Journal of Physics: Conference Series **219** (2010) 032016 doi:10.1088/1742-6596/219/3/032016

Validation and performance studies for the ATLAS simulation

Zachary Marshall for the ATLAS Simulation Group

California Institute of Technology, Physics Department, 103-33, Pasadena, CA 91125, United States of America and Columbia University, Nevis Laboratory, 136 So. Broadway, Irvington, NY 10533, United States of America

E-mail: zmarshal@caltech.edu

Abstract. We present the validation of the ATLAS simulation software project. Software development is controlled by nightly builds and several levels of automatic tests to ensure stability. Software performance validation, including CPU time, memory, and disk space required per event, is monitored for all software releases. Several different physics processes are checked to thoroughly test all aspects of the detector simulation. The robustness of the simulation software is demonstrated by the production of 500 million events on the World-wide LHC Computing Grid in the last year.

1. Introduction

ATLAS [1], one of the general-purpose experiments at the Large Hadron Collider at CERN [2], began operation in 2008. The detector was designed to collect data from 14 TeV proton-proton collisions as well as 5.5 TeV per nucleon pair heavy ion (Pb-Pb) collisions. During proton-proton collisions at design luminosity of 10^{34} cm^{-2}s^{-1}, beam bunches will cross every 25 ns and provide on average 23 collisions per bunch crossing. ATLAS has been designed to record 200 bunch crossings per second, keeping only the most interesting interactions for physics studies.

In order to study detector response and the effectiveness of proposed data analysis strategies, a detailed simulation has been implemented that carries events from event generation through to output in a format identical to that of the actual detector. The software is integrated into the Athena software framework [3] and uses Python as a scripting language for the runtime configuration of jobs. Libraries are loaded on demand, keeping each job as light as possible in memory.

The simulation data chain is divided into three steps for production, though they may be combined into a single job. First, the event is generated using Pythia, Herwig, or one of several other available event generators [4]. Second, the Geant4 toolkit [5, 6] is used to propagate the generated particles through the detector and simulate physics interactions within it. Third, the energy deposited in the sensitive regions of the detector is converted into voltages, currents, and times for comparison to the readout of the real ATLAS detector [7]. The output of the simulation chain can be presented in an object-based format ("RDO") or in a format identical to the output of the ATLAS data acquisition system ("bytestream"). Thus, both the simulated and real data from the detector can be run through the same ATLAS trigger and reconstruction packages.

The data flow of the simulation software chain can be seen in figure 1. Algorithms to be run are represented by square-cornered boxes, and persistent data objects are placed in round-cornered boxes. The optional steps required for pile-up or event overlay are shown with a dashed outline.

