactions in the same bunch crossing) energy are applied.
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 (i.e. cavern background) are dropped from the minimum bias sample simulation.
    - For current experimental conditions this background can be neglected, but in the future this will not be the case.
    - This sample will be 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:

- **Signal**
- **Minbias**
- **Cavern**

= Filled bunch crossing

25ns tick ('bunch')

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.