ATL-COM-SOFT-2003-013
05.11.2003

ATLAS Data Challenge 1

The ATLAS DC1 Task Force[1]

The ATLAS Collaboration at CERN is preparing for the data taking and analysis at LHC that will start in 2007. Therefore, in 2002 a series of Data Challenges (DC's) was started whose goals are the validation of the Computing Model, of the complete software suite, of the data model, and to ensure the correctness of the technical choices to be made. A major feature of the first Data Challenge (DC1) was the preparation and the deployment of the software required for the production of large event samples for the High Level Trigger and Physics communities, and the production of those large data samples as a worldwide distributed activity.

It should be noted that it was not an option to "run everything at CERN" even if we had wanted to; the resources were not available at CERN to carry out the production on a reasonable time-scale. We were therefore faced with the great challenge of organising and then carrying out this large-scale production at a significant number of sites around the world. However, the benefits of this are manifold: apart from realising the required computing resources, this exercise builds worldwide momentum for ATLAS computing as a whole.

The first phase (event generation, simulation) of DC1 was run during Summer 2002, and involved 40 institutes in 19 countries. In the second phase (October 2002-March 2003) the next processing step ("pile-up") was performed with the participation of 56 institutes in 21 countries. Distributed reconstruction of the most demanding high-statistics samples was carried out at the 9 largest sites (April-June 2003)

Much has been learned from DC1, and much more will doubtless be learned over the next months. However, we can already be rather confident that ATLAS will be able to marshal world-wide resources in an effective way; let us hope that the Grid will make it all rather easy.

This report describes in detail the main steps carried out in DC1 and what has been learned from them.

1.) Introduction

The LHC Computing Review[2] recommended having data challenges (DC) of increasing size and complexity. The ATLAS collaboration[3] planned to perform these DC's in the context of the LHC Computing Grid (LCG) project[4]. The experience gained during these exercises will be used to formulate the ATLAS Computing TDR, which is due, according to the present planning, in 2005. The Grid technologies promise several advantages for a multinational, geographically distributed project: they allow for a uniform infrastructure of the project computing-wise, simplify the management and coordination of the resources while potentially decentralizing such tasks as software development and analysis, and last, but not least, the Grid is an affordable way to increase the computing power. If the ATLAS Data Challenges can demonstrate that usage of the Grid, indeed, gives all those advantages, the collaboration should become committed to "gridification" of its sites and tools, by making use of the best available Grid middleware.

During the LHC preparation phase, all experiments have large needs for simulated data, to design and optimise the detectors. This "Monte Carlo" simulation is done in the following steps:

- Particles emerging from the collisions (called collision final state or simply final state) are generated using programs usually based on physics theories and phenomenology (called generators);

[1] See author list at the end of the paper
[2] http://lhc-computing-review-public.web.cern.ch/lhc-computing-review-public/Public/Report_final.PDF
[3] http://www.cern.ch/Atlas
[4] http://lcg.web.cern.ch/lcg/

- The particles of the generated final state are transported through the simulation of the detector according to the known physics laws governing the passage of particles through matter;
- The resulting interactions with the sensitive elements of the detector are converted into information similar to the digital output from the real detector (the "digitisation" step);
- The events are reconstructed.
- The (Monte Carlo) generated information (sometimes called *truth*) is saved for comparison with the reconstructed information.

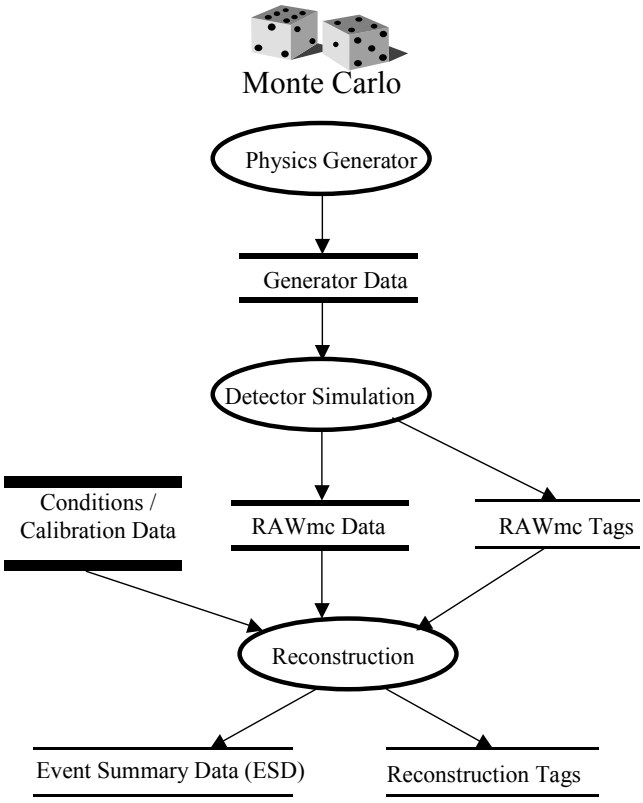


Fig. 1 The different steps of the Monte Carlo production

In this report the ATLAS Data Challenge 1 (DC1) is described. After some short section about the LCG Project (Section 2) and the ATLAS Data Challenges (Section 3), the different phases of DC1 are discussed: Generation and Simulation (Section 4), Pile-Up Generation (Section 4) and Reconstruction (Section 5). This is followed by a description of the various tools developed and used during DC1: Bookkeeping and Databases (Section 7), Production Tools (Section 8) and Software Distribution (Section 9). The resource available in the different phases and the amount of data that was processed are described in Section 10 (Phase1), Section 11 (Phase 2) and Section 12 (Phase 3). The DC1 activities using the different Grid testbeds are discussed in Section 13.

2.) The LCG Project

The job of the LHC Computing Grid Project – LCG – is to prepare the computing infrastructure for the simulation, processing and analysis of LHC data for all four of the LHC collaborations. The LCG scope spans both the common infrastructure of libraries, tools and frameworks required to support the physics application software, and the development and deployment of the computing services needed to store and process the data, providing batch and interactive facilities for the worldwide community of physicists involved in LHC. The main emphasis of the LCG project is the deployment of Grid technologies for the LHC computing.

The first phase of the project, from 2002 through 2005, is concerned with the development of the application support environment and of common application elements, the development and prototyping of the computing services and the operation of a series of computing data challenges of increasing size and complexity to demonstrate the effectiveness of the software and computing models selected by the experiments. This first phase will conclude with the production of a Computing System Technical Design Report, providing a blueprint

for the computing services that will be required when the LHC accelerator begins production. This will include capacity and performance requirements, technical guidelines, costing models, and a construction schedule taking account of the anticipated luminosity and efficiency profile of the accelerator.

A second phase of the project is envisaged, from 2006 through 2008, to oversee the construction and operation of the initial LHC computing system.

3.) ATLAS Data Challenges

For all data challenges it is essential to have physics content in order to bring the physicists community into the exercise and for a more through validation of the software.

The goals of the ATLAS Data Challenges are the validation of the Computing Model, of the complete software suite, of the data model, and to ensure the correctness of the technical choices to be made. It is understood that these Data Challenges should be of increasing complexity and will use the software which will be developed in the LCG project, to which ATLAS is committed, as well as the Grid middleware being developed in the context of several Grid projects like EU Data Grid or GridPP. The results of these data challenges will be used as input for a Computing Technical Design Report and for preparing a Memorandum of Understanding in due time.

It is important to mention that for DC1, in 2002-2003, a major goal was to provide simulated data to the High Level Trigger (HLT) community needed for the preparation of the HLT Technical Design Report (TDR) by mid 2003[5]. The ATLAS Trigger system must accept the high 40 MHz bunch crossing frequency and reduce it to a manageable rate of roughly 200 Hz. Events from the first-level (Level-1) hardware-based trigger are passed on to a second-level software-based trigger (Level-2) at a rate of 75 kHz which must derive a decision within an average latency of 10 ms. Level-2 accepted events are passed on to the third-level software-based Event Filter (EF) at a rate of roughly 3 kHz which has a more generous latency of roughly 1 s to pass the event on to offline mass storage with a rate of roughly 200 Hz. It is axiomatic that only events surviving this three-stage triggering system can be part of subsequent physics analysis. Together, the Level-2 and EF are referred to as the HLT.

DC1 was scheduled to run from April 2002 to early part of 2003. It was divided into three phases. In the first phase, April-August 2002, we put in place the infrastructure and the production tools to be able to run the 'massive' production worldwide. The second phase, started in October 2002, the goal was to produce the pile-up data. The third phase was the reconstruction of the simulated data. The reconstruction with the 'pure' offline reconstruction code is completed. Reconstruction using HLT algorithms is ongoing as of September 2003.

The event generation can use several event generators (e.g. Pythia, Herwig, Isajet, etc.) and can run either in the Fortran ATLSIM[6] framework or in the official ATLAS ATHENA[7] framework. The fast simulation, ATLFAST[8], was used to control the quality of the generated data. The current detector simulation code called DICE is Fortran-based, uses Geant 3.21 to track the events through the detector and runs in the ATLSIM framework. Events are written out in the form of ZEBRA[9] banks. Most of the reconstruction packages have been moved to OO/C++, even if some packages are still in Fortran. The new reconstruction uses the ATHENA framework and was used, for the first time, in a large-scale production.

Essential components required for ATLAS Monte Carlo production are the associated bookkeeping and meta-data services, as described below.

[5] ATLAS TDR 016; CERN/LHCC/2003-022; ISBN 92-9083-205-3
[6] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/ATLSIM/atlsim.html
[7] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html
[8] http://atlas.web.cern.ch/ATLAS/GROUPS/PHYSICS/HIGGS/Atlfast.html
[9] CERN Program Library Q100/Q101

4.) Generation and Simulation

4.1 Event Generation

The generation of all event samples was done at CERN using Pythia 6.203[10] running inside ATHENA (ATLAS releases 3.2.0 and 3.2.1). The events were converted into HepMC[11] and then written out into ROOT I/O using the ATHENA Root conversion service.

Several samples of physics events were generated[12]. Among them: **"jet"; "minimum bias"; single W**; **single Z**: **W+jet; Z+jet; Photon+jet; inclusive top; Higgs** and **MSSM Higgs Samples; selected samples of b-physics events with exclusive and semi-inclusive b-decays;** with different characteristics (transverse momentum; decays, etc).

4.2 Generation Monitoring

The quality of the generated events produced was monitored with the use of histograms of various characteristic properties of those events. These histograms were produced after the generation by running a job that invoked a purpose-written algorithm called HistSample (in the ATLAS ATHENA framework) on the generator output (ROOT-IO format) HistSample produced the basic histograms that were then written to RZ-format output files. An n-tuple was also written into the same file, of which more will be said below.

The `prototype' sample considered comprised Z+jet events with the Z decaying to e^+e^-, $\mu^+\mu^-$ and $\tau^+\tau^-$. The HistSample histograms plotted the p_T and the mass of the generated Z as well as some more general quantities such as the rapidity, pseudorapidity and number of charged tracks in the event. In addition, the mass of e^+e^-, $\mu^+\mu^-$ and $\tau^+\tau^-$ pairs and p_T of leptons were histogrammed. Despite being constructed for a specific sample, these histograms are of use for many other event classes, although clearly the reference histograms to which they should be compared will differ.

The n-tuple that was also produced by the HistSample algorithm contains quantities related to the jet structure of the event. Jet finding was performed by running ATLFAST with the normal smearing turned-off, and then making use of the associated ATLFAST utilities to perform the jet finding at the particle level in the generator output. The n-tuple is then used in a secondary job, which runs a KUMAC in the PAW[13] environment to produce histograms of the number of reconstructed jets, their p_T spectra and pseudo-rapidity distributions. These are normalised in various ways: to the number of events; to the number of jets; and to the total cross section. It is because of the need for these latter normalisations that this second set of histograms was produced in a second step and not in the HistSample job.

Finally, the two-histogram samples were merged and a postscript summary of all of the histograms produced was made and checked for consistency with the physics expectations for the given sample. The various output files were put into long-term storage using the CERN Advanced Storage Manager CASTOR[14].

The system encountered various technical difficulties, notably the access to the very large input files.

4.3 Event Simulation

The ATLAS detector simulation was done in the ATLSIM framework using GEANT3[15]. ATLSIM is a PAW-based framework, which uses KUIP[16] for job control. It has an improved memory management, eliminating any hard limits on the track/vertex/hit numbers. It also has improved hadronic physics based mainly on the

[10] http://www.thep.lu.se/~torbjorn/Pythia.html
[11] http://mdobbs.web.cern.ch/mdobbs/HepMC/
[12] Details: see Appendix A
[13] http://paw.web.cern.ch/paw/
[14] http://castor.web.cern.ch/castor/
[15] CERN Program Library W5013
[16] http://wwwasdoc.cern.ch/wwwasdoc/shortwrupsdir/i202/top.html

GCALOR[17] package. A number of known infinite loops were eliminated prior to major DC1 productions. Low energy K^0_L particles are traced by GHEISHA[18] to avoid known problems in FLUKA[19].

ATLSIM uses plug-in components (shared libraries) to provide extra I/O facility (e.g. ROOT) and to load ATLAS detector geometry. The description of the ATLAS geometry is taken from the DICE[20] package.

Compared to the Physics TDR[21], several updates have been made to reflect the design modifications since 1998:

- Beam pipe: multi-layer beam pipe design.
- Pixel detectors: symmetrical, insertable layout
- Pixel detectors: innermost layer at larger radius
- SCT: tilt angle reversed (to minimize cluster size)
- TRT: modular design of the Barrel
- Inner detector services material updated
- Realistic field in Inner Detector

- All End-Cap Calorimeters shifted by 4 cm
- ENDE: dead material, readout updated
- TILE: material and readout update
- HEND: dead material updated + readout update (4^{th} sampling added)
- FWDC: detailed design with precise rod positions
- ACCB: readout update
- Muon system design corresponds to the AMDB version P.03

During the simulation-phase di-jet events produced by PYTHIA where analysed by a filtering routine which looked for a predefined energy deposition in two neighbouring towers in η-ϕ space. Only events selected by the filter were passed to the simulation step and then written out.

4.4 Quality assurance and data validation

The aim of the ATLAS DC quality assurance and validation procedure[22] was:

- to ensure the compatibility and reproducibility of the samples produced at different sites,
- to monitor the changes and improvements to the ATLAS detector geometry,
- to check the physics contents of the generated samples.

An essential tool for site validation is a semi-automated system to compare the outputs from two simulation runs identify differences between them. The validation test-suite consists of a modular analysis structure based on PAW, which runs off a general-purpose n-tuple (CBNT) from the ATLAS reconstruction framework (ATRECON[23]), and which contains information on MC event generation and the reconstruction for all ATLAS sub-detectors.

The analysis procedure consists of two steps. First, a (open-ended) list of sub-detector specific macros is run from a master process to produce the two sets oft validation histograms. Secondly, a histogram-by-histogram comparison is performed between two sets of validation histograms, providing a bin-by-bin significance plot and a χ^2 test. At the end a summary χ^2 bar-chart for all compared histograms is made.

The validation of participating institutions was done by comparing the simulation of identical input samples from different sites and by comparisons of larger, statistically independent, samples of the same physics process. The validation provided an important checking of the simulation infrastructure at the contributing DC sites. For, example, it allowed to spot slight but significant differences of the run-time libraries. During the initial phase this was a quite complex and intensive but absolutely necessary activity.

[17] http://wswww.physik.uni-mainz.de/zeitnitz/gcalor/gcalor.html
[18] Simulation of hadronic showers, physics and application, PITHA 85-02, H.C. Fesefeldt, RWTH Aachen
[19] http://www.fluka.org/
[20] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/DICE95/dice95/dice95.html
[21] ATLAS Detector and Physics Performance Technical Design Report LHCC 99-14/15
[22] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DC/Validation/www/
[23] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/reconstruction.html

A broad variety of validation samples of dedicated single particle scans and physics benchmark processes (H, Z) were produced to validate the full simulation chain. This initial phase proved to be important for the discovery of missing/faulty/new aspects of the detector geometry implemented in the simulation. This phase demonstrated also that for a data-handling exercise of this scale, the quality and stability of services such as AFS, CASTOR or the batch system have to be improved, in order to optimise the use of effort and time.

The physics validation of the data was carried out in parallel with the other checks. The di-jet 17GeV sample (15M events) has been processed with the fast simulation package AtlfastOO. A comparison of the events content for multiplicities of jets, b-jets, c-jets, electrons and photons, with a similar sample produced in '96/97 large-scale production, was performed. In addition fully simulated samples were inspected and were used e.g. for detailed detector calibration purposes. New samples were also used extensively to study b-tagging with the full or reduced layout of the inner detector.

The b-physics group validated the DC1 simulation-reconstruction software chain by performing the reconstruction in the Inner detector using different releases (4.4.0 - 6.5.0). These studies were done for several detector layouts (DC1, Initial and Complete) and the performance characteristics were compared to those obtained with the TDR layout[24].

Only relatively few people participated in the first two aspects of the validation. The complexity and intensity of this initial stage of the process did not lend itself to the engagement of a larger group. However, far more people particularly from the physics groups took part in the “physics validation”, and this wider engagement was essential.

5.) Pile-Up Generation

5.1. Pile-Up Procedure

The cross-section for inelastic, non-diffractive pp interactions at the LHC is expected to be around 67 mb. At design luminosity (10^{34} $cm^{-2}s^{-1}$), the average number of minimum-bias events is 23 per bunch crossing. This number varies according to a Poisson distribution. Any collision recorded in the ATLAS detector, contains therefore a superposition of particles coming from several events. In general the particles from a single "physics" event will have triggered the readout, and other particles will come from other un-selected pp collisions. The total number of observed particles per recorded event depends on the signal collection time, which varies from a few ns in silicon detectors to about 700 ns in the Muon Drift Tubes (MDT). In the muon system additional pile-up arises from the cavern background. To take care of this effect, special minimum-bias files were produced which included the cavern background on top of the “normal” pile-up event. The full pile-up is simulated as a number of minimum bias collisions properly distributed in time and overlaying the “physics” collision.