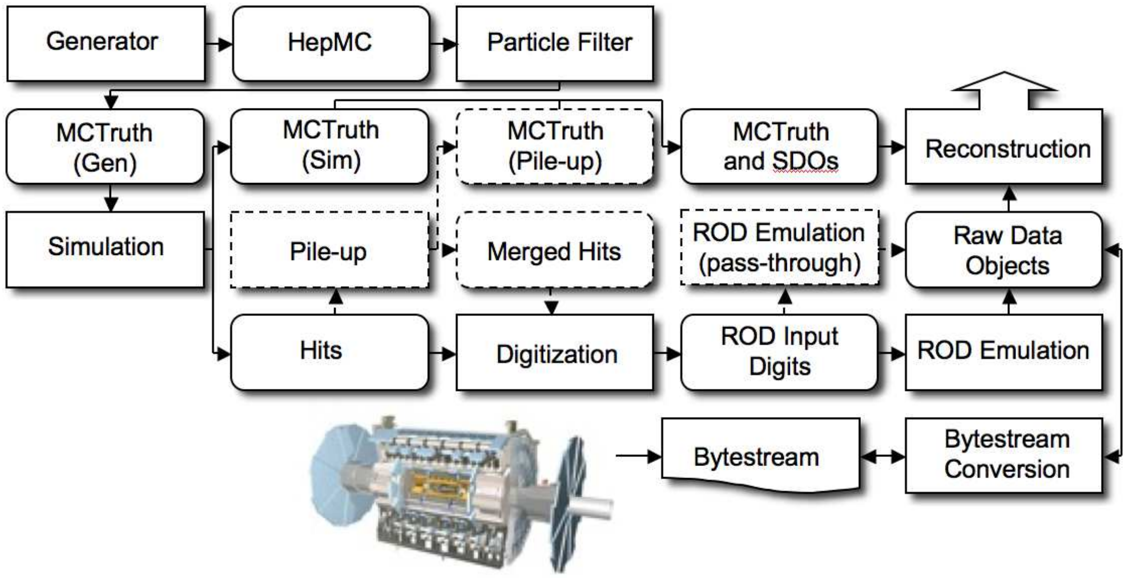

Figure 1. The general layout of the ATLAS simulation. The flow can be seen from Monte Carlo generators through output identical to the data read out from the detector. Generators produce output in HepMC format. Simulation is run after some filtering, and digitization is afterwards run to simulate the detector read-out drivers (RODs). At each stage, some Monte Carlo "truth" (MCTruth) information is kept, including the output of the generator, parameters of event generation, and certain interactions in Geant4 (e.g. decays in the inner detector). The mapping of this truth information to hits is stored in simulated data objects (SDOs).

The simulation chain, divided in this way, uses resources more effectively and simplifies software validation. Event generation jobs, typically quick and with small output files, can be run for several thousands of events at a time. By storing the output rather than regenerating it each time, it becomes possible to run identical events through different versions of the simulation software or with different detector configurations. The simulation step is particularly slow, and can take several minutes per event. In production, simulation jobs are therefore divided into 25-50 events at a time; only a few events may be completed in a single heavy ion collision simulation job. Digitization jobs are generally configured to run ~ 1000 events in order to ease file handling. As much as is possible during this chain, options that were set in a previous step are stored in the output file and used to automatically configure the subsequent step. In that way, for example, the detector geometry used for a simulation job and subsequent digitization job are consistent.

Large-scale production is done on the World-wide LHC Computing Grid (Grid) [8]. A single task on the Grid is split into many jobs depending on the content and complexity of the task. A job can be completed by a single CPU within the maximum allowed time for a job on the Grid (typically 2-3 days). In the case of a full chain of jobs being run (generation, simulation, and digitization), each subsequent step is automatically held in the queue until the required data is available from the previous step.

2. Validation

Validation of the ATLAS simulation chain is done in two distinct phases. First, the software performance must be assessed. Then, the physics performance must be tested and compared to available data. The first step includes testing robustness, measuring software performance, and testing basic functionality. The second step includes comparison to test beam, cosmic data, and physics results obtained from previous simulation productions. In this section the infrastructure for each stage of validation is described.

In a large, international collaboration such as ATLAS, having a fresh software build every night is crucial for coordinating software development. Each nightly build is run through a rigorous test cycle, and as a release deadline approaches the test results are increasingly scrutinized to evaluate stability and performance. Thanks to the evaluation prior to release, generally only rare bugs appear in production for the first time. The automatic testing infrastructure also allows evaluation of many different versions of the Athena software. Separate bug-fixing and development branches are employed, for example, and major code changes take place in separate branches until they are sufficiently stable to be merged into the main branch. Each version of the software comes in several flavors for different system architectures, operating systems, compilers, and so on. The simulation currently uses 32-bit builds with gcc 3.4.6 [9] on CERN's Scientific Linux 4 [10], with migrations to gcc 4.3 and Scientific Linux 5 expected before data taking in 2009. External dependencies include CLHEP 1.9.4.2 and LCG 56A.

2.1. Automated Testing

The software performance of the simulation is monitored in three types of automated tests [11, 12]: ATLAS Testing Nightly (ATN) tests, RunTimeTester (RTT) tests, and Full Chain Tests (FCT). ATN tests are run every night on every software build and are basic functionality tests. RTT tests are run on a subset of builds and include 50 simulation tests to ensure functionality and, in some cases, consistent results. FCT tests are run on only a few builds each night and test the entire software chain in a production-like environment. Releases are required to pass a minimal number of milestones before being declared ready for production.

