The Fast Simulation Chain for ATLAS

A Basalaev\textsuperscript{1} and Z Marshall\textsuperscript{2}, on behalf of the ATLAS Collaboration

\textsuperscript{1}National Research Centre “Kurchatov Institute” B.P.Konstantinov Petersburg Nuclear Physics Institute, St. Petersburg, Russia
\textsuperscript{2}Physics Division, Lawrence Berkeley National Laboratory, Berkeley CA, United States of America

E-mail: artem.basalaev@cern.ch

Abstract. In order to generate the huge number of Monte Carlo events that will be required by the ATLAS experiment over the next several runs, a very fast simulation is critical. Fast detector simulation alone, however, is insufficient: with very high numbers of simultaneous proton-proton collisions expected in Run 3 and beyond, the digitization (detector response emulation) and event reconstruction time quickly become comparable to the time required for detector simulation. The ATLAS Fast Chain simulation has been developed to solve this problem. Modules are implemented for fast simulation, fast digitization, and fast track reconstruction. The application is sufficiently fast—several orders of magnitude faster than the standard simulation—that the simultaneous proton-proton collisions can be generated during the simulation job, so Pythia8 also runs concurrently with the rest of the algorithms. The Fast Chain has been built to be extremely modular and flexible, so that each sample can be custom-tailored to match the resource and modeling accuracy needs of an analysis. It is ideally suited for analysis templating, systematic uncertainty evaluation, signal parameter space scans, and simulation with alternative detector configurations (e.g. upgrade), among other applications.

1. Introduction

Monte-Carlo simulation plays a key role in ATLAS experiment [1]—it is the only tool available to predict the response of the complex detector, and therefore to be able to compare experimental results with theoretical predictions. In addition, it allows the study of new detector layouts and planning for the upgrade. For data analyses, it is essential to have vast samples of Monte-Carlo events. Currently, over 14 billion Monte-Carlo events have been produced in the “MC15 campaign”\textsuperscript{1} alone. Computing resources of ATLAS, however, are limited and Monte-Carlo production is the single biggest resource consumer (see figure 1); therefore efforts are focused on reducing the required number of events and CPU time consumption by Monte-Carlo production.

The vast majority of this time is spent in simulation of particle propagation through the detector, or simply “simulation” [2]. The Geant4 package is traditionally used in ATLAS for the simulation step, it has proven to be reliable and accurate for a wide range of particle types, energies and detector materials. The most time-consuming part of the simulation step is calorimeter simulation, because particles are stopped there followed by generation of secondary particle showers. The first step in reducing the time of the simulation step is therefore improving calorimeter simulation. To address this issue, the ATLAS Fast Calorimeter Simulation (FastCaloSim) package [3] was developed. It uses a parameterisation based on the Geant4

\textsuperscript{1} Monte-Carlo simulation to use with data produced in 2015 and 2016.
Figure 1. Relative CPU grid usage (left) and relative grid disk (right) averaged over all ATLAS grid sites for the year 2012.

Simulation of single particles in a fine grid of particle energies and directions, which allows the bypassing of the shower generation step, directly obtaining the energy released in calorimeter cells. FastCaloSim in conjunction with Geant4 (referred to as ATLFASTII, see figure 2) allowed to reduce simulation time by a factor of 10 or more. This, however, comes at a cost of poor modelling of jet substructure variables [3]. Thus, only some analyses can use ATLFASTII, and the “full simulation” (with Geant4) share is still about a half of all simulation.

Figure 2. ATLFASTII simulation configuration: FastCaloSim for calorimener, Geant4 for the rest of detectors.

With increased average number of simultaneous proton-proton collisions ($\mu$) expected in Run 3 and beyond, the digitization (detector response emulation) and event reconstruction time quickly become comparable to the time required for detector simulation. The latter, in fact, even starting to dominate, because of the combinatorial nature of the track finding problem, which yields an exponential increase in reconstruction time with respect to $\mu$. Sample sizes grow considerably with $\mu$ as well.

The problems described above require complex solutions. Not only is improvement of resource consumption needed on every step of the simulation and reconstruction chain (“full chain”), but the configuration of the chain has to be flexible, tailored for the needs of specific analyses, allowing users to plug in more precise tools where the accuracy is critical, saving resources in other places. This idea is underlying the Fast Chain project.