While in Liquid Argon (LAr) calorimeters the signal is measured shortly after the trigger, so that it is affected only by previous bunch crossings, measurements in drift detector such as Transition Radiation Tracker (TRT) or MDT continues for the maximum signal collection time so that they are sensitive to the same amount of posterior bunch crossings.

As every collision is normally simulated only for few 100 ns of propagation time, there is one additional component, which must be added explicitly: neutrons may fly around the ATLAS cavern for few seconds until they are thermalised, thus producing kind of a permanent neutron-photon creating a constant rate of Compton electron and spallation protons, which are observed in the muon system. This component, i.e. additional hits created by long living particles, is called "cavern background". Technical details about the pile-up generation are given in Appendix E.

5.2 Resources needed for Pile-up production

As for the standard ATLSIM/Geant3 simulation the digitisation is accounted for in the simulation. We estimate the following average numbers for piled-up events in the η range $|\eta| < 3$:

[24] DC1-b-physics validation: ATL-COM-PHYS-2003-003

Luminosity	Output size/event (MB)	CPU-time/event (SI95-s)
$2*10^{33}$ cm^{-2} s^{-1}	3.6	2000
10^{34} cm^{-2} s^{-1}	7.5	8000

The list of runs, which were processed, can be found on the web[25].

6.) Reconstruction

6.1 Standard Offline Reconstruction

The new Atlas reconstruction software is based on the Athena/Gaudi framework. In short, the Athena framework embodies a separation between data and algorithms. Data are handled through Transient Event Store (TES) for event information and a Transient Detector Store (TDS) for condition information. Data is written in the TES either by converters reading the input data (decoding the Zebra bank in this case) or by Algorithms. Data (specified by object type and string key) is read by Algorithms, or persistified by Converters. Algorithms are driven through a flexible event loop. Common utilities are provided through services. ASCII files called job Options allow to specify algorithms and services parameters and sequencing. The Athena executable itself is very small, all the significant software being dynamically loaded at run time, with typically one library per package.

The full reconstruction suite was evolved from and tested against Atrecon, the Fortran reconstruction program used for the Physics TDR. Although a complete suite is available, only the so-called e/gamma slice, required by HLT needs, was used in this Data Challenge. Tracks were reconstructed with two independent algorithms, paying special attention to electron parameters fitting. Electronic noise was simulated in all calorimeters. Electromagnetic calorimeter clusters were then searched for, matched to the tracks, and identification variables calculated.

Since the data challenge happened before the new C++ database POOL being developed in the LCG context was available, the output of reconstruction was stored in HBOOK ntuples. This was done using a special algorithm, named CBNT for ComBined NTuple, capable of writing the TES content into an ntuple through the Gaudi ntuple converter. The algorithm is fully configurable, so that all and only the needed information is written out, which is especially important for the large truth information. The main drawback is that the downstream analysis can only be done in PAW (or ROOT after conversion), having lost on the way the C++ design of the original objects. Different limitations in HBOOK were also hit.

Luminosity	Output size/event (MB)	CPU-time/event (SI95-s)
$2*10^{33}cm^{-2}s^{-1}$	0.02	3000
$10^{34}cm^{-2}s^{-1}$	0.03	7600

The output size and CPU usage is shown in the table above. Given the large CPU time needed, it was decided to have each reconstruction job to reconstruct only one file at a time, so typically 300 di-jet events. The reconstruction code in the release used (6.0.3) had remaining memory leaks of order 200 kBytes per high luminosity event (down from several MBytes in earlier versions)[26], which was manageable given the small number of events processed in each job.

6.2 HLT Reconstruction

[25] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DC/DC1/DC1_2/production_requests.html

[26] The memory leak was reduced to about one kByte per event in later versions of the code.

Algorithms running in the HLT Selection Software (HLTSSW) environment reconstruct objects and extract features from event data; these features are used to derive the trigger decision. At Level-2, highly specialized algorithms use a restricted portion of event data usually defined in terms of Region-of-Interests (RoI) derived from the Level-1 decision. Modified algorithms from the offline software are used as Event Filter algorithms and have potential full access to event data. In both cases, algorithms must be capable of being seeded from results derived at a previous stage of the trigger chain.

To facilitate the importation of algorithms from the offline software as well as to permit a configurable continuum of selection in the HLT by means of interchangeable Level-2 and EF algorithms, a common Event Data Model (EDM) is essential. The establishment of a common EDM within the ATLAS offline and online software environments is an on-going effort; for DC1 purposes, special LVL2 event data objects were retained. Furthermore, at Level-2, algorithms actively request and process only small fractions of event data. The relevant data are defined by RoIs based on information from the decision from Level-1 or a previous result in Level-2 processing. For each RoI, the total data volume with respect to the whole detector is roughly a few percent.

The HLT processing flow is disaggregated into Steps. The Step Controller (SC) of the Steering software replaces the Athena Event Loop Manager and has the responsibility of calling algorithms. XML files encode Sequences and Signatures that in turn instruct the Steering on when and how to run an algorithm and if a physics signature is fulfilled. Signatures and Sequences are built upon Trigger Elements (TE). The TEs characterize abstract physics objects with a succinct label (e.g., "e20i" for an isolated 20 GeV electron). Input TEs provide seeds to algorithms executing in each step. The decision to go further in the process is taken at every new Step by the comparison between active TEs in the TES and the corresponding configuration Signature. An event is accepted if its entire constituent Sequences have been executed and at least one of the corresponding Configuration Signatures has been satisfied.

In the case of Level-2 track reconstruction involving the precision Pixel and SCT sub-detectors, two parallel algorithms have been developed: IDSCAN and SiTrack. A clustering algorithm for electromagnetic (EM) showers, T2Calo, is seeded by the Level-1 EM trigger RoI positions and separates isolated EM objects from jets using the cluster E_T and certain shower-shape quantities. The muFast algorithm is a Level-2 track reconstruction algorithm for the Muon Spectrometer, steered by the RoI given by the Level-1 Muon Trigger and uses both RPC and MDT measurements.

Event Filter algorithms consist of algorithms imported directly from those developed in the offline Software and are described in the HLT TDR.

7.) Bookkeeping and Databases

Essential components required for ATLAS Monte Carlo production are the associated bookkeeping and meta-data services. Therefore several bookkeeping and production tools were developed for or adapted to the usage in DC1 (as described in this section):

- *AMI*[27] *(ATLAS Metadata Interface*), developed at LPSC Grenoble, a database containing meta-data on produced datasets and partitions (name, size, processing time, physics contents, transformations, etc.), with command-line and web interfaces, and various search possibilities

- *MAGDA*[28] *(MAnager for Grid-based DAta*), developed at BNL, which has been in use as an automated file registration and replication tool;

- the *VDC (Virtual Data Catalogue*), developed at BNL, a database containing production 'recipes', and a tool to assemble production scripts.

In addition production tools (as described in section 8) were developed and used to ease the production and the monitoring of the DC1 data production. Among them:

- AtCom[29] (short for ATLAS Commander), developed at CERN, an automated job definition, submission and monitoring tool, directly working with AMI

[27] http://atlasbkkl.in2p3.fr/AMI/
[28] http://www.atlasgrid.bnl.gov/magda/info

- GRAT[30], the Grid Application Tools developed in the context of the US Grid projects.

All tools make use of, or are based on, MySQL databases. Based on experience gained during DC1, these tools are constantly being developed further and improved.

7.1 The AMI Database

The Atlas Metadata Interface (AMI) project aims to provide a set of generic tools for managing database applications. The application was originally developed as an online electronic notebook for the LAr test beam data, and later adapted as a prototype application to store the metadata of offline calculations. DC1 AMI was greatly expanded adding a command line interface, a generic web search interface and some specialized web interfaces. For DC1 AMI was used to store the "metadata" needed to describe the data itself. Thus AMI makes it possible:

- to understand the contents of a file of binary physics data without actually having to open it,
- to search for a data logical filename or list of logical filenames, given a set of "logical" attributes[31],
- to get information about the provenance of each dataset.

The ATLAS Metadata base has been interfaced to the EDG WP2 package "Spitfire18". This package provides a secure grid-enabled front-end to relational databases.

Database Design

The bookkeeping application is written in JAVA. Consequently, it is independent of platform, operating system and database technology. The only prerequisite is that JAVA is installed on the client system. A 3-tier architecture is used. The core packages manage the remote connection to the database, and the transmission of SQL commands. Any database which understands SQL, and for which a java JDBC driver is available may be used. The middle layers provide generic classes for accessing the bookkeeping databases, using their internal descriptions. Top layers of the software are project specific.

The architecture allows geographic distribution of bookkeeping; all connections pass through a central router, which redirects requests to the correct site. This central router should be mirrored. For DC1 however, all the databases were physically at the LPSC Grenoble, and situated on the same server.

The Command Line Interface

This is the interface used by physicists to input and update information in the databases. In general, scripts operating on the batch job log files generate a set of AMI commands. A large number of commands are available, including commands to query the database schema.

The Web Interfaces

AMI contains a generic read-only web interface for searching. It is generated from the auto-descriptions of the core databases, which means that the database schema can be changed without touching the web interface code. The interface has a "quick" search, where certain fields for searching are pre-selected, and an "advanced" search, which gives access to all the database fields. In both cases, the SQL constructed is visible to the user, and can be directly edited if the user desires. Users can navigate from the search results to a graphical showing the

[29] http://atlas-project.atcom.web.cern.ch
[30] http://heppc1.uta.edu/atlas/software/grat
[31] A logical attribute is an attribute specific to the application, and which could not be guessed by any outside system, for example Grid software.

provenance of a dataset, and also to the MAGDA database which contains information on the physical location of the files.

Other special Atlas Production interfaces have been provided:

- Users can request a new dataset by specifying the desired characteristics. The request is then sent to the production manager.
- The Production manager can examine the requested dataset, and either validate the dataset, with the possibility of editing some fields, or refuse it.
- A third interface is available to give a rapid overview of the state of production. Some simple statistics are available, and it is possible to obtain a pie chart of jobs done per production site.

7.2 MAGDA

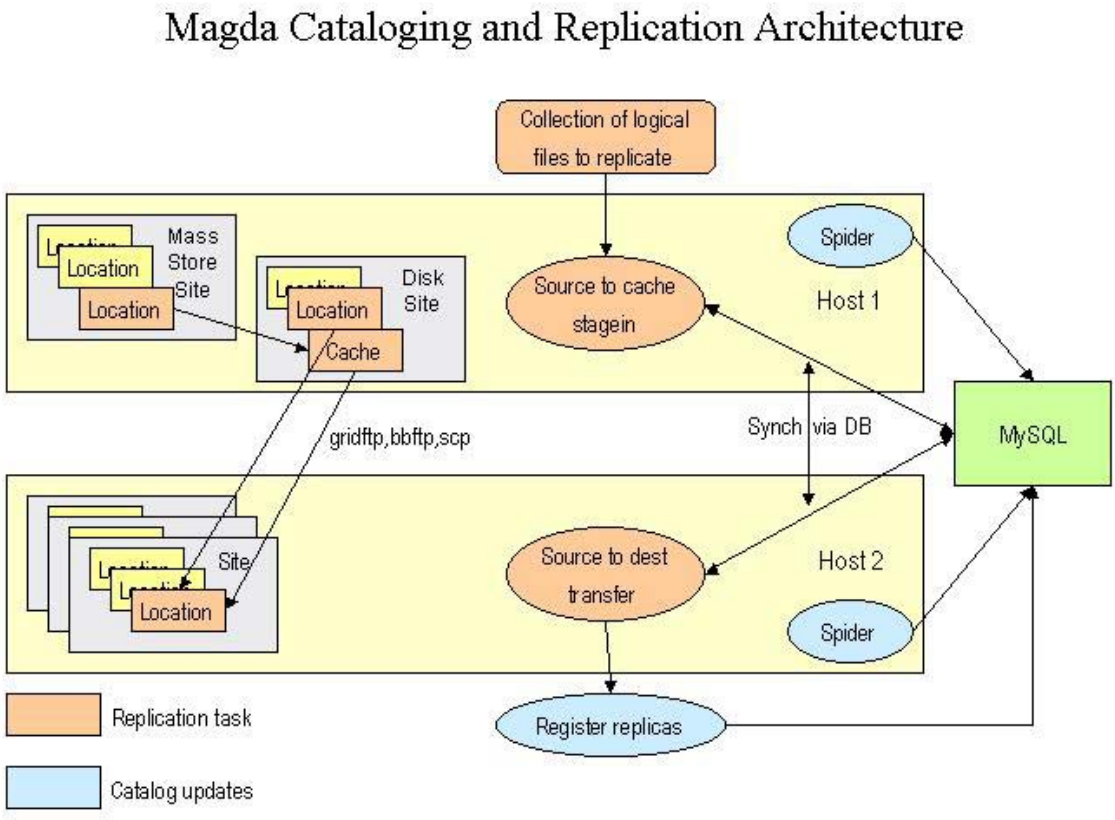


Fig 2. MAGDA Cataloguing and Replication Architecture

MAGDA manages the database, which describes where the data reside, thus it complements AMI, which describes what the data is. MAGDA was developed to fulfil the principal ATLAS '01-'02 deliverable for the Particle Physics Data Grid (PPDG) project[32] of a production distributed data management system deployed to users and serving BNL, CERN, and many US ATLAS grid test-bed sites.

MAGDA makes use of the MySQL open source relational database, Perl, Java, and C++. MySQL was chosen because of an existing experience base with it and because of its proven performance in data management for the STAR/RHIC experiment. For data movement gridftp and (where grid infrastructure is not available or usable) bbftp, scp are used. The Globus replica catalogue is currently being integrated.

[32] http://www.ppdg.net

The 'core' of the system is a MySQL database, but the bulk of the system is in a surrounding infrastructure for setting up and managing distributed sites with associated data locations, data store locations within those sites, and the hosts on which data-gathering servers and user applications run; gathering data from the various sorts of data stores; interfacing to users via web interfaces for presenting and querying catalogue info and for modifying the system; and replicating and serving files to production and end-user applications.

All files generated for DC1 in the U.S. Grid test-bed were put in the BNL HPSS storage system using MAGDA. MAGDA also managed the replica location for these files – so more than 10000 files were automatically registered in MAGDA. In a subsequent step all DC1 files were registered in MAGDA.

7.3 Prototyping Virtual Data Approach

Because of the physics-oriented content of ATLAS Data Challenges the recipes for producing the ATLAS data (ATHENA job Options and other similar "input data cards" files) have to be fully tested. The data produced have to be validated through a subsequent quality assurance and validation step. Preparation of the production recipes takes time and efforts, encapsulating considerable knowledge inside. Once the proper recipes have been prepared, producing the data is straightforward. Because of the prevailing vision that the data are primary and the recipes are secondary (they needed just for the data production) it has not been clear how to treat the developed recipes after the data have been produced. It was decided to store these recipes outside of the scope of the ATLAS Bookkeeping Database AMI.

A valuable insight for ATLAS production workflow has been provided by introduction of the "virtual data concept". The GriPhyN project[33] emphasises this perspective:

- recipes are as valuable as the data,
- production recipes are the virtual data.

Taking this approach to the extreme means that if you have the recipes you do not need the data (because you can reproduce them), i.e., the recipes are primary and the data are secondary. According to the virtual data architecture, recipes are stored in the virtual data catalogue database.

In the process of the ATLAS Data Challenge we have evaluated the virtual data approach for the production of several datasets. The ATLAS database group developed and delivered an infrastructure for early application of virtual data concepts and techniques to ATLAS data production. A virtual data catalogue database prototype was deployed in the spring of 2002 for evaluation in the context of the ATLAS Data Challenges. The prototype has been used successfully for data challenge event generation and detector simulation. Production job options for physics event generation and production scripts for detector simulation were recast as parameterised transformations to be catalogued, with the resulting parameterisations represented as derivations. ATLAS DC0 and DC1 parameter settings for simulations are recorded in the virtual data catalogue database.

The production system, based on the virtual data catalogue prototype, implemented the scatter-gather data processing architecture to enable high-throughput computing. The production fault tolerance has been enhanced by the use of the independent computing agents, adoption of the pull-model for agent tasks assignment (instead of push model typically used in batch production) and by the local caching of output and input data. An interesting feature provided by this architecture is the possibility for the automatic "garbage collection" in the job planner in the following sequence: production agents pull the next derivation from the virtual data catalogue; after the data has been materialized, agents register "success" in the database; when previous invocation has not been completed within the specified timeout period, it can be invoked again.

8.) Production Tools

8.1 ATCOM

[33] http://www.griphyn.org

The purpose of AtCom is to automate as much as possible the task of a production manager: defining and submitting jobs in large quantities, following up their execution, scanning log files for known and unknown errors, updating the various ATLAS bookkeeping databases in case of success, cleaning up and resubmitting in case of failure.

The design of the tool is modular, separating the generic basic job management functionality from the interactions with the various databases on the one hand, and the computing systems on the other hand. How to interact with the various computing systems is defined separately in the form of plug-ins, which are loaded dynamically at run time. In anticipation of the likely eventuality that different flavours of computing systems (legacy and GRID) will be deployed concurrently at the various, or even a single ATLAS site, AtCom allows several of them to be used at the same time transparently.

The design of the tool assumes that jobs can be defined in a computing system neutral way. The current implementation features a virtual-data-inspired approach that equates job definitions with a reference to a transformation definition and actual values for its formal parameters. The transformation definitions include a reference to a script/executable, its needed execution environment in the form of 'used' packages, and a signature enumerating the formal parameters and their types.

The figure below shows the top-level architecture of AtCom. In the middle is the AtCom core application that implements the logic of defining, submitting and monitoring jobs. On the left are the two modules that interface AtCom to the ATLAS bookkeeping databases, respectively AMI and MAGDA. On the right there is the set of plug-ins that interface AtCom to the various flavours of computing systems.