FCT tests are run daily on a small set of jobs. The aim of the FCT is to verify the readiness of a software release candidate for Grid production. The FCT runs jobs that test the functionalities of generation, the different flavors of fast and full simulation, digitization, bytestream conversion, and reconstruction of Standard Model processes, black holes, and heavy ion collisions. In the case of Standard Model processes and black holes, a full chain[1] of jobs are run per release, with hard scattering events that stress the software. If successful, the output from each day's run is saved for use in the next step of the test the following day (e.g. Monday's generation provides input for Tuesday's simulation, which provides input for Wednesday's digitization). The typical number of events processed (50) is limited by the CPU requirements for the full simulation.

These tests only check for the success or failure of the job, the number of events in the output file, and unknown error messages in the log file. If any of these checks fail, the release candidate is rejected, and an additional iteration of bug fixing is undertaken. Only once a release succeeds in all FCT tests is it distributed to the Grid.

2.2. Software Performance Monitoring

Besides automated testing, simulation-specific software performance tests are run by hand using a variety of physics processes in each stable release. CPU time, memory consumption, and output file size are measured. Single muons, electrons, and charged pions are used at several transverse momenta (p_T), as well as di-jets in bins of leading parton p_T, a supersymmetric

[1] A single chain of jobs runs all steps from event generation through reconstruction, sequentially, using the output of one step as the input of the next.

benchmark point[2], minimum bias, Higgs boson decaying to four leptons, Z→ e^+e^-, Z→ $\mu^+\mu^-$, and Z→ $\tau^+\tau^-$ events. At least 200 events are simulated for each process, more for simpler processes, in order to ensure that the timing results have stabilized. The same input files have been used for these measurements for the last several years, allowing a fair and unobscured evaluation of the performance of each release. Because event generation is fast (hundredths of a second per event) and requires little memory (under 500 MB), its performance is not monitored from release to release.

Simulation of the full detector requires typically $\sim$ 850 MB of memory (VSIZE) and includes loading of almost 400 libraries into memory. The memory is independent of the number of events simulated and only varies by $\sim$3% in different physics processes. Memory consumed during simulation is also broken down into its three key components: GeoModel, the ATLAS-side detector geometry, typically about $\sim$ 100 MB; G4Atlas, the purely Geant4 component of the memory, typically about $\sim$ 350 MB; and load modules, the remaining algorithms and services loaded during the job, typically about $\sim$ 350 MB. Significant changes in any of these can indicate the proper source of a change in memory. The different components are monitored by comparing snapshots taken before and after the various initialization steps. Although more than 2 GB of memory may be reserved for a Grid job, by keeping the memory requirements of a typical simulation job under 1 GB, more machines can be used.

Of major concern is any increase in memory ("leaks") during the event loop once all libraries have been loaded and setup is complete. Some increase due to caching is expected during the processing of the first few events. However, if the memory required by the application continues to grow beyond the system limits, memory corruption and memory pressure can result in serious problems. The memory required by the ATLAS simulation has been found to increase by less than 0.25 MB per event under normal circumstances. The increases are not steady, but come in large ($\sim$ 10 MB) and sporadic jumps due to the resizing of large vectors. Nonetheless, any 50 event simulation job still consumes well under 1 GB of memory.

In the digitization, memory consumption can be a serious concern during jobs with many overlaid events. Table 1 shows how resource consumption during digitization of 50 $t\bar{t}$ events scales with pile-up luminosity[3]. The memory required increases beyond the physical memory of the machine for luminosity of 10^{34} cm^{-2} s^{-1}, leading to the non-linear increase in CPU time. The memory allocated per event is provided as an estimate of the change in memory over the course of a single event.

Table 1. Digitization computing resources for 50 $t\bar{t}$ events as they scale with luminosity. CPU times are normalized to the time required by a no pile-up job. Cavern background (i.e. photons and neutrons in the cavern due to radiation from the LHC and previous collisions) was overlaid during these jobs. Beam gas and beam halo events were ignored.