2. Fast Components

In recent years, considerable efforts were focused on developing faster alternatives for all components of the full chain. Most of them are planned to be included in Fast Chain configurations and some of them are developed specifically for use in Fast Chain. Basic description of these fast tools is given below.
2.1. FastCaloSim

The FastCaloSim package provides a parameterised simulation of the particle energy response and of the energy distribution in the ATLAS calorimeter, reducing the calorimeter simulation time from several minutes to a few seconds per event. The fast simulation parameterisation reproduces the longitudinal shower properties, including fluctuations and correlations, but only average lateral shower properties and uncorrelated lateral energy fluctuations. FastCaloSim is actively used for Monte-Carlo sample production in ATLAS as a part of ATLFASTII configuration.

Currently work is ongoing on a new version of FastCaloSim, where machine learning techniques are used, such as Principal Component Analysis and TMVA neural network regression analysis [4]. The new version promises improvements in jet substructure modelling, along with improvements in speed, precision, and memory management.

2.2. Fatras

The Fast ATLAS Tracking Simulation (Fatras) [2] produces a Monte Carlo simulation based on the software modules and the simplified geometry used by the standard ATLAS track reconstruction algorithms. Fatras simplifies the layout description while at the same time guaranteeing a fair amount of accuracy in the simulation. Instead of volumes, the detector is described by thin layers on which the properties of detector volumes’ material are projected (see figure 3). During the propagation through the detector, interactions with the detector material are performed according to different particles types, modelled by fast algorithms. The extrapolation tools sample the material effects from parametrised functions taking ionization, bremsstrahlung photon emission, photon conversion, positron annihilation, multiple scattering effects and hadronic interactions into account.

First estimates show that Fatras in conjunction with FastCaloSim (known as the ATLFASTIIIF configuration) could provide a reduction of simulation time of up to a factor of 100 or more compared to full simulation with Geant4 [2]. Fatras is currently undergoing physics validation.

Figure 3. Visualization of the simplified geometry used by the standard ATLAS track reconstruction and Fatras, derived from photon conversion vertices [2].
2.3. *Pileup with Pythia8 on the fly*

The simulation of pileup effects in ATLAS is done by overlaying presimulated minimum bias events (pileup) over each signal (hard scatter) event. This allows the reuse of pileup collections simulated fully with Geant4. However, the number of presimulated events required grows significantly with $\mu$, and reading input pileup collections slows down the simulation. Besides, some correlations induced by reusing pileup events are unavoidable. The concept of on the fly pileup generation with Pythia8 and simulation with FastCaloSim was developed for Fast Chain to avoid these problems and combine the pileup generation, simulation, and digitization steps without producing and reading any intermediate output files.

Currently only in-time pileup generation is done with Pythia8 in Fast Chain. The first estimates have shown that on the fly in-time pileup generation with $\mu=10$ takes just a few percent of the fast simulation (FastCaloSim + Fatras) job time (figure 4). Therefore out-of-time pileup can be generated as well for each HS event without reusing any pileup events, fully avoiding the problem of induced correlations. The LAr calorimeter readout function can be roughly approximated with just two weights [5]. Thus, as a first step it is possible to generate pileup for just two bunch crossings per HS event.

![Figure 4. Relative timing of fast components in Fast Chain job.](image)

2.4. *Fast Digitization*

The inner detector (ID) is the most time-consuming part (about 50%) of the digitization due to very high density of hits there. Therefore, faster solutions are needed for the Transition Radiation Tracker (TRT), SemiConductor Tracker (SCT), and Pixel detectors. Fast TRT Digitization, Fast SCT Digitization, and Fast Pixel Digitization packages were developed as an alternative to standard “full” digitization algorithms [6].

For the SCT and Pixel detectors, the charge deposition is estimated for each readout channel by projecting the simulated track length on the readout surface. This is corrected for the Lorentz angle drift due to the ATLAS magnetic field. The charge deposit positions and path length are furthermore smeared to account for multiple scattering of the drifting charge carriers (presented in figure 5). In this approach, clusters are formed directly using track information, thus eliminating the need for cluster finding algorithms. In this aspect fast digitization is different from full digitization, which provides output Raw Data Objects (RDOs) with the cluster finding step performed by reconstruction algorithms.