The computing system plug-ins implement an abstract interface that defines methods and signatures for the usual operations: submitting a job, getting the status of a job, killing a job and getting the current output (stdout and stderr) of a job.

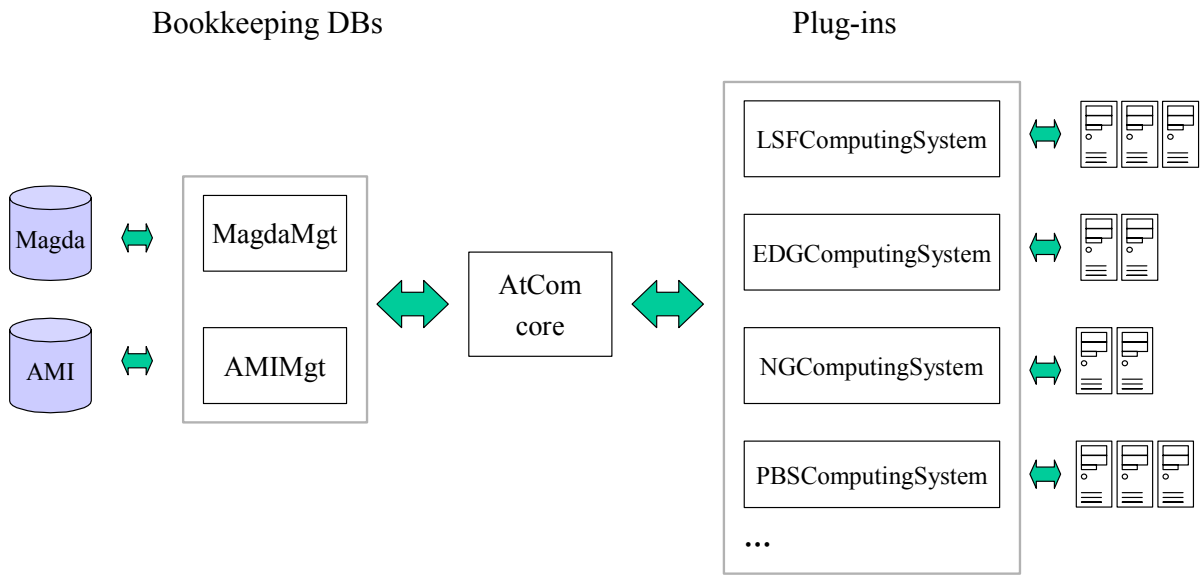


Fig. 3 The AtCom architecture

The underlying production model is based on the concepts of datasets, partitions, transformations and jobs. A dataset is a chunk of data that logically forms a single unit. Because of file-size limitations, datasets are for practical reasons split in a number of partitions each corresponding to a separate logical file. On the dataset level, abstract transformations create datasets based on a number of parameters and possibly taking one or more other datasets as input. Again, for practical reasons, this transformation process is implemented using a number of concrete transformations, each coinciding with a single job operating on the partition level.

AtCom supports three classes of operations: job definition, job submission and job monitoring.

From the definition panel, the user can select a dataset he/she wants to define into partitions, by means of an SQL query composer. The user defines the fields of the dataset he/she wants to see and the selection criteria. Pull-down menus allow the composition of the most common queries, but the query text can, if needed, be arbitrarily edited. The search is executed and the result is displayed. The user can then select a single dataset and choose a particular version of the associated transformation. Based on this concrete transformation's signature AtCom will compose a form that will allow the definition of the values for all required parameters for all the wanted partitions.

The second AtCom panel allows the user to submit any defined partition to any configured computing system. The procedure starts again with an SQL composer allowing the retrieval of a set of partitions. Given a set of retrieved partitions the user can select an arbitrary subset and select a target computing system for submission. The jobs are submitted and automatically transferred to the next panel for monitoring.

The monitoring panel allows the user to check the status of all monitored jobs on demand, or poll automatically at regular intervals. Additionally, the user can select a number of jobs and right click on them to invoke one of a large set of operations: kill, submit, refresh, revalidate, …

When a job moves from 'running' to 'done', post-processing is automatically started. If the job has terminated successfully, the output files are registered with the replica catalogue (MAGDA). If the job failed, the output as defined in the partition's output mapping are deleted and the status is set to 'failed'. If the job is 'undecided', the status is changed accordingly, pending a decision by the user.

AtCom has been used extensively at CERN since October 2002 and consequently has become optimally suited to the specific type of productions that take place there. CERN usually is the first ATLAS site to run any ATLAS code in production mode and consequently possible error conditions while running are often discovered there. Pre-productions are started, closely monitored, aborted, restarted, etc. Additionally, it has become customary that CERN processes the many smaller datasets, while outside institutes process a few smaller datasets or even just part of a single bigger dataset.

Even though AtCom's user base has been extremely small, it has been a major driving force in defining the bookkeeping databases, has acted as a catalyst for defining an ATLAS-wide uniform production framework (to be gradually introduced in the course of 2003), and has made a substantial contribution to this framework.

8.2 GRAT

The GRid Application Toolkit (**GRAT**) was developed to facilitate automated Atlas Monte Carlo production in a Grid environment. It currently consists of some 40 bash and python shell scripts, and is constantly undergoing modifications and updates to both add new features as well as adapting to the evolution of grid middleware (Globus, VDT, etc.).

At present GRAT provides the following production utilities:

1.) **Job definition –**
- adding new datasets
- incorporating additional execution steps

2.) **Job submission –**
- create single or multiple jobs at a remote site
- create jobs at multiple sites from a given dataset
- handles all execution steps (**gen**, **simul**, **redigi**, **lumi02**, **lumi10**, **reco**, etc.)

3.) **Verification –**
- data quality checks (via analysis of job log files)
- automatic error correction (move files, restart steps, etc.)
- failed job recovery (cleanup, database updates)

4.) **Storage management –**
- move/delete verified input files as necessary
- cleanup temporary storage areas upon job completion
- dispose of replica copies from intermediate steps

5.) **Site management –**
- monitor jobmanager queues and running jobs (via dynamic query and database checks)

- provide disk storage status (usage, free space, etc.)
- software checking for availability of required packages
- minbias (pileup) files prestaging and management

Data management tools are provided to facilitate interactions with the various databases (Production, MAGDA, VDC & AMI). These include:

1.) **Adding new information –**
   - create/update AMI entries and production information
2.) **Database queries –**
   - of single entries in the Production database
   - accessing summary information
   - as needed for decisions regarding jobs in "hung" states
   - characteristics of jobs waiting to process
3.) **Consistency checks –**
   - scan for and correct bad records
   - ensure the accuracy of replica copies
   - verify the existence of generated files in MAGDA
   - common data across multiple databases

Phases of Execution

An example of the execution flow for a typical simulation job is shown in the figure below:

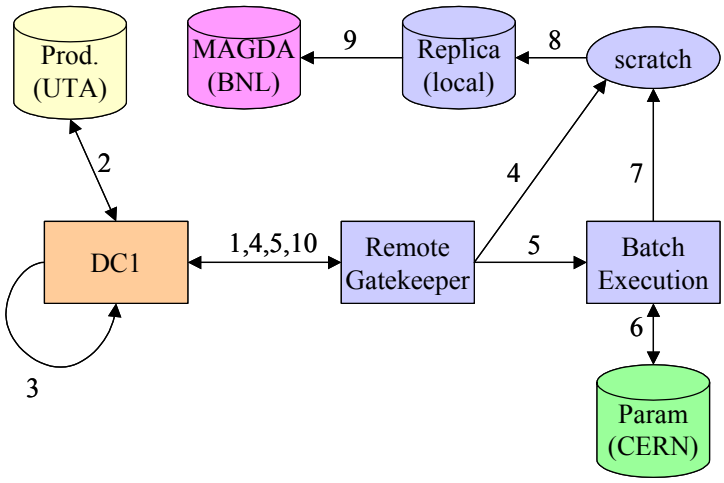


Fig. 4 Execution flow for a typical simulation job

The various steps in the process include:

1. **Resource Discovery** - The remote system is interrogated to discover the software environment, location of scratch space, and what services are configured.
2. **Production Coordination** - The Production db is contacted to determine which dataset should be simulated. The next available set is reserved and job-specific information is registered.
3. **Job Customization** - The information collected from the previous steps is used to create a job (*i.e.*, a set of scripts) for the remote system.
4. **Job Staging** - The job is transferred to the remote system's scratch area using GridFTP.
5. **Job Submission** - The control script for the job is submitted to the remote system's batch scheduling system via the Globus gatekeeper service.
6. **Execution/Parameterization** - At simulation start-up, the AMI db is contacted to retrieve the full set of parameters needed for the specific simulation chosen.
7. **Results -** The results of the simulation are stored within the remote system's scratch disk area.

8. **Staging** - The results are copied via GridFTP into the remote system's replica cache or alternate replica location and registered using MAGDA.
9. **Cataloging** - The job schedules a transfer, via GridFTP using MAGDA, from the replica location to the master location at Brookhaven National Laboratory (BNL) for permanent storage in the HPSS system.
10. **Monitoring** - Monitoring the state of the job, through query of the Production db, Globus queries and MAGDA queries, begins after step five and lasts until the results are stored at BNL. If any failures are discovered, the database records are deleted so that the job becomes available for the next production run.

9.) Software Distribution

The ATLAS software is split into circa 500 packages residing in a single CVS repository at CERN. A flexible tool, CMT[34], manages package dependencies, libraries and executable building. New releases are built at CERN approximately every three weeks, following a predefined plan for new features introduction. Integration is made easier by nightly builds of the release in development.

The compilation process is done on Linux machines (RedHat Linux 6.1 in the first two phases of DC1, RedHat 7.3 since then). Users with a good network connection and access to AFS may use executables and the data files directly linking them from CERN. This approach is of course not suitable for remote sites with a bad connection to CERN or without access to AFS. To deal with this situation, a set of RPM packages has been produced, in order to install the full ATLAS software distribution on machines both with and without AFS. This functionality has been available since release 3.0.0.

The RPM kit has been designed to be used on standard as well as on EDG machines, in order to fulfil the requirements of DC1. Each ATLAS software release is packaged into RPM format. The kit, along with the installation script, is downloadable[35] via secure web connection or, otherwise, from the EDG site[36].

The general criteria, followed during the package architecture development phase, have been to build a self-consistent distribution procedure, which is independent of the LINUX release. To fulfil these requirements the RPMs have been designed to keep the same directory structure as in the CERN repository and to include the reference gcc compiler (gcc v2.95.2), the ROOT version used for the build of the release and the required libraries not part of the ATLAS software. To be consistent with the reference software, produced at CERN, the executables and libraries included in the kit are the exact copies of the files stored in the public AFS software repository.

The packages are organized in a set of base tools, required for all the installations, and several additional components. A minimal installation should provide at least the following items:

-the set-up and management scripts;
-the official ATLAS compilers;
-the ROOT version using during the compilation phase;
-the required libraries not part of the ATLAS software (external packages).

This corresponds to the ATLAS-conf, ATLAS-tools, ATLAS-release, ATLAS-compilers, ATLAS-root and ATLAS-external RPMs. If the local system compiler is gcc v2.95.2 users may choose not to install the ATLAS-compiler package. Other packages are anyway required to generate, simulate and reconstruct the data; therefore it is highly recommended that the full set of RPMs be installed on each machine.

The kit installs itself under the directory /opt/ATLAS, using about 1 GB of disk space. Relocation is also possible, providing that the change of the root directory of the kit, from /opt/ATLAS to some other place, is also reflected in the configuration scripts, by editing them after the installation. For convenience, a relocation script is included in the kit, under the /opt/atlas/etc directory. To work with this kit, users must first configure the environment via the set-up script (/opt/atlas/etc/atlas.shrc). After this is done, the applications are ready to be executed. Some examples on how to run a simulation job are included in the kit in the ATLAS-DC1 package.

[34] http://www.cmtsite.org/
[35] https://classis01.roma1.infn.it/atlas-farm/atlas-kit
[36] http://datagrid.in2p3.fr/distribution/applications/wp8/atlas

The RPM suites have proven to be robust and efficient. Most of the countries and sites have installed the software using the official set of RPMs, but other types of installations have also been used in some sites for the DC1 production. In particular a procedure based on a full mirroring of the distributions directly from the CERN AFS repository, and an alternate procedure from a different set of RPMs, were developed by the Nordic Countries and used within the NorduGrid test-bed.

The main drawback found in the use of RPMs was the lack of flexibility: bug fixes in the new reconstruction software required entire new releases to be built and distributed. Fortunately fine-tuning of reconstruction through modification of the jobOption parameters was possible by distributing a lightweight RPM.

10.) Resources in DC1 Phase 1(Generation and Simulation)

In DC1 phase1 the data needed for the High Level Trigger (HLT) TDR were generated (i.e. 4-vector production using PYTHIA), followed, after some selection, by full simulation of the ATLAS detector response using ATLSIM (Dice, GEANT3). Due to the huge amount of computing time needed it was essential to make use of the computing resources available in ATLAS institutes around the world.

10.1 Countries participating in DC1 Phase 1

The following 40 institutes in 19 countries participated in DC1 phase 1:

1.) Australia (Melbourne**)**
2.) Austria (Innsbruck**)**
3.) Canada (Alberta, CERN**)**
4.) CERN
5.) Czech Republic (Prague)
6.) France (Grenoble + Marseille; using Lyon)
7.) Germany (München; using FZK)
8.) Israel (Weizmann)
9.) Italy (CNAF Bologna, Frascati, Milano, Napoli, Roma)
10.) Japan (Tokyo)
11.) NorduGrid: Denmark, Norway, Sweden (Bergen, Grendel, Ingvar, ISV, LSCF, Lund, NBI, Oslo)
12.) Poland (Cracow)
13.) Russia (Dubna, ITEP Moscow, MSU Moscow, Protvino)
14.) Spain (Valencia)
15.) Taiwan (Taipei)
16.) UK (Cambridge, Glasgow, Lancaster, Liverpool, RAL)
17.) USA (Arlington, BNL, LBNL, Oklahoma)

10.2 Resources available for DC1 phase 1

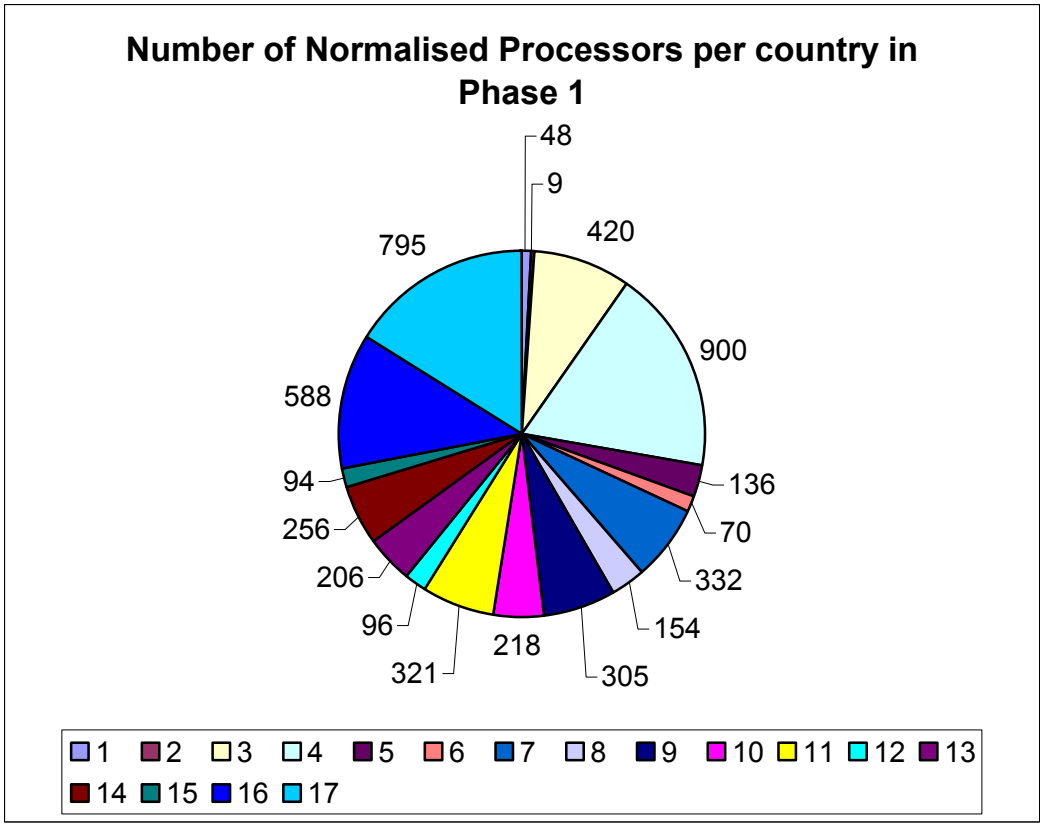


Fig. 5 Number of normalised processors per country accessible in DC1 phase 1. The numbering corresponds to the one in 10.1

The numbers of processors per site varied between 9 and 900. At peak time we used worldwide ~3200 processors (~5000 NCUs[37]) in 40 institutes located in 19 countries. This corresponds to ~115 kSI95 or ~50% of the CPU power estimated for one Regional Facility at the LHC start-up (2007). The hardware investment made by those institutes in the last 12 months corresponds roughly to 50 % of the yearly hardware investment needed from 2006 onwards for the non-CERN part of the ATLAS Offline Computing.

10.3 Data Samples for DC1 phase 1

Event type	Output size/event (MB)	CPU-time/event (SI95sec)
Single Particle	0.05	300
Minbias	1.00	4000
Di-jet event	2.40	13000

Average numbers for the different samples

During Phase 1 of DC1, about 50 million events in total were generated via PYTHIA; about 51 million events in total were passed through detailed detector simulation via ATLSIM. About 40 million were single-particle events (muons, photons, electrons, pions), the remaining ~ 11 million were complete physics events.