Resource	**No Pile-Up**	10^{33} cm^{-2} s^{-1}	3.5×10^{33} cm^{-2} s^{-1}	10^{34} cm^{-2} s^{-1}
CPU Time Factor	1.0	2.3	5.8	160
Memory Leak [kB/event]	10	270	800	2100
Virtual Memory [MB]	770	1000	1300	2000
Allocated Memory [MB/event]	12	21	40	985

The CPU time consumed by event generation, full simulation, digitization, and reconstruction

[2] ATLAS mSUGRA benchmark point SU3: m_0 =100 GeV, $m_{1/2}$=300 GeV, $A_0 = -300$, $\tan\beta = 6$, and $\mu > 0$. Events were generated with Herwig using output from Isasugra.

[3] Many of the leaks reported here have since been found and repaired. It is now possible to run high-luminosity pile-up on the Grid [7].

for various types of events is shown in tables 2 and 3 and is typically several minutes per hard scattering event. All times are normalized[4] to kSI2K seconds [14]. For the purposes of performance measurements, log file output was suppressed and no output files (e.g. hit or RDO files) were created. CPU time is also measured as a function of other simulation input parameters prior to significant changes, for example using different physics lists. For these runs, output files were disabled; in simulating $t\bar{t}$ events the time per event is increased by $\sim 0.5\%$ when file writing is enabled. The hard scattering events shown in table 3 were generated with a 14 TeV center of mass energy; for 10 TeV center of mass energy the simulation time is reduced by 17% for $t\bar{t}$ events.

For the samples in tables 3 and 6, event generation of W-boson production, minimum bias, di-jet, and photon and jet events was done using Pythia. $t\bar{t}$ production was done using MC@NLO for the hard scattering and Herwig for hadronization and showering. Heavy ion event production was done using Hijing.

Table 2. Simulation times per event, in kSI2K seconds[4], for single particles generated with $|\eta| < 3.0$. All times are averaged over 500 events. Logging and data output were suppressed.

Particle	$p_T = 1$ GeV	$p_T = 5$ GeV	$p_T = 50$ GeV	$p_T = 500$ GeV
Single electrons ($e^{\pm}$)	3.62	17.8	179.	-
Single muons ($\mu^{\pm}$)	-	0.879	1.63	12.0
Single charged pions ($\pi^{\pm}$)	2.40	10.4	94.7	-

Table 3. Generation, simulation, digitization, and reconstruction times per event, in kSI2K seconds[4]. Event generation times are averaged over 5000 events, except generation of heavy ion events, which were averaged over 250 events. The generation time for $t\bar{t}$ events includes only the hadronization time, not the time consumed by MC@NLO generation. Simulation, digitization, and reconstruction times are averaged over 250 events, except simulation of heavy ion events, which were averaged over 50 events. The heavy ion event simulation time is for events with a random impact parameter. Central collisions require on average 3.4 times longer to simulate.

Sample	**Generation**	**Simulation**	**Digitization**
Minimum Bias	0.0267	551.	19.6
$t\bar{t}$ Production	0.226	1990	29.1
Jets	0.0457	2640	29.2
Photon and jets	0.0431	2850	25.3
W$\rightarrow e\nu$	0.0788	1150	23.5
W$\rightarrow \mu\nu$	0.0768	1030	23.1
Heavy ion	2.08	56,000	267

2.3. Simulation Output Storage

The output from the simulation is a hit file, containing some metadata describing the configuration of the simulation during the run, all requested truth information, and a collection

[4] All performance measurements were done on Sun Fire X2200 M2 units with dual dual-core 2.6 GHz AMD Opteron 2218 processors. Normalization was done using the peak specmark int 2000 rating 1794. For the same system, the peak specmark floating point 2000 rating was 3338. The normalization follows the published results, rather than the WLCG formula in [13]

of hits for each subdetector. The hits are records of energy deposition, with position and time, during the simulation. Each subdetector is responsible for implementing their own code for the selection, processing, and recording of these hits. In the inner detector and muon systems, this consists simply of recording all hits that occur in sensitive regions of the detector for subsequent storage. Some additional manipulation is done at the end of each event to compress the output as much as possible; still, the files are typically 2 MB per event for hard scattering events (e.g. $t\bar{t}$ production). The file sizes here are uncompressed and taken from ROOT. In practice, compression reduces the actual disk space required for the files, but file-level metadata adds several hundred kilobytes.