In the TRT, the response is emulated from simulated hits by evaluating the radius of closest approach (figure 6). The uncertainty of the measurement is also taken into account by creating a smeared hit position. In addition the transition radiation response is parametrized to still allow particle identification [6].

Fast TRT digitization has already passed physics validation; for SCT and Pixel fast digitization the validation is ongoing.

---

2 In-time pileup consists of proton-proton collisions happening in the same 25 ns window. But because the readout of some subdetectors can be affected by previous collisions, it is necessary to consider effects of out-of-time pileup. Notably, for Liquid Argon (LAr) calorimeter this time can be up to 500 ns [1].
2.5. Truth-assisted Reconstruction

Track reconstruction finds particles trajectories from digitized hits. The combinatorial nature of the problem, which rapidly grows in complexity with high pileup, yields exponential increase in reconstruction time. The main consumer is again the ID reconstruction. This motivates the switch to use MC generator information (truth information), in approach called truth-assisted or truth-seeded track reconstruction, which emulates the effects of the default algorithms. The truth-seeded approach is realized by skipping the time consuming pattern recognition, track seeding and ambiguity treatment completely and instead using the MC truth information directly to assign the correct hits to each track and then fit the particles’ trajectories. The tracks are manipulated by changing the hit content and efficiency, as well as by applying similar selection criteria as in normal reconstruction, in order to mimic the effect of skipped steps (see figure 7). In order to preserve the effects of the skipped steps for the signal event, the most feasible configuration for this approach is to reconstruct only tracks from pileup interactions using truth association. Truth-assisted reconstruction is currently undergoing physics validation.

Figure 5. Schematic view of fast digitization approach: track is projected on sensitive surface, corrected for Lorentz shift, and smeared [6].

Figure 6. Drift radius $r_D$ is estimated from the radius of closest approach, calculated considering the uncertainty on the measurement from simulation [6].
Figure 7. Truth track creation in the ID. All hits from the simulation step are fed into the digitization. The resulting PRD (Prepared Raw Data) object is used, along with PRD MultiTruthCollection, which connects PRD contents with truth information. Tracks are built from this input, manipulators and selectors are applied, and finally tracks are refit to get the output track collection [7].

3. Fast Chain configurations

Multiple configurations are possible and being developed combining fast and standard components in a single Fast Chain transform. The most flexibility is possible in the simulation step, where the Integrated Simulation Framework [2] allows for a complex sets of rules defining simulation strategies to be created, virtually by any desired parameter, e.g. using full simulation around the cones of particles of interest with fast simulation applied everywhere else.

As a first working Fast Chain configuration, a split between hard scatter and pileup is developed, with Pythia8 on the fly + fast simulation (FastCaloSim + Fatras), fast digitization and truth-assisted reconstruction for pileup, and full simulation (Geant4), standard digitization and standard reconstruction for the hard scatter. Fast and full components for such a “split” hard scatter/pileup configuration are easily swapped independently in the simulation and digitization steps. There’s also a possibility to extend such configuration using a more sophisticated “simulation mix” in the simulation step with the help of ISF. One possible Fast Chain configuration is presented on figure 8. Eventually most I/O operations can be avoided as well by running all steps of the chain in single Fast Chain transform, including the final step—DxAOD [8] production—a format tailored for analyses.

Figure 8. One possible Fast Chain configuration scheme. Hard scatter (HS) and pileup (PU) are kept split throughout the chain, with fast components used for PU and standard components used for HS. On simulation step the split can be more sophisticated, with components selection depending on particle types, parameters, cones around particles of interest, etc.
4. Conclusion
The Fast Chain project is aiming to improve the resource consumption of ATLAS MC production with a minimal impact on the accuracy of simulation, using various fast components on every step of the simulation chain and combining them with standard, more precise components in a flexible way. Currently fast components are undergoing physics validation, and some of them already passed it. Fast Chain configurations incorporating these components are being developed in parallel and then will undergo independent validation. New configurations can be created and tuned, tailored for specific analysis needs.

References