The production requests from the High-Level-Trigger community were organised into three main parts: validation samples (very high priority), high-statistics samples (mostly high priority), and medium-statistics samples (ranging from low to high priority). A web page[38] was set up to monitor the progress of the production activities. In addition to the original information (physics contents, simulation specifications, number of events, priority), this page contains also organisational (dataset numbers, groups-in-charge, status, etc.) and statistical

[37] Here we use as unit the Normalised CERN Unit (NCU) unless it is explicitly specified otherwise: 1 NCU corresponds to 1 Pentium III 500 MHz equivalent to 21 SpecInt95 (SI95).
[38] http://atlasinfo.cern.ch/ATLAS/GROUPS/SOFTWARE/DC/DC1/DC1_1/production_requests.html

(numbers of generated/simulated events, time to process one/all events, etc.) information relevant for the individual sub-samples. Most of this information will eventually be accessible from the bookkeeping database.

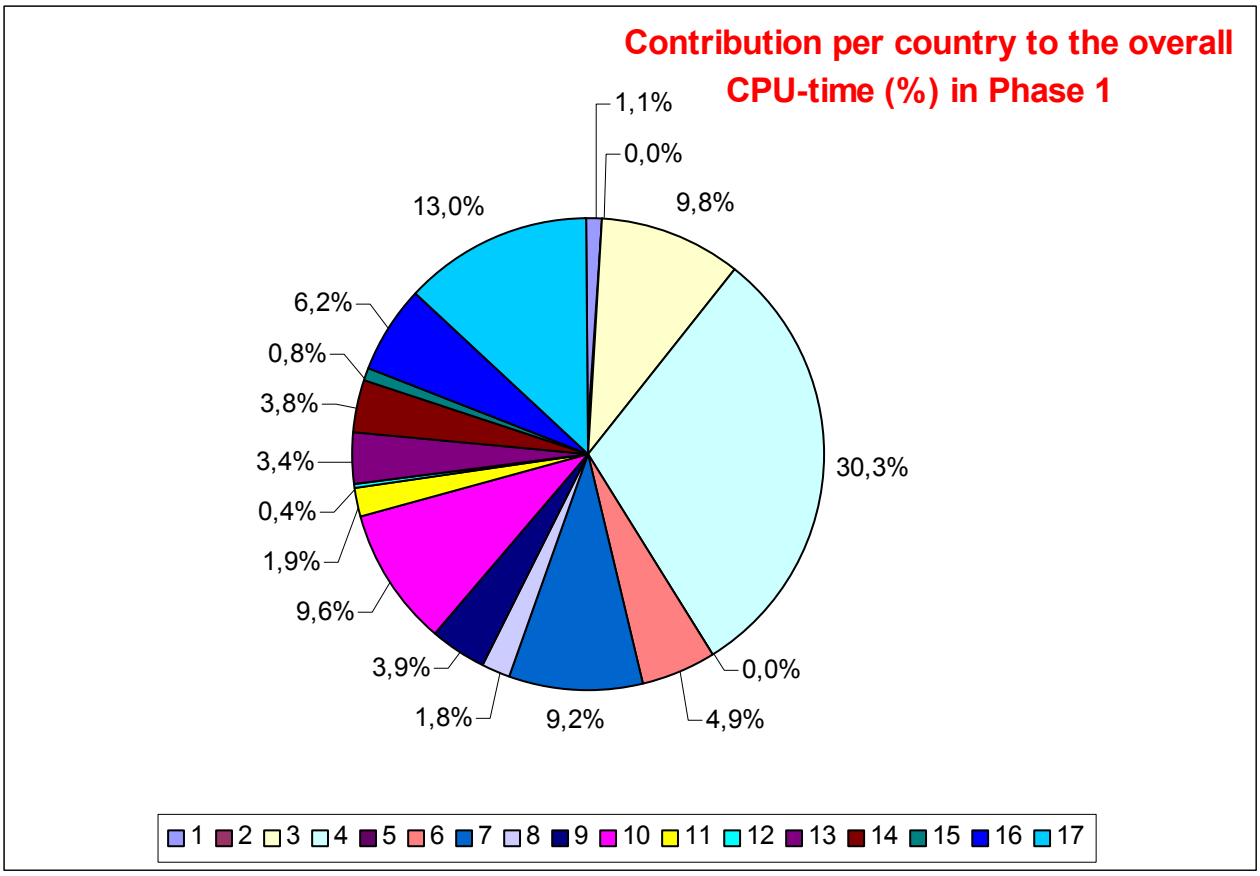


Fig. 6 Contribution per country to the overall CPU-time (%) in Phase 1. The numbering corresponds to the one in 10.1

The samples assigned the highest priority were the validation samples[39]. They consist of single-particle events, jet-scan samples, and some physics event channels taken from old TDR tapes. About 740k events were processed and 110 GBytes of data were produced, the time needed being about 900 CPU days.

The most challenging part, w.r.t. CPU and data storage requirements, was the production of the high-statistics samples[40]. They consist of 36 million single-muon events, about 5 million di-jet events of different E_T(hard scattering) cuts (applying particle-level filtering or not), and 1 million minimum-bias events simulated with different $|\eta|$ cuts. The data volume of the whole sample amounts to about 15 TBytes, the total CPU time needed to about 44000 NCU-days. Note that not all the produced data will be stored in the CERN CASTOR system; about 10 TBytes will be kept at different distributed production sites.

The medium-high statistics samples[41] comprise production requests by various subgroups of the HLT community: the e/gamma, Level-1, jet/ETmiss, B-physics, b-jet, and muon trigger groups; and a sample of about 80k single pions. The e/gamma samples contain a huge production of single-electron (1.1 million) and single-photon (1.6 million) events at different energies and η values. Sub-samples of the B-physics trigger and b-jet trigger samples were simulated for the Inner Detector only, the rest either with the "central" detector (ID+Calorimeters; e.g., the e/gamma single-particle production) or the full detector. All the ~ 7 million simulated

[39] http://atlasinfo.cern.ch/ATLAS/GROUPS/SOFTWARE/DC/DC1/DC1_1/validation/validation_samples.html

[40] http://atlasinfo.cern.ch/ATLAS/GROUPS/SOFTWARE/DC/DC1/DC1_1/highStat/high_statistics_samples.html

[41] http://atlasinfo.cern.ch/ATLAS/GROUPS/SOFTWARE/DC/DC1/DC1_1/mediumStat/medium_statistics_samples.html

events correspond to a data volume of about 9 TBytes, the total CPU time necessary to process them was 30000 NCU-days.

In summary, the total estimated data volume produced during DC1/1 is about 24 TBytes and about 8 TBytes for generated events; the total CPU time necessary to generate all the events was about 1000 NCU-days, the time to simulate all the events about 74000 NCU-days.

11.) Resources in DC1 Phase 2(Pile-up production)

In DC1 phase 2 pile-up was added to a sub-set of data samples as described in section 5.

11.1 Countries participating in DC1 Phase 2

New countries, China, Greece and new institutes from Canada, Italy, NorduGrid, UK and USA have joined the effort in the course of the second phase of DC1 so, now 56 institutes in 21 countries are participating in DC1 phase 2 giving a total of ~8000 NCU's.

1.) Australia
2.) Austria
3.) Canada
4.) CERN
5.) China
6.) Czech Republic
7.) France
8.) Germany
9.) Greece
10.) Israel
11.) Italy
12.) Japan
13.) NorduGrid: Denmark, Norway, Sweden
14.) Poland
15.) Russia
16.) Spain
17.) Taiwan
18.) UK
19.) USA

11.2 Resources available for DC1 phase 2

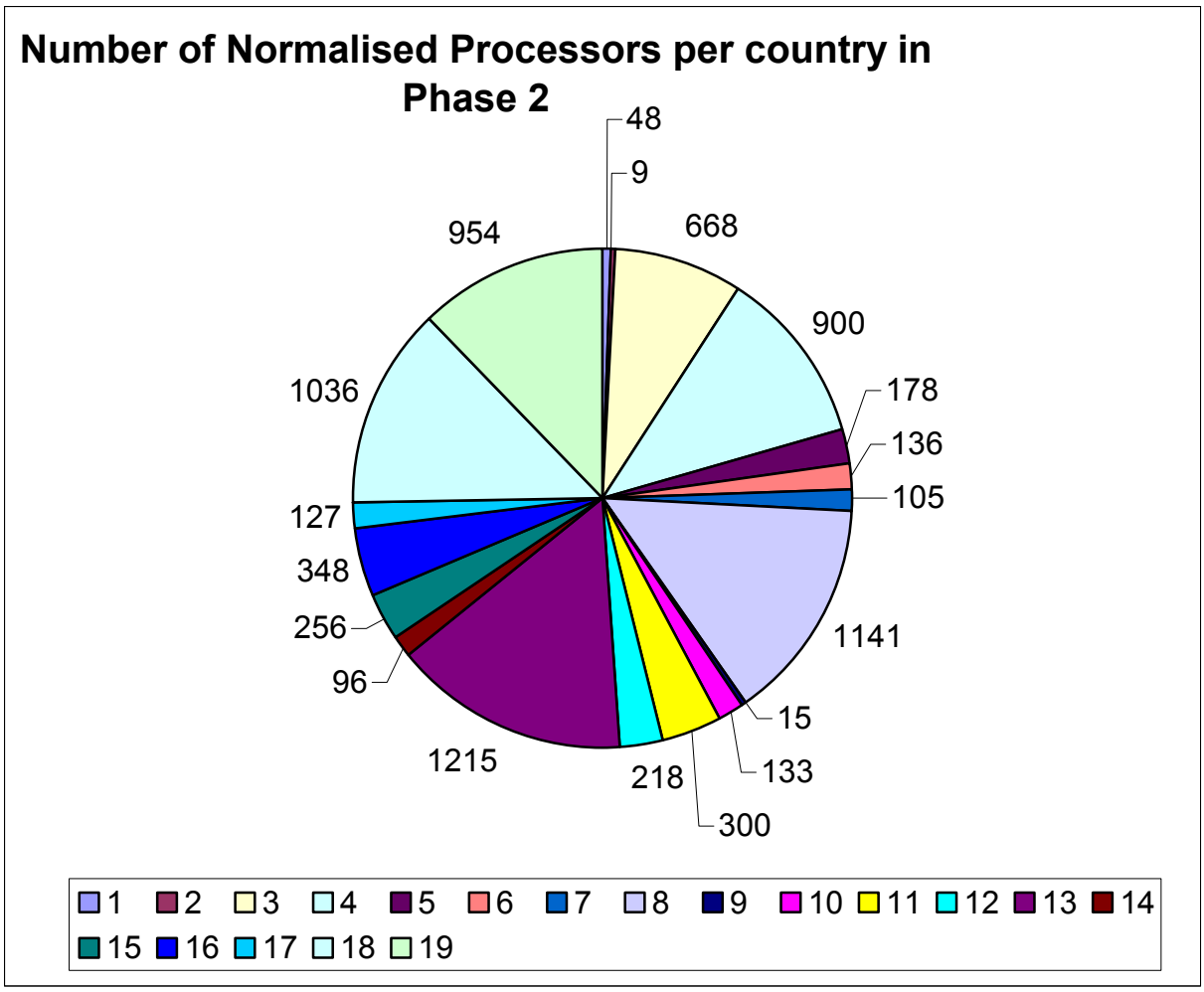


Fig. 7 Number of normalised processors per country accessible in DC1 phase 2. The numbering corresponds to the one in 11.1

11.3 Data samples for DC1 phase 2[42]

About 3900k events were produced for low and 2650k events for high luminosity. This part of DC1 took about 17000 NCU-days and produced a total data volume of about 34 Tbytes in 32000 partitions.

[42] Reference to the request from HLT

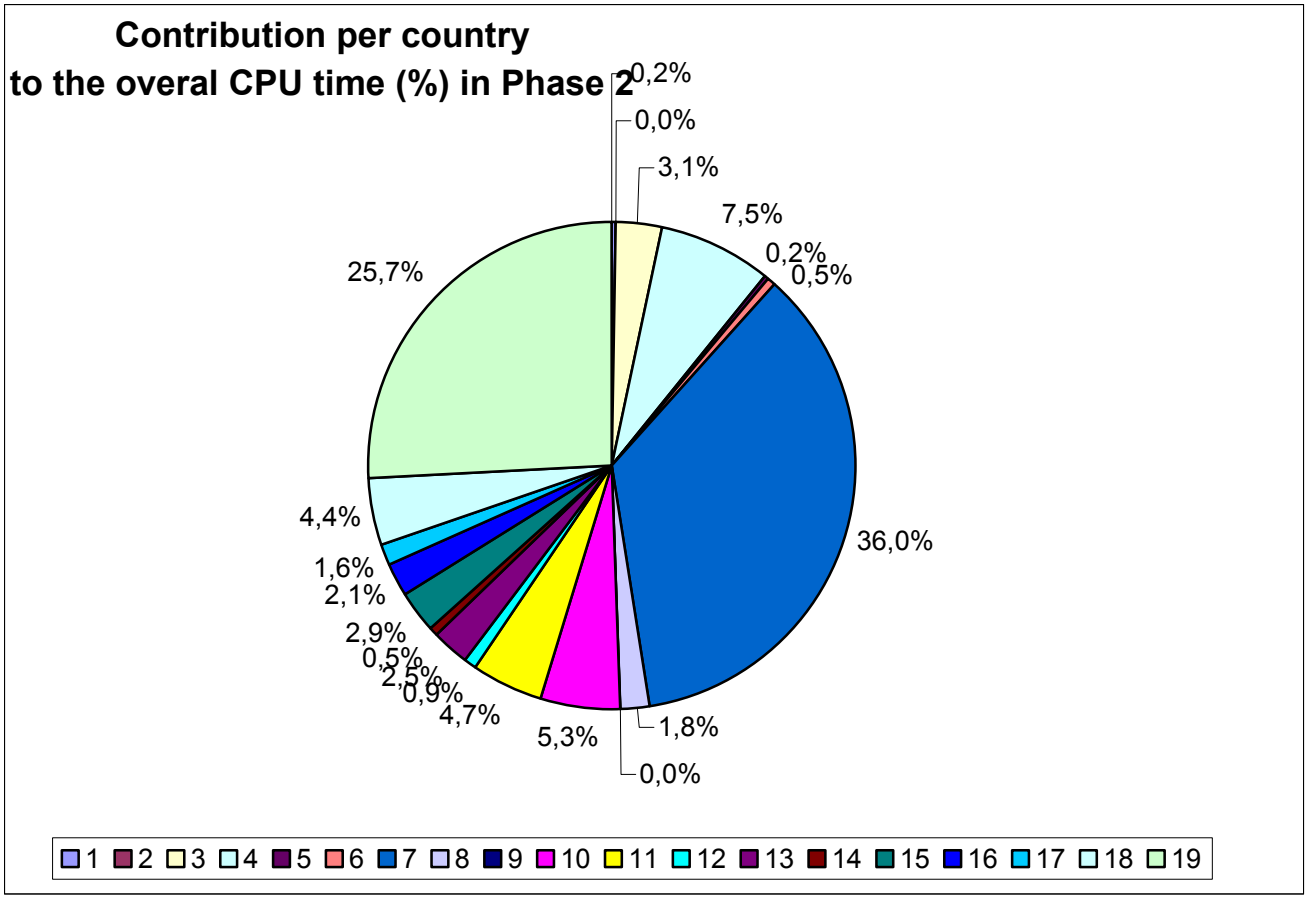


Fig. 8 Contribution per country to the overall CPU-time (%) in Phase 2.
The numbering corresponds to the one in 11.1

12.) Resources in DC1 Phase 3 (Reconstruction)

In DC1 phase1 the data needed for the High Level Trigger (HLT) TDR were generated (i.e. 4-vector production using PYTHIA), followed by full simulation, after some selection, of the ATLAS detector response using ATLSIM (Dice, GEANT3). As in the earlier phases the huge amount of computing time needed made it essential to make use of the computing resources available around the world.

12.1 Countries participating in DC1 Phase 3

To facilitate the access to the large distributed datasets, since not all production sites were accessible via Grid tools, the data were replicated to 8 sites (see Appendix D for details). Therefore, the processing of the data was mostly done in those countries, as can be seen in the figure below.

12.2 Data samples for DC1 Phase 3

About 6400 k events were processed during the reconstruction phase. This part of DC1 took about 23000 NCU-days and produced a total data volume of about 200 GBytes in 25000 partitions.