The file size is large, mostly due to the inner detector, for which the majority of hits must be independently stored and cannot be merged. In $t\bar{t}$ events, for example, these consume 65% of the disk space in a hit file. In the calorimetry, there are far too many hits created by electromagnetic and hadronic showers for the individual storage of a four-vector for each. Instead, hit merging occurs at the end of each event. By optimizing time binning, hits can be compressed to a large extent. About 10% of the hit file is consumed by "calibration hits" for the calorimeters, hits in dead material, stored to improve the detector calibration and missing energy calculation and to study simulation-based calorimeter calibration schemes. These calibration hits are only included when certain samples (e.g. di-jets) are simulated. Under normal circumstances, the muon systems contribute a negligible portion of the hit file. The contributions by subdetector, including calibration hits, can be found in table 4 for the average of 50 simulated $t\bar{t}$ events.

Table 4. Hit collection size, in kb per event, by subdetector. The average was taken of 50 simulated $t\bar{t}$ events. Calorimeter calibration hits are hits in the dead material of the calorimeters stored for studying simulation-based calorimeter calibration schemes.

Collection Name	**Size [kb/event]**	**Percentage of File**
Silicon pixel tracker	82	4%
Silicon strip tracker	356	16%
Transition radiation tracker	921	46%
Electromagnetic Barrel Calorimeter	89	4%
Electromagnetic Endcap Calorimeter	104	5%
Hadronic Barrel Calorimeter	29	1%
Hadronic Endcap Calorimeter	22	1%
Forward Calorimeter	42	2%
Calorimeter calibration hits	243	12%
Muon system (all collections)	3	<1%
Truth (all collections)	134	7%
Total	**1987**	**100%**

Although the muon system is large, the majority of it is shielding. Therefore, it collects far fewer hits than the other subsystems and requires less disk space for the hit records. The calorimetry produces 95% of the hits in sensitive regions during simulation. Because of the compression applied prior to storage, the calorimetry comprises only 25% of the hit file.

The output files from event generators are tens of kB per event for physics events (e.g. 40 kB per event for $t\bar{t}$ events) and far less for simpler events (e.g. less than 1 kB per event for single electron events). They consume a negligible amount of disk space on the Grid. Digitization file output (RDOs) are 2-3 MB per event, depending on the activity of the event and the luminosity simulated (i.e. the number of minimum bias events overlaid on the hard-scatter). On the Grid, digitization and reconstruction are frequently done in a single pass and the digitization output file is immediately deleted to save space. The collection sizes are not, therefore, as carefully monitored as those of the hits files.

2.4. Physics Validation

Once a new release is distributed to the Grid sites, a set of several physics samples is produced. Typically, a "validation sample" includes 10,000 events for each process, a total of 110,000 single particle events and 250,000 hard scattering events. This standard validation sample includes single muons, pions, and electrons, Standard Model processes ($t\bar{t}$ production, vector boson production, B-physics), and exotic processes (e.g. supersymmetric events and black hole production). The composition of the validation sample has been chosen to test all aspects of the event reconstruction.

The running of the validation sample on the Grid usually exposes rare software problems in the release. It is unlikely that software bugs that appear with a frequency much lower than 1/1000 events are caught by the automatic validation procedure. This first round of production provides a feedback mechanism for the developers, who produce bug fixes before the next production cycle.

The last step before using a release for production is physics validation [15]. A dedicated group of experts, including representatives from every detector performance (e.g. tracking, b-tagging, and jet reconstruction groups) and physics group (e.g. Standard Model, supersymmetry, and exotics search groups) in ATLAS, runs physics analyses on the validation samples. Their task is to verify the quality of the single object reconstruction (e.g. jets, electrons, and muons) and the results of more complex physics analyses (e.g. mass reconstruction in Z$\rightarrow \mu^+\mu^-$, Z$\rightarrow e^+e^-$, and $t\bar{t}$ events). The relatively large validation samples may expose minor problems that could not be found with lower statistics, for example a shift of a few percent in the reconstructed energy. In order to properly validate each version of the software, the results from each release are typically compared to those of previous validated releases. The software must, therefore, maintain backwards-compatibility in order to allow fair comparisons. Shifts in file format are carefully coordinated, and maintenance of the old format is continued for as long as necessary to ensure result consistency. The physics validation procedure is also used for checking major changes in the fast and full simulation (detector description, change in the simulation parameters, etc.).

The Geant4 simulation has also been validated in a physics sense with all available detector data. Combined test beam studies have proven invaluable in understanding the performance of each of the subsystems, and the standalone test beam analyses have provided crucial input towards the optimization of the simulation and choice of parameters [16, 17, 18]. In 2008, a significant sample of cosmic ray data was collected with multiple subdetectors. The data have provided an important test of the simulation [19, 20].

Extensive efforts are underway to compare simulated data to real data and validate the output from each detector. For example, subdetectors can be "weighed" in the simulation to ensure that the amount of material is within a few percent of the constructed detector. Thanks to the multiple detector descriptions, several analyses have already been prepared and tested to find discrepancies between the detector description of the simulation and that of the as-built detector. Although the agreement with first collisions data is not expected to be perfect, a great deal of testing and tuning experience has been gained. The effects of modifications to Geant4 parameters have also been studied in some detail, so that differences between data and simulation might be remedied rapidly after first collisions.

Digitization algorithms have been tuned against laboratory test results, test beam data, and, where possible, cosmic ray data taken during the detector commissioning. These studies will continue once beam data are available.

3. Fast Simulations

Because of the complicated detector geometry and detailed physics description used by the ATLAS Geant4 simulation, it is impossible to achieve the required simulated statistics for many physics studies without faster simulation. To that end, several varieties of fast simulation programs have been developed to complement the Geant4 simulation. In this section, the standard Geant4 simulation will be referred to as "full simulation."

Almost 80% of the full simulation time is spent simulating particles traversing the calorimetry, and about 75% of the full simulation time is spent simulating electromagnetic particles. The Fast G4 Simulation aims to speed up this slowest part of the full simulation [21, 22]. The approach taken, therefore, is to remove low energy electromagnetic particles from the calorimeter and replace them with pre-simulated showers stored in memory. Using this approach, CPU time is reduced by a factor of 3 in hard scattering events (e.g. $t\bar{t}$ production) with little physics penalty. This simulation may eventually become the default simulation for all processes that do not require extremely accurate modeling of calorimeter response or electromagnetic physics.

Atlfast-I has been developed for physics parameter space scans and studies that require very large statistics but do not require the level of detail contained in the full simulation [23, 24]. Particles from the event generator are smeared by detector resolutions to provide physics objects similar to those of the reconstruction (e.g. an "electron" object without any shower shape variables). Object four-vectors are output, without any detailed simulation of efficiencies and fakes. The smearing functions are currently derived from the Geant4 simulation. A factor of 1000 speed increase over full simulation is achieved with sufficient detail for many general studies.

Atlfast-II is a fast simulation meant to provide large statistics to supplement full simulation studies. The aim is to try to simulate events as fast as possible while still being able to run the standard ATLAS reconstruction. Atlfast-II is made up from two components: the Fast ATLAS Tracking Simulation (Fatras) for the inner detector and muon system simulation [25, 26] and the Fast Calorimeter Simulation (FastCaloSim) for the calorimeter simulation. Fatras uses a simplified geometry and parameterized physics modules to speed up simulation of tracking detectors. FastCaloSim uses a parameterization of electromagnetic and hadronic showers. It simplifies the parameterizations further by depositing energy at the readout-cell level, rather than in every volume of the detector. The term "Atlfast-II" is used to refer to FastCaloSim with full simulation of the inner detector and muon system, and "Atlfast-IIF" is used to refer to Fatras and FastCaloSim used together. Optionally, any subdetector can be simulated with Geant4 to provide the higher level of accuracy without the same CPU time consumption as full simulation of the entire detector. An improvement over full simulation time of a factor of 10 is achieved with full Geant4 inner detector and muon simulation and FastCaloSim, and a factor of 100 is achieved with Fatras and FastCaloSim.