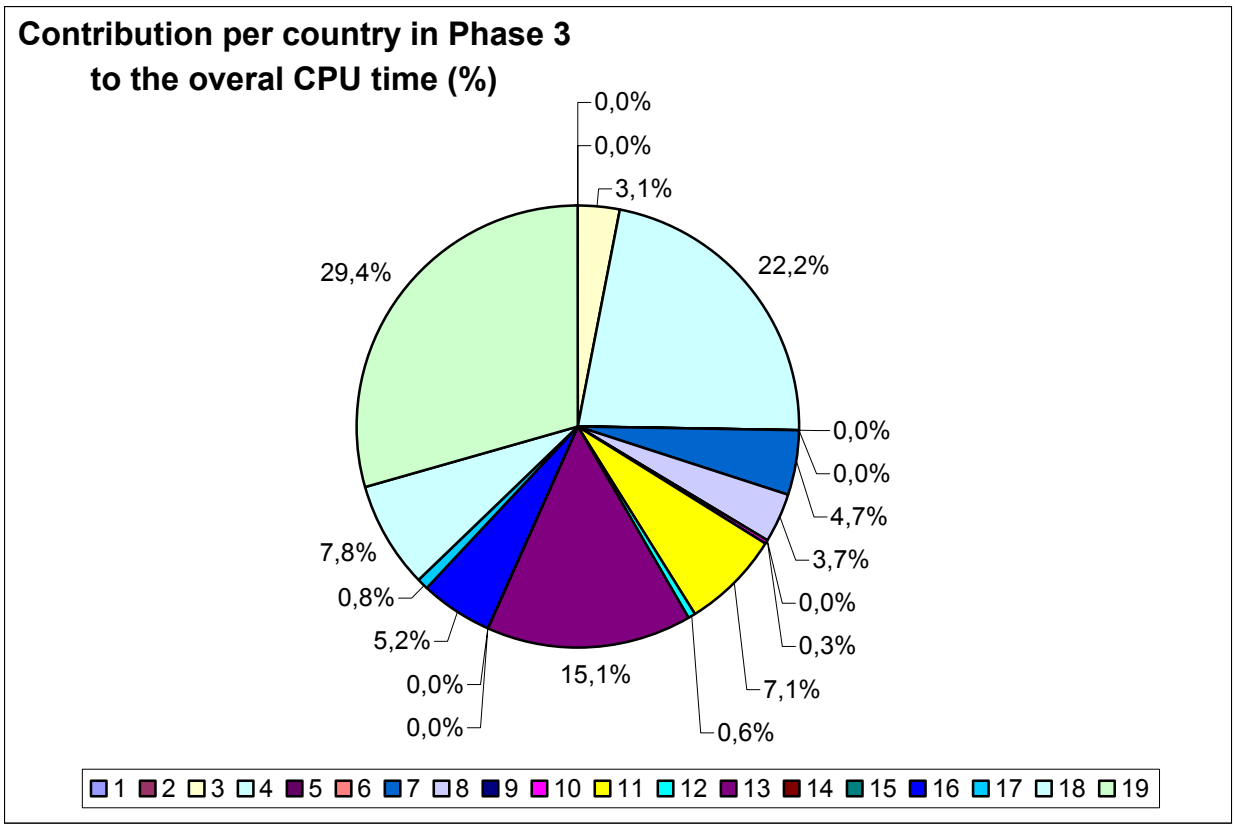


Fig. 9 Contribution per country to the overall CPU-time (%) in Phase 3.
The numbering corresponds to the one in 11.1

13.) Data Challenge 1 and the Grid

A recent and highly significant advance in computing is the emergence of Grid technologies. Powered by various middleware, Grid computing infrastructures are becoming a reality, and as such are particularly important for large distributed projects like the High Energy Physics experiments, and ATLAS in particular. By harnessing distributed and scarce resources into a powerful system, the Grid is expected to play a major role in the near future. Apart from optimisation of the usage of distributed resources, the Grid will naturally offer all the collaboration members a uniform way of carrying out computing tasks. This is essential for large production tasks, which need plenty of resources, both hardware and human, worldwide.

Data Challenges are the perfect opportunity to evaluate the current status of the Grid middleware and assess what has to be done by the collaboration in order to make a smooth transition to the Grid tools. Therefore ATLAS has been extremely active in Grid matters since mid-2002. A significant fraction of DC1 was performed in the Grid environment (NorduGrid, USGrid), involving about 20 sites and several flavours of Grid middleware. Members of the ATLAS DC Team also participated in a task force to test EDG middleware on a dedicated test-bed, and provided valuable feedback to EDG developers.

All the data processing of the Nordic Countries was done on the NorduGrid[43], and the whole of Dataset 2003. The test-bed included 8 Linux clusters across Scandinavia and Finland. Despite having different operating

[43] http://www.nordugrid.org

systems and hardware characteristics, the clusters performed as a single farm, having jobs distributed in optimal way, and writing the output onto a dedicated storage area at Oslo University. A detailed report can be found on the web[44]. Three sites of the US Grid test-bed took part in DC1 phase 1. About 10% of the US contribution was done using this test-bed.

In phase 2 all the Nordic production was done on NorduGrid. On the US 6000 jobs for the pile-up exercise (DC1 phase 2) have been run corresponding to ~3000 NCU-days and 10 TBytes of data being used (input and output). For both the US and NorduGrid test-beds have been intensively used for the reconstruction step.

On the European side an ATLAS-EDG task force was put in place in August 2002. A first test was successfully run in September 2002 followed by other tests at different periods. The work of the ATLAS-EDG task force in 2003 is described in 13.3.

13.1 DC1 Production on NorduGrid[45]

The aims of the NorduGrid project have been, from the start, to built and operate a production Grid in Scandinavia and Finland. The project was started in May 2001 and has been running a testbed since May 2002. Taking advantage of the existence of the NorduGrid testbed and tools, physicists from Scandinavia and Finland were able to participate in the overall exercise using solely the NorduGrid environment. During the whole DC1, more than 2 TB of input data was processed and more than 2.5 TB of output data was produced by more than 4750 Grid jobs.

The NorduGrid resources range from the original small test-clusters at the different physics-institutions to some of the biggest supercomputer clusters in the region. It is one of the largest operational Grids in the world with approximately 1000 CPU`s available 24 hours a day, 7 days a week. It is, however, not exclusively dedicated to ATLAS.

In Phase 1 of DC1, all the input files were pre-staged (replicated) at all the sites and output files were stored at a designed Storage Element.

During Phase 2, those output files, together with files containing minimum bias events, were the input for the pile-up production. Therefore, pre-staging, as in phase 1, was unfeasible, and so the Grid Manager had to download input files for each job. However, to optimise the task, “minimum bias” files were pre-staged at several sites, sometimes only partially (i.e. not the entire set). Thus, whenever an input file (containing either signal or minimum bias events) was missing for a specific job, the Grid Manager would proceed to download it and cache it for potential use by another job. This caching was particularly convenient for “minimum bias” files, as they were often re-used by several jobs.

Phase 3 followed the same scheme as Phase 2, except that there were no “minimum bias” files to be pre-staged.

It is worth mentioning that part of the NorduGrid success was due to the RPM installation of the ATLAS software releases, different from the by then standard “build-in-place” structure. The approach to group binaries, libraries etc. “Linux-style” was adopted by CMT via the “install area” and is now widely accepted as the production installation. NorduGrid RPMs are used by the US Grid via PACMAN.

NorduGrid has contributed substantially in all 3 Phases. Important lessons about the NorduGrid middleware have been learned during this production periods, which have been used to extend the stability, flexibility and functionality of the software and NorduGrid itself.

13.2 DC1 Production on the US Test-Bed[46]

[44] http://www.nordugrid.org/documens/ATLASdc1.html
"Building a Production Grid in Scandinavia". P.Eerola et al., IEEE Internet Computing, 2003, vol.7, issue 4, pp.27-35.
"The NorduGrid architecture and tools". P.Eerola et al., in Proceedings of CHEP 2003.
"Atlas Data-Challenge 1 on NorduGrid". P.Eerola et al., in Proceedings of CHEP 2003.
[45] More details: See ATLAS Note : ATL-SOFT-2003-002
[46] More details: See paper “DC1 Production in the U.S.; in preparation

DC1 production in the U.S. was carried out using both batch and grid facilities. Batch processing was done at the regional Tier 1 centre, Brookhaven National Laboratory (BNL). Grid processing took place in the U.S. ATLAS grid testbed, a widely distributed computational grid comprising of eleven institutions. During Phase 1, the average usage was 40-60 nodes, which grew to 200 nodes by the end of DC1. The generated data occupied 10TB disk, and 20TB HPSS tape storage at BNL.

The grid testbed became available after a few months of batch production. A special tarball of the GEANT executables was made for the grid, containing binaries only for Red Hat Linux. The executables were installed on Globus gatekeeper machines at 3 U.S. testbed sites

A grid scheduler was used to submit the jobs and had an 80% success rate - most failures happened due to hardware and software failures or scheduled outages. The submission process was completely automatic and required very little supervision or intervention. In most cases of a site being unavailable, the scheduler continued production with the other available sites without problem).

Each production job on the grid had many stages. First the Globus gatekeeper of the site selected by the scheduler is queried for software location information. Next, a suitable available partition is chosen for production. The proposed logical filename (LFN) is registered in MAGDA along with various production related information. All executables are staged into a temporary location. A script with the location of the executables and environment variables is sent to the queue on the selected site. The job is started asynchronously by the batch queue system. The scheduler checks every 5 minutes if the production job has finished. Once it finishes (on average after 14 hours), the files are moved to the BNL HPSS tape storage system by MAGDA. All LFNs are registered in the MAGDA catalogue. A replica is also made by MAGDA at one of the available grid sites.

An independent semi-automatic quality of service (QOS) process is run periodically. This job checks the MAGDA production database for the job status of every partition (the production job updates this database periodically during staging and execution). It checks the job status on the submitted queue through Globus. It verifies through MAGDA that all files are correctly stored in the HPSS and replica locations. It checks if the temporary staging location has been cleaned up after production. This process can correct for many failures and updates the production database if it can recover files. For example, the BNL HPSS was unavailable for a couple of days - production continued without any changes. When HPSS was available again, the QOS process automatically copied and catalogued all primary files from the replicas using MAGDA.

Most of the problems during the 2 weeks of production were typical of distributed systems spread out over 4 locations thousands of miles apart (New York, California, Texas and Oklahoma). Various machines were not available at critical times. Even when empty queues were available, however we could not run production faster than about 70-80 jobs per day at any one site. After some tuning of the production, this limitation was eliminated for Phase 2 and all U.S. pile-up production was done on the grid.

13.3. DC1 Production using the EDG[47] testbed

Beyond the tests held during autumn/winter 2002 (~400 DC1 simulation jobs on EDG application testbed version 1.2), 250 reconstruction jobs (4 datasets, previously simulated both at high and low luminosity) have been processed in spring 2003 on the EDG production testbed, using an improved version of the EDG middleware (version 1.4).

The reconstruction ATLAS software (6.0.4) required RH 7.3, which was not yet officially supported by the EDG middleware. Additional work had to be done to create new LCFG profiles to install both the operating system RH 7.3 and the EDG software on the Worker Nodes.

The input data have been copied on the Storage Elements of the involved sites, distributed among Italy (Milan, Rome and CNAF), France (Lyon) and the UK (Cambridge). The job submission has been as transparent as possible, specifying only the required input file and the job type.

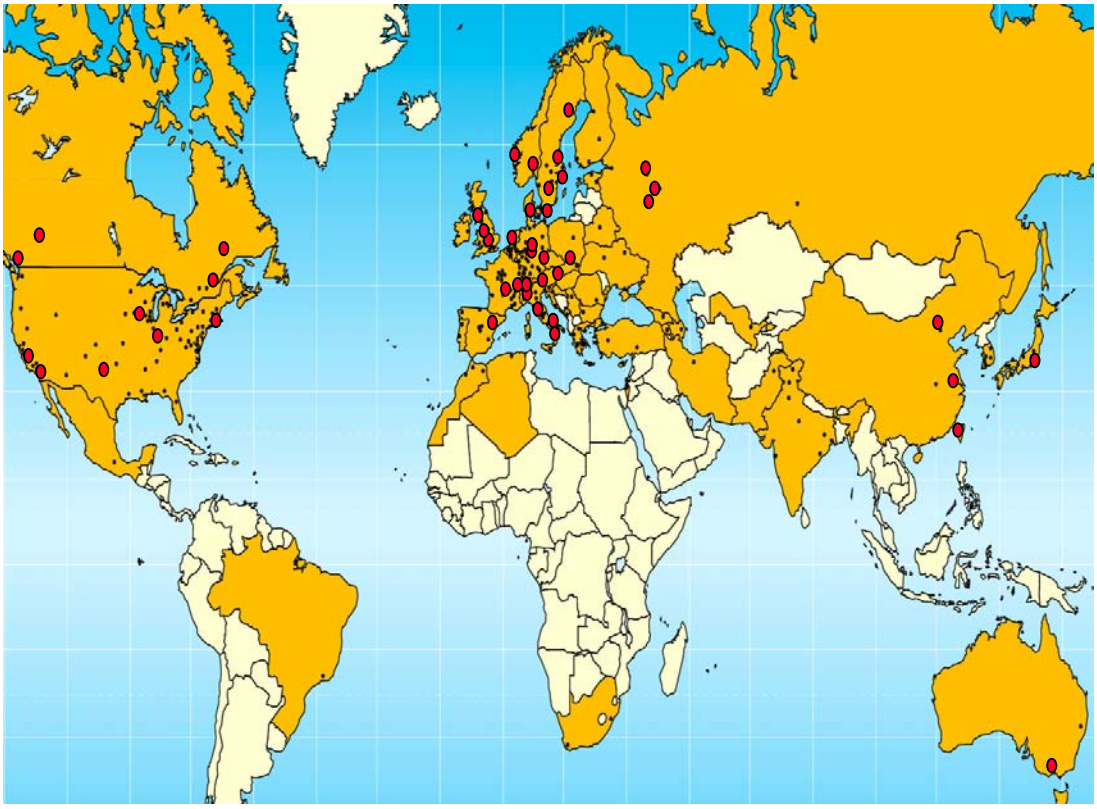
[47] http://eu-datagrid.web.cern.ch/eu-datagrid/

The task of the matchmaking of the resources has been assigned to the EDG Resource Broker (RB), which performed it successfully. The RB and the whole EDG middleware have shown good stability over a period of about 2 weeks, requiring only few slight interventions of the site managers.

It has, however, to be noticed that this mini-production did not constitute a stress test: the ATLAS job rate was modest and only few activities from other users were going on in parallel. It has, however, demonstrated that the EDG s/w is actually well capable of handling ATLAS production jobs.

Although the ATLAS was not of a scale to verify the stability and the scalability of the middleware in case of a real huge production, it has provided evidence that the EDG performances were greatly improved in the last few months and many the problems previously spotted by ATLAS were solved. As soon as available, the ATLAS collaboration will perform intensive tests using the LCG testbed (LCG-1), which has the EDG 2.0 as major component.

14.) Conclusions



around the world → around the clock

Fig. 10 Map of the sites taking part in the ATLAS DC1 activities

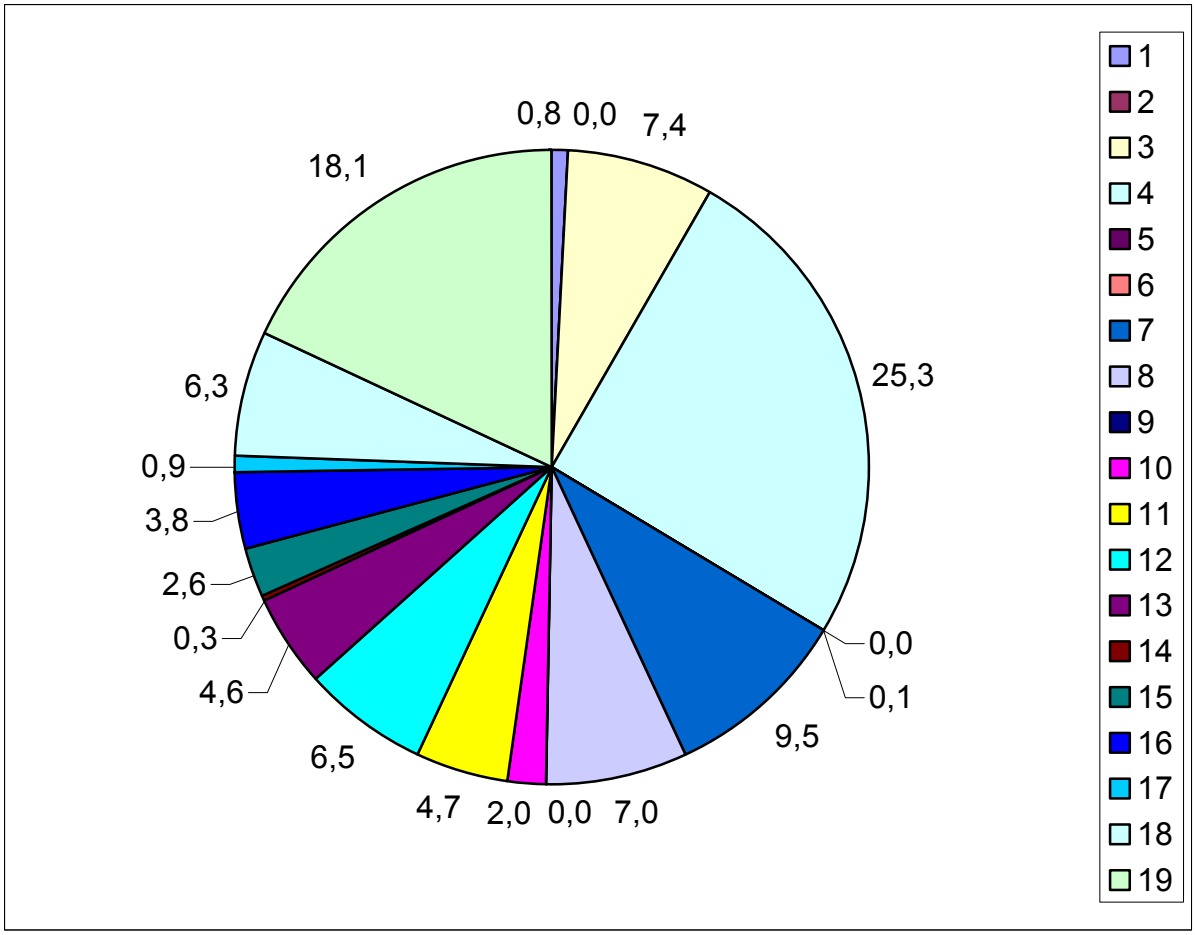


Fig. 1 Contribution per country to the overall CPU-time (%) in DC1.
The numbering corresponds to the one in 11.1

ATLAS Data Challenge 1 ran from spring 2002 to spring 2003. For several reasons it was divided in several phases.

Phase 1 was used to put in place the worldwide production infrastructures and to produce the bulk of simulated data needed by our colleagues of the High Level Trigger for their Technical Design Report. Over a period of 40 calendar days the equivalent of 1.5 million of SI95-days were used to produce 10 million physics events and 40 million of single particle events for a total volume of 30 TBytes. The success of a worldwide exercise of this scale certainly exceeded our most optimistic expectations. 40 institutes in 19 countries actively participated to the effort.