3.1. Computing Performance

Examples of simulation times normalized[4] to kSI2K seconds [14] for various types of events in the full and fast simulations are provided in tables 5 and 6. In single central ($|\eta| < 3$) electron events the simulation time is decreased by a factor of ten or more by the fast G4 simulation, and in hard scattering events the simulation time is decreased by a factor of 2-5. Atlfast-II decreases simulation time by a factor of 20-40, and Atlfast-IIF decreases simulation time by a factor of 100. FastCaloSim accounts for about 10% of the total simulation time in Atlfast-II and 60-70% of the total simulation time in Atlfast-IIF. Atlfast-I requires a relatively negligible amount of CPU time even for hard scattering events. FastCaloSim, and Fatras run during the reconstruction step, but for these purposes the time consumed by their methods is included in "simulation time."

In evaluating these CPU times, it is necessary to keep in mind the additional steps required

Table 5. Simulation times per event, in kSI2K seconds[4], for single particles generated with $|\eta| < 3.0$ and with the same transverse momentum. All times are averaged over 500 events. Atlfast-II uses full simulation for the inner detector and muon system and FastCaloSim in the calorimetry. Atlfast-IIF uses FastCaloSim in the calorimetry and Fatras in the inner detector and muon system.

Sample	Full Sim	Fast G4 Sim	Atlfast-II	Atlfast-IIF	Atlfast-I
5 GeV $\mu^{\pm}$	0.879	0.899	1.28	0.633	0.011
50 GeV $\mu^{\pm}$	1.63	1.15	2.71	0.606	0.011
500 GeV $\mu^{\pm}$	12.0	10.4	11.8	0.615	0.011
1 GeV $e^{\pm}$	3.62	0.734	0.825	0.513	0.011
5 GeV $e^{\pm}$	17.8	1.64	1.00	0.542	0.011
50 GeV $e^{\pm}$	179.	4.86	1.25	0.588	0.013
1 GeV $\pi^{\pm}$	2.40	1.48	0.701	0.515	0.011
5 GeV $\pi^{\pm}$	10.4	4.27	0.811	0.540	0.011
50 GeV $\pi^{\pm}$	94.7	30.3	1.04	0.569	0.011

Table 6. Simulation times per event, in kSI2K seconds[4], for the full simulation, Fast G4 simulation, Atlfast-II, Atlfast-IIF, and Atlfast-I. Atlfast-II uses full simulation for the inner detector and muon system and FastCaloSim in the calorimetry. Atlfast-IIF uses FastCaloSim in the calorimetry and Fatras in the inner detector and muon system.All times are averaged over 250 events, except heavy ion times which were averaged over only 50 events. Because the memory required to reconstruct heavy ion events exceeds 3 GB and because FastCaloSim runs during the reconstruction step, the amount of time taken by FastCaloSim could not be measured in that sample. It was estimated as 10% of the full inner detector simulation time, consistent with the other hard scattering events.

Sample	Full Sim	Fast G4 Sim	Atlfast-II	Atlfast-IIF	Atlfast-I
Minimum Bias	551.	246.	31.2	2.13	0.029
$t\bar{t}$	1990	757.	101.	7.41	0.097
Jets	2640	832.	93.6	7.68	0.084
Photon and jets	2850	639.	71.4	5.67	0.063
W→ $e\nu$	1150	447.	57.0	4.09	0.050
W→ $\mu\nu$	1030	438.	55.1	4.13	0.047
Heavy ion	56,000	21,700	∼3050	203	5.56

before analysis of the data can be performed. For both full and fast G4 simulation, the data must be digitized and reconstructed. For Atlfast-II, the inner detector and muon system must be digitized[5] and reconstructed, but the calorimeter requires only reconstruction. For Atlfast-IIF, only the muon system must be digitized before reconstruction is performed. The output of Atlfast-I is in a format similar to that of the reconstruction and needs no further processing. The CPU time required for these additional steps is given in table 3.

4. Summary and Conclusions

ATLAS has produced robust and well-validated simulation software which has been used to simulate 500 million events in the last year. The performance of the software, including CPU

[5] The inner detector and muon system together require about 2/3 of the total digitization time.

time required per event, memory required, and output file size, is continuously monitored. Thanks to this monitoring, crashes are very rare in production (less than one event in 10^7).

Three flavors of fast simulation have been introduced. These fast simulations complement the full Geant4 simulation and allow studies with much higher statistics than would otherwise be possible. Each is well-suited to a different use-case, and each is undergoing thorough validation by the ATLAS physics groups.