The pile-up production in the second phase ran smoothly. 17000 NCU-days were necessary to produce about 34 TBytes of data.

Most of that data has already been reconstructed. A large fraction of the data has been reconstructed in offline and/or trigger reconstruction mode. 23000 NCU-days were necessary to produce about 200 Gbytes of data.

The numbers for all 3 phases together are approximately:

- 120000 NCU-days
- 70 Tbytes produced
- 100000 partitions

During that exercise we have seen the emergence of the production on the Grid. Grid tools were used intensively on NorduGrid and US test-beds. We are confident that their use will continue to grow.

ATLAS DC1 has proved to be a very fruitful and useful enterprise, with lots of experience gained, providing feedback and triggering lots of interactions between various different groups, for example groups involved in ATLAS Computing (e.g., HLT, offline-software developers, Physics Group), Grid middleware developers, and CERN IT. There can now be every for confidence that ATLAS will be able to marshal worldwide resources in an effective way.

Finally, perhaps the most important benefits of DC1 have been to establish a very good collaborative spirit between all members of the DC team and to increase the momentum of the ATLAS computing as a whole.

Acknowledgments

We would like to thank all members of the ATLAS collaboration who participated to the effort. It would not have been possible to run successfully our Data Challenge without the involvement of numerous people from many institutes and computing centres who helped us to put in place the production chain and to run the production. We thank all of them most sincerely.

.

Appendix A: Pythia Parameters used for the event generation

The default Pythia parameters were used with the following exceptions:

The multiple interaction model is used for underlying and minimum bias events MSTP (82)=4 and PARP (82)=2.2, since this gives better agreement with the CDF data[48]. In addition the fragmentation parameters were set to: MSTJ (11)=3, PARJ (54)=-0.07 and PARJ (55)=-0.006. MSTJ (22)=2 was used to ensure stable K^0 and Λ.

The following samples were generated:

A **"jet" sample** with the following processes activated; 1,2,11,12,13,28,53,68,81,82,14 and 29. No kinematics' cuts were applied, except for a minimum transverse momentum CKIN (3), whose value can be found in the full list of events. The largest sample was generated with CKIN (3) =17. This sample is dominated by the $2 \rightarrow 2$ QCD processes such as $gg \rightarrow gg$.

A **"minimum bias"** sample with MSEL=1. These events are used for pile-up for which it is recommended that the cross section should be set to 67 mb[49]. To generate minimum bias events for DC1 the PYTHIA generator version 6.203 is used. The parameters used for the event generation represent the best tuning for energies up to the Tevatron energies.

Single W sample: Inclusive W production using process 2. For the sample, where W is forced to decay to $\tau\nu$, the kinematics' range is restricted by setting CKIN (1)=71 and CKIN (2)=91. For the sample, where W is forced to decay to $e\nu$, only CKIN (1)=71 is used.

Single Z samples: Three sets forcing the Z to decay to e^+e^-, $\mu^+\mu^-$ and $\tau^+\tau^-$ were generated. Process 1 was activated with MSTP (43)=2 so that only the *Z* boson contributes. To improve efficiency, CKIN (1)=81 and CKIN (2)=101 was used.

W+jet samples: Processes 16 and 31 were activated and CKIN (3)=100 was used to force the generation of events at high transverse momentum. The W bosons were forced to decay leptonically.

Z+jet samples: MSEL=13 was used with MSTP (43)=2 to turn off contributions from virtual photons. The Z bosons were forced to decay leptonically. (Note that the sample is not identical to the one generated for Data Challenge 0, as the parameters and version of Pythia are different.)

Photon+jet sample: MSEL=10 was used with CKIN(3)=100.

Inclusive top sample: Processes 81 and 82 were used with MSTP (7)=6 to force the production of top quark final states. The decays are unbiased.

Higgs Samples: Inclusive Higgs production, using processes 102, 123 and 124, was generated for Higgs masses of 120 and 130 GeV/c², respectively. The former is forced to decay to $\gamma\gamma$ and the latter to four leptons which can be either e or μ.
The WH process (26) was used to generate events with Higgs masses of 120 and 400 GeV/c². The W was forced to decay to $\mu\nu$ and the Higgs to one of bb, uu, cc or gg thus making eight sets in all.
A separate ttH production using processes 121 and 122 was made; the H, with a mass of 120 GeV/c², was forced to decay to bb, One W was forced to decay leptonically to $e\nu$ or $\mu\nu$ and the other to jets. Only one sign of leptons is generated.

[48] A Moraes, I. Dawson and C. Buttar, ATLAS note in preparation
[49] ref to discussions in the MC4LHC group

MSSM Higgs Samples: These were generated, using a SUGRA model so that sensible widths are obtained; IMSS (1)=2, RMSS (4)=1, RMSS (5)=39, RMSS (1)=212, RMSS (16)=0 and RMSS (8)=483.4. The H has a total width of 10.3 GeV and a mass of 400 GeV/c². Note that all the H (400) cases correspond to the same model. For an H mass of 300 GeV/c² the following parameters are used: IMSS (1)=2, RMSS (4)=1, RMSS (5)=42, RMSS (1)=144, RMSS (16)=0 and RMSS (8)=436.
Processes 152, 173 and 174* *were used to generate the final state H $\rightarrow$ hh and the h is forced to decay, so that samples with bb bb, uu bb are produced for the 400 GeV/c² Mass; only the bb bb state is produced for 300 GeV/c². (3 samples in all)
Processes 181 and 182 with KFPR (121,2)=5 and KFPR (121,2)=5 were used to make the bb H final state; H is forced to decay either to bb or uu for the 400 GeV/c² mass case.

b-physics Samples: The ATHENA b-physics generator, PythiaB, uses a dedicated set of parameters tuned to Fermilab and LEP data. The first set are the b-production related parameters:

pypars mstp 51 1 (CTEQ3)
pysubs msel 1
max parton virtuality factor : pypars parp 67 1
the factorization scale : $Q^2_{hard} = p_t^2 (P_1^2 + P_2^2 + m_3^2 + m_4^2)/2$
pypars mstp 32 8

The second set of parameters determine the properties of the b-hadrons:

spin probabilities: pydat1 parj 13 0.65
pydat1 parj 14 0.12
pydat1 parj 15 0.04
pydat1 parj 16 0.12
pydat1 parj 17 0.2

and the Peterson fragmentation parameter ε_b

pydat1 parj 55 -.006

The complete description of the parameters and the method of the event generation in PythiaB can be found in an ATLAS note[50].

[50] ATL-COM-PHYS-2003-038

Appendix B: Production Reports From Different Sites

B.1 Australia

a.) Software installation:

Prior to DC1 Phase 1, we were trying to install the software via the CERN /afs tree and this was extremely painful and time-consuming since there appeared to be so many assumptions made that the software was running on a CERN node and not at a remote site. In addition it wasn't clear which parts of the tree we needed to copy so we ended up copying a lot more than necessary, which was very slow given the speed of our international links from Australia. And there was a complete lack of basic documentation explaining how to install and run the software.

However, in time the RPMs from INFN became available and after that installation became far, far easier for us. We run the CERN release of RedHat Linux on our cluster since we thought that would make things easier for us in the long run. The RPMs install onto this with no difficulty in a very short amount of time, and since we make heavy use of NFS we only needed to install the RPMs on one node and the software was then instantly available to all nodes in our cluster.

Once the RPMs were installed the biggest hurdle was the front-end or ``prodscript'' as it's known. This still required considerable customisation since the default one is very CERN-centric. I also found it a bit frustrating that the prodscript was not really part of the ``release'' - even though it's an absolutely vital component. A version of this script was provided with the RPMs, but there was also a version available from the CERN /afs tree in someone's home directory (no less) with an obscure name like ``prodscript4'' which most people seemed to be using! I found this very confusing and wasn't sure which one should be used. The only reference on the DC1 details screen simply said “prodscript needs updating ...”.

Basic documentation was still a little lacking. The DC1 ``details'' page was extremely useful but needed to be more comprehensive and tackle things from start to finish. A physicist working at CERN can get most of what he needs from it to set-upset-up and run the simulation, but if you are not working at CERN and/or you're a computer scientist then it would be very, very hard to set-up and run the simulation. I suppose the hope is that use of the grid for Phase 2 will largely solve this problem?

b.) Problems during the data processing:

Most of our problems were the same as those experienced at other sites I imagine - the bugs discovered along the way required us to reinstall software and restart the simulation from scratch. One job failed due to the random number problem and had to be restarted with a new seed. Other jobs needed a restart because I made an error editing the KUMAC in the prodscript.

The only other problem then was just monitoring the jobs for success or failure and resubmitting where required. prodscript does not seem to exit with a failure code when things go wrong so the only way to be sure the job worked is to eyeball the logfiles and there can be quite a lot of those!

c.) Problems getting data to/from CERN

No real problems as such here. We made heavy use of bbftp, which sped things up quite a bit. Since we are at the end of a (relatively) slow international link it took quite a while to send/receive data. Uploading results typically took 1-2 days per dataset (we were seeing around 600KB/sec xfer rate with bbftp to castor).

Sometimes transfers would fail part way through but bbftp will auto-restart from where it left off and resume so that wasn't a problem.

One comment I will make on this point however is that I think that in future perhaps it would be good to provide checksums/md5s of input files (either on a web page somewhere or alongside the file on castor with a .sum or

.md5 extension). During one early trial run we spent quite some time trying to figure out why our job kept crashing only to discover we had a corrupt input file. A simple checksum comparison would have saved us a lot of time in this case.

B.2 Canada

a) Software installation

For the DC1 Phase 1 the ATLAS software was installed on the two main Alberta clusters using CMT and the cvsupd daemon to download the required software packages for the release.

This procedure was successfully started some time ago with release 3.0.0 and included all releases up to and including 3.2.1. Future releases will also likely be installed in this fashion. I find this method more flexible than the RPM based method. The goal of the installation at Alberta with cvsupd is to provide the user with an environment, which allows for code development in a fashion similar to what is found at CERN.

The RPM method was used to install the ATLAS software on two other clusters at the University of Alberta to which I did not have root access. The fact that the RPM files required root access to be installed was a problem. Furthermore, the RPM files were not relocatable to anywhere other than /opt.

Therefore the software installation using the RPM files proceeded by first unpacking the RPM files onto a system, which I controlled, and subsequently creating a tar file to distribute to the other machines. With the tar file I could place the ATLAS software where needed without root intervention.

b.) Problems during the data processing:

Difficulties encountered included the occasional occurrence of CALOR errors requiring rerunning with a different seed. Overall the number of problems encountered was small, most of which being hardware in nature.

On a couple of occasions difficulties were had was with one of the raid servers used (failed disk drive causing system reset, and then having to be replaced, kernel panic due to Ethernet interface troubles). Two brief slowdowns in production also occurred in order to add and then remove a temporary RAID array to one server used for testing. Otherwise productions have gone smoothly.

c.) Problems getting data to/from CERN

All remote access to CERN for the DC1 productions was through wacdr.cern.ch to download the necessary EVGEN partition files. This was done early on in the DC1 Phase 1 period. At that time only regular ftp was used for the downloads. Protocols such as bbftp will be used for future downloads.

Other than the files from dataset 002000 for validation all files are stored locally at Alberta and were not shipped to CERN.

B.3 CERN

CERN DC1 experience for non-single particles production

After a period of tests the production started for real on Friday July 12. This report covers the period up to Friday August 23, i.e. 43 days.

Allocated computing capacity

The original plan was to run the production over a period of three weeks on a set of 200 dedicated CPUs (part of the LXSHARE cluster). However, during spring CERN-IT decided to change its cluster deployment model from one with many sub-clusters dedicated to single experiments, to one with a single cluster shared by all experiments. At the same time the scheduling scheme used by the LSF scheduler would be changed from ‘first come first served’ combined with priorities/pre-emption, to a fair-share scheme.

In theory the new plan should have been simple. Given a cluster of about 1500 CPUs, during a period of three weeks ATLAS' share should be increased with a percentage equivalent to 200 CPUs (13%). Reality was quite different.

To start with ATLAS's share was not adjusted until beginning of August. Fortunately, the shares only come into the game when the cluster is full. This was the case only during one out of five days (rough estimate). The other days the production capacity was limited by an artificial 400 job (i.e. 266 CPU) limit, preventing ATLAS to grab all free processors and hold them for a longer period. As a consequence, during the first weeks of production our capacity was going up and down between 50 and 400 jobs, depending on the overall load on the cluster. This was quite annoying especially because at that time we did not understand the reason why this happened.

During the first week of production CERN-IT upgraded their scheduling software from LSF 3 to LSF 4. In this transition period ATLAS had access to two parallel clusters with production sized quota on both. As a result, we were able to run as much as 700 jobs in parallel during a short period.

Two other consequences of abandoning the dedicated cluster scheme are worth mentioning. Firstly, as the production jobs were running on machines shared with other users, they occasionally failed because of misbehaviour (memory or disk space) of these other users. Secondly, the slowest job could take up to three times as long as the fastest job. This is the combined effect of a factor two in raw speed and a factor 1.5 depending upon whether your job needs to share the CPU with a third job or not.

Given that, due to a high failure rate discussed in the next section, often up to three re-submission rounds were needed to finish a large group of related jobs (a dataset), the production of every dataset was spread out over a much longer period in time than expected, leading to a situation that when the production of a new dataset was started, more than 5 others were still not finished.

In hindsight it would have probably been much easier to do this production on a dedicated cluster. On the other hand, on average we managed to use close to 200 CPUs during six weeks, i.e. we did practically twice what we planned to do. Another major advantage is that when for some reason we could not have jobs running, the CPU resources were not necessarily lost. This was in fact one of the main motivations for introducing the new cluster deployment model in the first place.

Experienced problems

The time period over which the production took place, especially the first half of it, was probably the worst possible time of the year. During the first three weeks there was one major (in the sense that hundreds of jobs failed all over the cluster) incident during every weekend and one during every working week: AFS token problems, AFS file servers problems, general network problems etc. Failure rates peeked up to 40% and averaged at 20%. Earlier test productions enjoyed failure rates as low as 0% and consequently our production machinery did not foresee extensive automatic recovery. The manual recovery and cleaning up soon became a full time job for one person.

While 75% of the job failures happened at concentrated times (the major incidents), there were still 25% of continuous failures. Two changes were made to the production scripts in an attempt to remove these. The first version of the simulation script read its input directly from the CERN Castor mass storage system. It was suggested that this was the cause of the many random failures and consequently the script was modified to first copy its input to local disk. This did remove the few errors that were clearly related to time-outs on input but it did not seem to change the much higher rate of segmentation faults, bus errors, abrupt terminations, etc.

Triggered by the observation that in case of network/AFS problems many jobs fail in the middle of computation, whereas in principle they should be totally independent of the network at that time, a small investigation showed that indeed the running jobs have several tens of files on AFS open during execution. Besides the core executable and a limited number of shared libraries about 75% of these files were related to the ROOT package.

Modifying the script to first copy its executable and shared libraries and perform a complete local installation of the ROOT package, made failure rates drop to less than 2%. It is possible that the removed AFS dependencies were the cause of the high failure rates, but part of the improvement could also be due to a decreased incident rate on the cluster in general, which seems to be the case as well. Conclusive evidence would require deploying the two versions of the script at the same time. This does not seem to be worth the effort at present.

Moving from reading the input from Castor directly, to making a local copy first, introduced a potential IO problem. When 400 jobs start simultaneously by copying a 2GB file one might expect problems. Given our

many other problems, we did not even attempt to test this. In practice only the first few datasets had 2GB input files, all others had more modest sizes up to 400MB. Additionally, we took care not to start too many readers of the same file at the same time (<50) and not to start too many readers in total at the same time (<200). With these precautions there was no significant contribution to the overall failure rate. We did not test whether these precautions were indeed necessary, or whether higher limits could have been used.

We conclude this section with a few words on the generator production:

Although all events were generated at CERN, not all of them were generated the same way. Two different production strategies were deployed. We report only on the second one used for all but 4 very high statistics datasets. Because the generator software runs within the ATLAS control framework (Athena) running jobs in batch mode is quite cumbersome. The current procedure for running ATHENA jobs requires you to check out a special Test-Release package from CVS and 'install' it relying upon the ATLAS release tool CMT. This installation sets up your environment variables, creates a multitude of links to executables and shared libraries, and copies a multitude of files locally from the release. In principle, all this can be done from a batch job as well but this is clearly less than ideal.

An alternative strategy is to perform this installation step once in some shared file space and have it reused by the many batch jobs (this is the approach used in strategy one). The approach we exercised requires no CMT based installation phase at all. Instead, one script does everything using the parts it needs from the release directly. Figuring out what these parts were was not easy, as the out-of-the-box set-up links to just about everything in the complete ATLAS release and to many things even three or four times. The script reduces the links from several hundreds to about seventy, some of those probably not needed. One obvious disadvantage of this trial and error approach is that the script can only run the jobs it was intended to run, and perhaps not even all of those.

It is extremely desirable that ATLAS invests some effort in providing reasonable installation support for its software suite, and that software authors specify exactly what their software needs to run instead of using a blind wildcard.

Single Particle and GENZ/Legacy production

A script was prepared to run the SPGUN (single particle gun) work in the zshell as the example script for the first job was in zsh. Then a job submission script was written to process the "one line per sample" file to generate however many jobs were needed. More or less the same was done for the GENZ/legacy jobs.

All of the bookkeeping needs are completed in the job (or at job submission for the information common to each partition in a dataset). There are at least two ways to keep information, so AMI and MAGDA files are written and "cat-ed" into the log, there are some recovery attempts for copy from the batch job to castor and so on.
At CERN MAGDA just registers the file. They are not yet processed.

The job submission logs each job as it is submitted and gives it a unique id appended to the log file name. In the event that the job actually runs and the log file can be found, where it was requested, in $HOME/LSFJOB... or on the mail server the log file can be inspected and moved into the medium term AFS location before eventual tar archive of the whole directory for that run.

Both the dataset id and the partition number were used for each job to generate the random seed and the parameter string for the script is stored for AMI. A further parameter was added, which is a further digit to add to the seed generation, facilitating a retry for jobs failing with code bugs rather than system errors.

For the legacy GENZ data an appropriate job/script was produced and there is a "pre-processing step".

http://soneale.home.cern.ch/s/soneale/www/GROUPS/SOFTWARE/DOCUMENTS/UTILITIES /atlsffan.html
http://soneale.home.cern.ch/s/soneale/www/GROUPS/SOFTWARE/DOCUMENTS/UTILITIES/atlsfcat.html

The atlsffan program was used which takes (say) a GENZ file and divides it up into "job size" files and sets sensible run event numbers in all locations - except for the GENZ bank. By setting a very large number of events per partition the complete file can be processed and the output is stored in castor with our newly allocated run number and our conventions for (logical) file names.

B.4 France

a) Software installation

The ATLAS software was installed on the Linux CCin2p3 clusters using CMT and the cvsupd daemon to download the required software packages for the release[51]. Part of the production (b-tag data samples) was done using the VDC (Virtual Data Catalogue) for storing and retrieving the production job options. The VDC was based on the NOVA MySQL database operated at CERN site.

Some difficulties were encountered due to the lack of information on the needed external libraries. There is no standard ATLAS procedure to deal with external libraries and ONE IS URGENTLY NEEDED. It will be important to publish the list of changes from a release to another one (external libraries like BOOST, ROOT, ANAPHE, CLHEP, G4 and so many others... are changing versions at each release). Therefore an automatic cvsup procedure would help but only for the external libraries handled by ATLAS. Those requiring the action of system administrators cannot be dealt with automatically. However the adequate information is needed.

b.) Problems during the data processing:

No problems have been encountered during the data processing. We have been using the BQS batch queuing system.

c.) Problems getting data to/from CERN

No problems have been encountered during the data transfer To/From CERN. We have used bbftp through wacdr.cern.ch (CCin2p3/HPSS<->CERN/CASTOR).

B.5 Germany

a) Software installation

Since a while we had been carrying out local ATLAS software builds in Munich and Karlsruhe, using the cvsup tool to transfer the software to the respective sites and CMT to build them. The software worked well and showed in the preliminary random-number-comparison-tests that it performs s just as the RPM distributions from Italy. Those were not really an option for us since getting super-user rights at FZK, where 8 HEP experiments share a computing environment is not foreseen.

On a side note: Although everybody seemed to be happy with the RPMs, I do not understand why we had to use them: if the goal is to distribute the necessary executables and libraries only as opposed to set-up a full blown ATLAS software development environment, why not really ONLY distribute those needed files? Especially since CMT makes it easy to identify those components: you only need to look in the build directory, usually 'Linux-gcc-opt'. I tried it out and produced an 18 MB (uncompressed) directory, which contained ATLSIM, and every thing needed, and it passed the above-mentioned test as well.

b.) Problems during the data processing:

We have been running a little less than 3000 jobs a 200-500 events (~24-48 h) and only one single job terminated for unknown reasons. The rest of the 'problems' consists of partitions which crashed due to the

"CALOR: FATAL ERROR IN EVAP"

[51] http://isnwww.in2p3.fr/atlas/fairouz/dc/dc1.html

error, which occurred in about 100 of the 3000 jobs. Those jobs had to be rerun with a different random number seed and finished all successfully the second time.

c.) Problems getting data to/from CERN

After a long and difficult installation procedure of bbftp at our site, data transfer of the root input partitions has been working smoothly ever since.

B.6 Israel

a) Software installation

We had the Release 3.2.1 software already installed at our site. Nevertheless, we have installed the right RPM distribution also. But we had to have super-user privileges to install the RPMs, due to the fact that paths were absolute and not relative. Then, moving the software in the desired location required fixing the symbolic links, which unfortunately are again absolute and not relative as they should be.

b.) Problems during the data processing:

Only one ZEBRA file had to be rerun with another RANLUX parameter to avoid the CALOR STOP problem.

c.) Problems getting data to/from CERN

Downloading the root files was not easy, we had to do it in two steps: first, an rfcp from castor to /tmp on a machine at CERN, then an rsync from our site.

B.7 Italy

a) Software installation

All the sites have installed the software via the RPMs, without any big problem. In two sites (CNAF and Rome1) the installations have been also done using LCFG on EDG machines.

Just one comment for the RPMs: many people complained about the non-relocability of the packages. This issue has been now (probably) solved and the new relocatable RPMs are available at the same place where they were before (I just overwrote the old ones)[52].

This should also solve the problems when the user doesn't have the root password, since the whole kit may be relocated in a directory whose owner is not root. If somebody needs more instructions, please let me know and I'll post them in the list.

b.) Problems during the data processing:

No big problems. Single muon production has not showed any apparent problem, while for the rest only < 2% of the jobs have been resubmitted after a fatal stop (CALOR: Error in EVAP--> STOP). We have anyway to think about a new strategy for such kind of errors, since to change the RANLUX by hand for each job, when the error occurs, is probably not the best solution.

c.) Problems getting data to/from CERN

[52] https://classis01.roma1.infn.it/ATLAS-farm/ATLAS-kit/3.2.1-2.

We had two kinds of approaches:

- Roma1: data were copied on LXPLUS, via rfio, and then transferred to the local farm via scp and vice versa;
- The Rest: data were copied directly to/from the local site via ftp using wacdr.

For the farm in Rome1 we had to execute that "double step" since the ftp protocol has been disabled in both directions for security reasons. There were no particular problems in transferring data to/from CERN, expect that sometimes the transfer was really slow, but it does depend on the connection and was solved, in some cases, by transferring using multiple streams.

B.8 Japan

a) Software installation

We didn't see any problem in installing the software with the RPM kit. It was easy and straightforward because the farm is dedicated only to ATLAS and the root user was in our group and working closely. It would have been nicer if we could decide the directory to which the software is to be installed.

Although, there is a worry to have an RPM kit in addition to another installation. Before the RPM installation, we have 'copied' the ATLAS releases to our system. We didn't see any interference between these copied software and the RPM one this time, probably the creator of the kit must have been very careful, but it is worrying to have two trees of the same software, especially when the two can be different versions.

A good feature of the RPM kit was that it was 'frozen'. During the DC1-0, we had experienced that the software in the copied tree didn't produce the same results as the other sites. This was caused by changes in the release under /afs/cern.ch after we copied the files and also by building some binaries (libraries) locally. Building the libraries at CERN and then copying them to Tokyo solved the problem. (And also by copying the up-to-date source files once the difference was confirmed)

b.) Problems during the data processing:

No known big 'problems', but some comments. It was, however, worrisome and tiring that we had to make changes to the job scripts for site specific parameters every time we get a new script because we can overlook something and can make mistakes, and had to repeat the same things. It would be nice to have site-dependent part separately (even in the same script file).

No recipe to check production results was provided at first. One had to check the log files without knowing what to look for. The errors reported to the ML and what to do with them were accumulated quickly, and it was useful. But it would be nicer to have the information on the web page. It would also be nice to have a checklist and even a script to check log, data size, and so on.

The ones who execute the jobs not experts, and a few experts (or a single one) cannot look after all the productions.

Rerunning with a different random number seed added some complexity to the process. Although, this was understandable since the situation had not been foreseen.
We need to reconsider how to treat this for the future productions.

Some troubles occurred mainly due to the fact that our farm was really new and not yet well configured.

c.) Problems getting data to/from CERN

We transferred the data using rfcp+scp, rfcp between CASTOR and CERN machines and scp between CERN and Tokyo.
No problem was seen. Although the transfer was slow, I copied multiple files in parallel, and both downloading and uploading were finished in several hours.

One last comment:

It was a good exercise for our newly built pc-farm + batch system.

B.9 NorduGrid

NorduGrid has made as comparison of different processors. They found some differences between Pentium and Athlon processors (Athlon processors tend to be less efficient.)

a.) Software installation:

ATLAS software was rebuilt at most sites; hence the pseudorandom number sequences may diverge from the "standard" (a la CERN) ones.

b.) Problems during the data processing:

Problems: few jobs crashed (mostly "ZEBRA banks screwed up"), but were successfully re-run with the altered random seed (+1000000)

B.10 Poland

The Krakow cluster is part of CrossGrid[53] testbed. The resources are shared with other CrossGrid applications e.g. weather forecasting. The CrossGrid Project closely collaborates with EDG.

a.) Software installation:

The installation is based on RPM's taken from official ATLAS RPM kit page. Because of the version of the operating system (RH Linux 6.1) we had problems to install ATLAS distributions based on RH 7.3 CERN Linux distribution. After some tricks the installations of RH 7.3 was possible on Worker Nodes.

b.) Problems during the data processing:

No serious problems were encountered during the data processing. Grid (GLOBUS) based jobs submission introduced technical problems like standard output retrieval etc.

c.) Problems getting data to/from CERN

The main problem was to take data from CERN and copy it back. Standard FTP transfer to CASTOR was rarely successful because of timeout, therefore bbftp was used as transfer tool. The data transfer required many pre-staging steps using bbftp and later GLOBUS grid-url-copy.

B.11 Russia

a.) Software installation:

We are using the "rsync" copy of the full /afs/cern.ch/ATLAS/software/dist/3.2.1 tree. It is linked to the faked local directory /afs/cern.ch/... The same is done for general CERN libraries and /afs/cern.ch/ATLAS/offline. The production scripts are CERN-oriented. It would be good to make them more flexible from the very beginning and single out site-independent core.

b.) Problems during the data processing:

[53] www.crossgrid.org

Some lack of information at the first stage (hard to follow "official" line if you are not present at last DC meeting). Improved with new DC-page, but need some more efforts. In my opinion it would be nice to have a PBS version of production scripts (I think most of sites use this batch system)

The main problems were job interruptions in GCALOR

CALOR: Fatal ERROR in EVAP ====> STOP

requiring rerun with changed RUNLUX. This error is not detected automatically in the script; so, it requires manual job results monitoring. The solution is hardly acceptable for mass simulation and needs to be changed.

In addition we had some minor problems since several input EVGEN files were corrupted and they were regenerated. Several jobs were interrupted with diagnostics "segmentation fault" but they were caused by H/W problems at one of our CPUs.

.

c.) Problems getting data to/from CERN

Some weird problem with my 1701-1720 partitions on castor, since files were not overwritten by newer version; this was solved by deleting the old version and copying once again. Otherwise there were no real problems since the rather moderate volume of transferred data. We used ftp to/from wacdr.cern.ch.

B.12 Spain

a) Software installation

The RPM format was a great advance from previous distribution formats. It would be desirable to have it relocatable.

b.) Problems during the data processing:

1 of 400 jobs in partition 002000 gave CALOR: Error in EVAP--> STOP
9 of 1000 jobs in partition 002030 gave the same

c.) Problems getting data to/from CERN

We used ftp (wacdr.cern.ch) to get and to put the data files. The problem with ftp is that it is difficult to automatise.

B.13 Taiwan

a) Software installation

We had installed the ATLAS software by a complete mirroring of the ATLAS software directory under CERN AFS. The fake AFS directory was created locally although we do not have the AFS system in our PC farm. The program compiled and ran successfully at our site. However the test run using this produced a different random number sequence for the same set of events with the same random number seed. We then installed the RPM version, which produced exactly the same random number sequence for the test run.

There are pros and cons of the different software installations. To summarise, before we make the remote CVS/CMT checkout works, the RPM installation is still preferred since it guarantees the equality of the data produced at different sites under different machines and OS. The key point is to keep using the same compiler.

b.) Problems during the data processing:

There were a series of run-time errors during the production, which have been communicated among the DC people already. I shall not mention them here.

One type of "error" happened when a long-running job hung, mostly on the 500 MHz CPU, the job then hangs forever while producing a huge log file with repeating error messages.

This kind of errors often disappear if re-run on the faster CPUs. Probably it was due to the screwing-up in the memory.

c.) Problems getting data to/from CERN

Currently the network connection from Taiwan to CERN is via the Japan-US-Europe route. A traceroute command shows that it passes APAN net in Japan, and then the Abilene gateways in US, before it goes over the SWISSCOM switch.ch into CERN.

A single stream ftp session to CERN castor via disk server wacdr.cern.ch produced about 95 KBytes/s transfer speed only. We have tried different programs and different ways of data transfer, from the multi-session normal ftp to the multiple stream capable programs such as bbftp. It proved that the simultaneous multiple stream data transfer improved significantly the effective bandwidth for data transferring. We used normally 10 streams, which produced an effective average transfer speed of about 1 MBytes/s between our hepfarm and the CERN castor.

We look forward to using the GridFTP for data transferring, which is multiple-stream, enabled. For the near future the ASCC is investigating for a several Gbps direct connection to StarLight, which will dramatically enhance the effective data transfer bandwidth between Taiwan and CERN.

Problems happen from time to time with the remote access to the CERN castor storage via the wacdr.cern.ch disk server. Sometimes it was due to the system development being performed by the castor support team at CERN. For example, we had difficulties to get connection to wacdr.cern.ch few days ago, and were later told by the support team that they were testing using different ports and different ftp modes (either active mode and/or passive mode). In principle, we have succeeded for most of the time in transferring to/from the CERN castor.

B.14 UK: Liverpool MAP (Monte Carlo Array Processor)

We have a 300 PC array (worker nodes) each with 20G of available disk space. One of 6 storage nodes (compass nodes) broadcasts the required software and data to each of the worker nodes at the start of a job. The required output is then transferred back to the compass node upon the conclusion of the job and stored on the 500GB we have available per compass node.

This architecture thus differs significantly from a PBS batch system set up on a PC farm. We have therefore encountered some (possibly unique?) challenges that had to be worked around:

a) Software installation

There was no problem in installing the software on the compass node. However, as mentioned above the worker nodes do not retain any data after the conclusion of a job. As a last resort, it would have been possible to install the RPMs on each worker node. This would have been very time consuming for 300 PCs and is certainly not a scalable method.

A mount point to the compass node was also an option. This is fine for relatively small farms but was unworkable when 300 PC are trying to access the same file simultaneously.

The only workaround available was to broadcast the contents of /opt/ATLAS to each worker node at the start of each job. Given the size of the whole directory tree (>1G) this seemed quite inefficient. Indeed, the contents of /opt/ATLAS were stripped down to only what was essential (relating to only 10% of the total file size). Ideally a static executable would have been preferred (a la LHCb) in this instance.

b.) Problems during the data processing:

The second problem was the large size of the MC file (2G). This had to broadcast to each node in its entirety although only a fraction of the file would be processed. This again slowed down the start of each job. Could these files be split up prior to execution? Or could an individual institute generate the MC rather than retrieval from a central repository?

These comments (a.) and b.) have been from the experience gained from running over 500k events in dataset 002000 - so the next point is relevant for this dataset at least.

From the scripts, the default number of events for one job to run over is 5000. It seems that the naming system of the output files is dependent on this being fixed. Ideally, we would have liked to run on a smaller number of events. This would have given us the opportunity to spread 1 partition (100k events) across 300 nodes (or say, 500 events across 200 nodes) to enable a much more economical production and a full use of our resources.

In order for the 5000-event structure to be retained we had to resort to installing 5 separate queues. Thereby splitting MAP into 5 farms of 20 PC's each. This meant discarding ~60% of the total processing power (or ~200 PC's not used) we had available for MC generation.

One solution for a previous run of LHCb MC generation was to split the MC file up but then merge the ZEBRA output upon completion of the job. Would that be possible for ATLAS? That would certainly be an easy solution to our problem.

The large job time is also a factor since MAP is a shared resource with other experiments. A much shorter run time would be therefore more preferable for us.

c.) Problems getting data to/from CERN

No problems transferring data from CERN.

B.15 US Test-bed

a) Software installation

For the DC1 grid production in the U.S., we used a tarball made by Pavel Nevski at BNL (containing binaries only for Red Hat Linux). The executables were installed on Globus gatekeeper machines at 3 (out of the 8) U.S. test-bed sites.

b.) Experience and problems during the data processing:

We used a grid scheduler to submit the jobs. This scheduler automatically submits about 60 jobs a day, wherever it finds available capacity among the 3 sites. A second scheduler was run independently to achieve a rate of ~120 submissions a day. We had an 80% success rate - most failures happened due to hardware and software failures or scheduled outages that we understand and should be able to improve next time. The submission process was completely automatic and required very little supervision or intervention (in most cases, if a site was unavailable, the scheduler continued production with the other available sites without problem).

Each production job on the grid had many stages. First the Globus gatekeeper of the site selected by the scheduler is queried for software location information. Next, a suitable available partition is chosen for production. The proposed LFN is registered in MAGDA along with various production related information. All executables are staged into a temporary location. A script with the location of the executables and environment variables is sent to the queue on the selected site. The job is started asynchronously by the batch queue system. The scheduler checks every 5 minutes if the production job has finished. Once it finishes (on average after 14 hours), the files are moved to the BNL HPSS storage system by MAGDA. All LFNs are registered in the MAGDA catalogue. A replica is also made by MAGDA at one of the available grid sites.

An independent semi-automatic quality of service (QOS) process is run periodically. This job checks the MAGDA production database for job status (the production job updates this database periodically during staging and execution) of every partition. It checks job status on the submitted queue through Globus. It verifies

through Magda that all files are correctly stored in the HPSS and replica locations. It checks if the temporary staging location has been cleaned up after production. This process can correct for many failures and updates the production database if it can recover files. For example, the BNL HPSS was unavailable for a couple of days - production continued without any changes. When HPSS was available again, the QOS process automatically copied and catalogued all primary files from the replicas using MAGDA.

Most of the problems during the 2 weeks of production are typical of distributed systems spread out over 4 locations thousands of miles away (New York, California, Texas and Oklahoma). Various machines were not available at critical times. Even when empty queues were available, however we could not run production faster than about 70-80 jobs per day at any one site. We are taking steps to fix this limitation.