The software project has been prepared for data since late 2008 and is ready for first collision data coming later in 2009. The validation program will be used to produce a high quality simulation sample for the ATLAS experiment in preparation for data.

References

[1] The ATLAS Collaboration, Aad G *et al* 2008 *JINST* **3** s08003
[2] Evans L 2007 *New J. Phys.* **9** 335
[3] 2008 Athena core software http://atlas-computing.web.cern.ch/atlas-computing/packages/athenaCore/athenaCore.php
[4] C Ay *et al* 2009 *J. Phys.: Conf. Ser.* Monte Carlo Generators in ATLAS Software (to appear in the CHEP 2009 proceedings)
[5] Agostinelli S *et al* 2003 *Nucl. Instr. Methods Phys. Res.* **A 506** 250–303
[6] Allison J *et al* 2006 *IEEE Transactions on Nuclear Science* **53** 270–278
[7] JD Chapman *et al* 2009 *J. Phys.: Conf. Ser.* The ATLAS Detector Digitization Project for 2009 Data Taking (to appear in the CHEP 2009 proceedings)
[8] Bird I (ed) *et al* 2005 *CERN-LHCC-2005-024*
[9] 2008 Gcc, the gnu compiler collection http://gcc.gnu.org/
[10] 2008 Scientific linux cern 4 (slc4) http://linux.web.cern.ch/linux/scientific4
[11] Obreshkov E *et al* 2008 *Nucl. Instrum. Meth.* **A584** 244–251
[12] B Simmons *et al* 2009 *J. Phys.: Conf. Ser.* The ATLAS RunTimeTester Software (to appear in the CHEP 2009 proceedings)
[13] 2009 Cpu normalization criterion http://www.egee.cesga.es/EGEE-SA1-SWE/accounting/guides/NormalizationCriterion/index.html
[14] Henning J 2000 *Computer* 28–35
[15] D Costanzo *et al* 2009 *J. Phys.: Conf. Ser.* Physics and software validation for ATLAS (to appear in the CHEP 2009 proceedings)
[16] Carli T, Hakobyan H, Henriques-Correia A and Simonyan M 2007 *ATL-TILECAL-PUB-2007-008* And notes in preparation
[17] Speckmayer P 2008 *XIII International Conference on Calorimetry in High Energy Physics (CALOR 08), Pavia, Italy, May 26-30, 2008* ed M Fraternali G G and Livan M (Journal of Physics: Conference Series)
[18] Kiryunin A E, Oberlack H, Salihagić D, Schact P and Strizenec P 2008 *XIII International Conference on Calorimetry in High Energy Physics (CALOR 08), Pavia, Italy, May 26-30, 2008* ed M Fraternali G G and Livan M (Journal of Physics: Conference Series)
[19] Cooke M *et al* 2007 *ATL-LARG-PUB-2007-013*
[20] R Nikolaidou *et al* 2009 *J. Phys.: Conf. Ser.* Muon identification procedure for the ATLAS detector at the LHC using Muonboy reconstruction package and tests of its performance using cosmic rays and single beam data (to appear in the CHEP 2009 proceedings)
[21] Barberio E *et al* 2007 *ATL-SOFT-CONF-2007-002*
[22] Barberio E *et al* 2007 *J. Phys.: Conf. Ser.* **119** 001001 proceedings of Computing in High Energy and Nuclear Physics 2007 Conference (CHEP2007), Victoria, BC, Canada, September 2-7, 2007. Also ATL-COM-PHYS-2007-012
[23] Richter-Was E, Froidevaux D and Poggioli L 1998 *CERN-ATL-PHYS-98-131*
[24] Dean S and Sherwood P 2008 Athena-atlfast http://www.hep.ucl.ac.uk/atlas/atlfast/
[25] Edmonds K, Fleischmann S, Lenz T, Magass C, Mechnich J and Salzburger A 2008 *ATL-SOFT-PUB-2008-01*
[26] U Husemann *et al* 2009 *J. Phys.: Conf. Ser.* Simulation strategies using FATRAS and Geant4 for a future upgrade of the ATLAS tracking detectors (to appear in the CHEP 2009 proceedings)