MAGDA has been used in DC1 in three aspects:

- **US grid test-bed production**
- **Automatic transferring files between BNL HPSS and CERN CASTOR**.
  About 3 TB data has been copied using MAGDA. gridftp or bbftp were used for the trans-Atlantic transferring.
- **Cataloguing files by using the file spider.**
  About 12K primary ZEBRA file instances are registered in MAGDA.

c.) Problems getting data to/from CERN

Not sent any data to CERN yet - all data is being stored at the Brookhaven tier 1 site.

B.16 USA: BNL

a) Software installation

BNL used its own installation of the production software, which encapsulated ALL components in one tarfile. All job control parameters are coming from a database (Virtual Data Catalogue). All jobs were running using a set of few common NFS discs.

b.) Problems during the data processing:

During the production the following problems where observed:
- 11 jobs have to be restarted with a different random seed due to GCALOR problem
- NFS server was taken for the maintenance twice during this period (July 11 - September 1st), thus leading to the loss of all running jobs twice.

No new problem was encountered during the production period.

c.) Problems getting data to/from CERN

For the data movement from BNL hpss to CERN castor, we do it by three steps: BNL hpss -> BNLdisk cache -> CERN disk cache -> CERN castor. Generally the movement runs smoothly.

For the transferring of BNL disk cache to CERN disk cache, we run 'globus-url-copy' client at LXPLUS (some time ago we ran 'bbftp' client there). We use /tmp on LXPLUS as cache, and clean it up after we are done. I installed 'globus-url-copy' client and 'bbftp' client in my area. To avoid typing password, I have a doe certificate to run 'globus-url-copy'. (ran bbftp client + ssh-agent some time ago).

For the transferring of CERN disk cache to CERN castor, run ftp to wacdr.cern.ch locally.

Appendix C : Error messages during event simulation

In this section we summarize the different error messages during the event simulation.

GCALOR problems:

CALOR: Fatal ERROR in EVAP and
CALOR: ZEBRA banks screwed up --> STOP

- bugs in GCALOR. Due to the lack of accuracy sometimes the energy in the evaporation model is not strictly conserved at a MeV scale. This happens rarely (in less than 10^{-4} events) and is rather invisible in the range of typical hadronic energies in ATLAS. In the future this error should be ignored in the code.

Suggested solution to by-pass STOP:
:
Program has to be restarted with a different random number seed: RANLUX \$sigma($OUTPARTNR + 1000000)

Other GCALOR non-fatal error messages:

GUSTEP ERROR: after 1 iterations and 0 particles done

wrong handshaking between GCALOR and GUSTEP when more than 100 secondary particles are produced in a single hadronic interaction. . Will be corrected later, <~1% events (but several lines printed). Actually this may produce a single hit with a huge energy (overflow) and should be taken into account later in the reconstruction.

ERROR GCALOR: Particle type 1380927008 not implemented in GEANT

- gcalor loosing the particle id, < 10^{-3} events affected. This is again caused by a wrong handshaking between GCALOR and GUSTEP. When more than 100 secondary particles are produced in a single hadronic interaction, some of the particles from the above excess can be lost.

*** Strangeness non conservation in Hadriv -1 15 8 ***

PROJECTILE HADRON MOMENTUM OUTSIDE OF THE ALLOWED REGION, PLAB= 0.90575E-05

- precision problems in GCALOR, should be ignored

Ferevv: Umo2 < Urmin2 !! 6.03708507 7.37604129 20 1 1.19743 0.93827231

- precision problems in GCALOR, should be ignored

I/O system problems:

1. ***** event loop ends because the IQUEST flag set by program is -1

- input error; input ROOT file is corrupted; input file should be re-copied or re-created.

2. Error: cannot open file "iostream" FILE:/tmp/filebCTcOM_cint LINE:2

*** Interpreter error recovered *** - problem in the local ROOT installation (wrong or missing system.rootc file), can be ignored

3. error in CFPUT : Can't open configuration file

- This and other CFPUT problems appear when a local output file cannot be written (due to lack of space, denied access etc.). Job should be re-run

Harmless warnings

1. GSNGTR ERROR - I,K = 2 2 SN = 0.86949E+03, etc

Old known problem with accuracy of a helice crossing 2nd order surface in spanish fan. The message is a trace back of the improvement done a while ago (The tracking inaccuracy was reduced to an acceptable (few micron) level). No improvement is planned; seen in every run

2. ***** ERROR in HNORMA: Unknown histogram: ID= 764

Leftover message from a developer control, no danger, seen in every run; should be removed by Serguei Baranov later.

3. MDTDIG WARNING! Digitisation Overflow Nvl are 2 6 48 2** 0

MDT digitisation, $< 10^{-4}$ events affected, (but MANY lines are printed); a looper produces too many hits in a tube. This problem will be corrected later during re-digitisation with a later version of the muon code.

4. PIXBDIG WARNING! Digitization Overflow (Layer,Sublayer,Iphi) 1 20 2 11 *

- some pixel digits are lost in the readout buffer in a highly occupied wafer. Actually same loss will happen in reality because the readout buffer will have a similar depth.

5. Aucun strip touche ! Nstrips = 0 steps = 21

A hit happened in LAr in a non-sensitive area. This is not an ERROR message.

Appendix D: Strategy and Rules for the Access to the Large Datasets

As discussed in the DC meeting during the ATLAS Software Week (September 2002) a well-defined strategy how to replicate and access the large worldwide distributed datasets is needed to ensure that:

- the provenance of each partition is uniquely defined and documented (including all processing and selection steps i.e. the Metadata information)
- identical results are obtained independent on the actual location of each replica
- the HLT and physics community can perform the necessary studies with an acceptable turnaround

To achieve this:

- all datasets (~60000 partitions) and their replicas have to be registered (including Metadata information) in the ATLAS database
- coherent, managed and validated data processing is needed
- agreement on a basic set of rules is needed
- users who need access to the large distributed datasets need to be registered under the direction of the DC team.

This document describes the procedures for storage and access of large datasets. The detailed information given on number of issues (such as the sites which will store the datasets) is specific for the DC1 data. At the same time this paper should also be considered as the first step in establishing routines for how ATLAS will administrate such matters in the future.

D.1. General policy rules

General rules for datasets:

a.) Registration of datasets

ALL non-private datasets have to be registered in AMI and MAGDA (in this case including replicas) so that one can follow the different processing steps and the provenance of each dataset partition is uniquely defined (as already done in DC1 phase 1).

b.) Movement of datasets

In principle all partitions are expected to stay and be accessed where they are produced

–BUT:

To ease the access to the worldwide-distributed datasets we will replicate the data (after pile-up production and after standard full reconstruction) at 7 sites (including CERN).
The large datasets will be concentrated such that only 1 or 2 sites need to be accessed when running over a particular dataset.
The replication has to be done in a coordinated way using appropriate tools.

c.) Large-scale production

Large-scale production (e.g. reconstruction and n-tuple production) will in most cases be done at more than 1 site and will involve a non-negligible amount of partitions and CPU cycles. Therefore it should be done in consultation with or by the DC team.
The output partitions (n-tuple, AOD, ESD) including the relevant Metadata information have to be registered in the ATLAS database.

d.) Small-scale production

Primarily HLT and physics groups in consultation with the DC team will do this.
Will in most cases be done involving datasets residing at one site.
Also in this case all output partitions need to be registered.

D.2. Storage of large data sets

To facilitate the access to the large distributed datasets, since not all production sites will be accessible via Grid tools, the data will be replicated to the following 8 sites:

Alberta
BNL
CERN
CNAF (Bologna)
GridKA (Karlsruhe)
Lyon
Oslo
RAL

The replication takes into account

where the data sets were produced
where the user community is concentrated
that a large dataset should not be spread over too many sites
that related datasets (i.e. b physics) should, if possible, reside at one place

The final assignment of the datasets to those sites will be made in January 2003 after consultations of the different groups.

All those sites guarantee that all ATLAS users who need access to the large datasets can use these facilities.

D.3. User registration

Besides the DC production team, individual users will need to access these datasets. We intend to use Grid middleware wherever possible to facilitate the access to the distributed datasets. However, not all sites may on the Grid, therefore we foresee as a fallback solution running the production jobs in standard batch mode (details see under D.4.)

In order to have access to the various sites users need:

to get an individual Grid certificate (for authentification)
to be registered as members of the ATLAS virtual organisation (for authorization)
individual accounts for the sites they want to access

Potential users have to send their request for accounts at those 7 places (including some information which large data sets they want/have to access) to Monika Wielers (HLT) or Fabiola Gianotti (physics groups).

Gilbert Poulard will collect this information and will perform the necessary steps to register these persons as members of the ATLAS VO and to get accounts at the relevant places.

Before this can be done each user has to get her/his Grid certificate. This procedure is described on the Web:

http://marianne.in2p3.fr/datagrid/ca/ca-table-ca.html

and here's the link with instructions on how to get to the ATLAS VO (and the VO list itself, so that people can cross-check whether they are in):

http://www.nordugrid.org/monitor/atlasvo

D.4 Access to the datasets

The existing toolset (AtCom, GRAT, AMI, MAGDA, VDC, ...) will be extended and integrated to support both an interactive and fully automatic mode of job execution. The goal is to have an automatic update of all relevant bookkeeping information upon completion of jobs, location of input files based upon logical file names using the metadata and replica databases, controlled replication, etc. The working model should be uniform and as automatic as possible still taking into account the current and near future diversity in job scheduling systems.

With these tools it should be possible to submit jobs to Grid sites (EDG, NorduGrid and US-Grid) as well to sites running standard batch queues (e.g. LSF at CERN).

Appendix E : Pile-up Production Details

E.1 Generation of the Cavern Background

Cavern background is simulated as a separate component that is added on top of every single minimum bias event. This is done in the following steps:

1.) A standalone dedicated GEANT3/GCALOR based detector simulation program with improved neutron propagation and a simplified ATLAS geometry is run on pp collisions. The output of this program provides particle fluxes in the envelopes surrounding muon chambers. The fluxes are provided as list of particles with all related parameters per a pp interaction on the entrance of each chamber envelope.

2.) ATLSIM randomly reads from these fluxes an average number of particles per single pp collision and feeds a subset of them into ATLAS DICE geometry. At this moment all photons and neutrons entering the chamber envelopes are selected (E_{kin}>10 KeV). Charge particles are selected only the first time they appear in the output list and only if their production time is bigger than the time cut-of of the DICE simulation, so that the prompt component of the calorimeter punch-through is not double counted. The starting time of all selected particles is reset to 0-25 ns interval.

A significant randomisation is achieved at this moment due to:

a.) random initial particle selection;
b.) low probability of neutron and photon interaction in the chamber envelopes;
c.) arbitrary selected particle rotation at the input

This allows multiple re-use of the particle fluxes simulated in the first, the most CPU-consuming step.

The detailed muon system geometry description provided by DICE is used to simulate signals induced by the cavern particles in the muon chambers.

The initially selected neutral particles are propagated only within chamber envelopes to avoid double counting of the n-gamma cascade. However, all their products and initially selected charged particles are trace until the GEANT program stops them.

Hits produced during the tracking (usually in the same 0-25 ns time range) are saved in pseudo-events normalized per one pp collisions as a standard (ATLSIM) simulation output.

3.) Output of the cavern background simulations is mixed with the standard fully simulated minimum bias events (dataset 2099), thus producing new minimum bias events with the cavern backgrounds included. Mixing proportion may varies from 1 to 10 as the "safety factor" requested by the Radiation Task Force. (K^0 and their decay product are already correctly simulated to some extend in the normal minimum-bias tapes as ATLSIM contains the known bug correction for the K^0 propagation) This approach drastically reduces the time needed to simulate the signals induced in the muon spectrometer by the cavern background comparing to the previously used technique. In the same time it allows for a realistic Compton electron and spallation proton production, which takes into account, all geometry details available in DICE properly convoluted with dedicated n-gamma fluxes calculations.

4.) The resulting minimum-bias events should be added as a pile-up to any physics events. This should be done taken into account the LHC luminosity and bunch structure. To fully simulate the complete detector pile-up mixing should be done for +/- 30 bunch crossings (in the same way as it was done for the inner detector for the Physics TDR) with the average number varying from 4.6 events per bunch crossing for the low luminosity ($L=2*10^{33}$ $cm^{-2}s^{-1}$) run to 23 events per bunch crossing for the high luminosity ($L=10^{34}$ $cm^{-2}s^{-1}$) run.

E.2 CPU and memory requirements:

Step (1) is made only once for a specific muon system layout. About 10K simulated events were generated, which is only a small fraction of regular flux calculations.

Step (2) is also done once by a special version of ATLSIM with the standard DICE geometry taken from the production release 3.2.1. This step takes about 6 SI95 seconds (SI95-s) per simulated event and requires standard ATLSIM memory (<100 MB per job). The output is produced in files that contain 10K event (for comparison, Data set 2099 has 500 events per file). This is more than is needed for one to one file mixing at any reasonable safety factor. The total number of events needed at this step is about 10 Million (1000 files of 10K events each); the simulations time is of the order of $60*10^6$ SI95-s.

Step (3) should be done several times per each minimum bias tape. (for every selected safety factor 1,2,5 as planned for the moment). As each job requires one “minimum bias” and one “cavern background “ file, all three mixing could be done in one job. Each such job requires less than 2000 SI95-s but is output extensive (each 300MB input file yields 3 files close to 1 GB in total).

All together 1000 pre-mixing jobs are needed. The resulting files should be distributed over the production sites involved in the physics pileup production. (If a big enough temporary disc storage is locally available, it is possible to make this step "in flight" as a part of step (4).)

Step (4) is the most time consuming procedure as in addition to the event mixing it requires running full digitisation of the ATLAS detector. Time required per job does not depend on physics but on the luminosity only. A high luminosity pile-up job requires a 500 MB machine and takes 4400 SI95-s (800 SI95-s for mixing and 3600 SI95-s for digitisation). This step produces output events of about ~<8 MB at high luminosity independent on the input physics event size. The memory requirement (500 MB) was a matter of concern at the beginning of the exercise, however, not anymore with the more recent releases.

E.3 Problems during Pile-up Production at CERN

The number of min bias events needed per signal event is quite large: 61 x 23 = 1403 for high luminosity and 61 x 4.6 = 280 for low luminosity. Out of the 61 about 7 will make a critical contribution to the total and hence in case there is some special event in the min-bias set, it better not appear too often in these 161 (resp. 32) positions.

Processing a single signal event takes only few tens of seconds, which implies a very high input rate (e.g. 61 x 23 x 500K / 100 s = 7 MB/s). To lower this rate the pile-up code recycles more than 90% of the min-bias events used from the previous event, effectively decimating this number.

It was decided to use about 5.7/26.8 (lumi02/lumi10) x 500 min-bias events per 100 signal events. For a typical signal input file with 500 events that makes ~30 x 250 MB = 7.5 GB min-bias input per low luminosity job. Additionally, it was decided that large signal samples (like 1M events) would be piled up with correspondingly large pile-up samples (like 200K events). To this end 2000 min bias files, each containing 500 events, were produced. Each job would use a particular subset (e.g. 32 out of 400) of the min-bias files. Taken over the complete signal sample the usage distribution of min bias files would be uniform.

With a 7.5/30 GB min-bias input per job it was not possible to keep all the input on the local disk of the batch machines. Consequently, it was decided to read the min-bias input directly from castor. At the same it became clear that low luminosity pile-up would take about 100 NCU seconds and high luminosity 400 NCU seconds. These numbers were higher than anticipated. On the positive side this meant that our input rates would be lower. On the negative side it meant that to finish the same amount of signal events in the same period of time we would have to use more machines in parallel. So while the bandwidth goes down for any individual job, it remains constant for the castor stager serving all these jobs. The input rate per job is approximately (800 MB signal + 32 x 250 MB min bias)/(550 x 100 s) = 160 Kb/s (800 MB signal + 72 x 250 MB min bias)/(250 x 400 s) = 188 Kb/s On the faster batch machines (CPU-factor 2.0) these numbers double, resulting in 320/376 Kb/s.

The ATLAS CASTOR stager is served by 10 disk servers that each can sustain an output rate of 30 MB/s in ideal circumstances (single reader, single file read in sequential order). In reality the 400 min bias files were not evenly distributed over all 10 disk servers. Some servers hosted two times the average while others hosted non at all. Additionally, while any single job reads the min-bias file sequentially, many are reading the same file at the same time and not all are reading from the same place in the file. Hence, the access to the file on disk is more random than sequential. In principal running more jobs in parallel than the disk servers can serve should just slow all of them down (they become input bound). In practice, up to 10% of the jobs failed with reading errors, which forced us to limit the maximum number of parallel pile-up jobs to less than 200 (75 Mb/s

i.e. 25% of maximal). Some initial investigation as to why the jobs experience reading errors as opposed to simply slow down was started, but soon after aborted. As usual with errors that only occur under extreme stress, pinpointing the exact cause of the errors would have taken a disproportionate amount of resources both from the castor and the atlas production team.

List of Authors

R. Sturrock
University of Melbourne, ***AUSTRALIA***

R. Bischof, B. Epp, V. M. Ghete, D. Kuhn
Institute for Experimental Physics, *University of Innsbruck,* ***AUSTRIA***

A.G. Mello
Universidade Federal do Rio de Janeiro, COPPE/EE/IF, Rio de Janeiro, ***BRAZIL***

B. Caron
Centre for Subatomic Research, University of Alberta and TRIUMF, Vancouver, ***CANADA***

M.C. Vetterli
*Department of Physics, Simon Fraser University, Burnaby****, CANADA***

G. Karapetian, G. Azuelos
Laboratoire de Physique Nucleaire, Université de Montréal, ***CANADA***

K. Martens
Department of Physics, University of Toronto, ***CANADA***

A. Agarwal, P. Poffenberger, R.A. McPherson[54], R.J. Sobie[1]
Department of Physics and Astronomy, University of Victoria, ***CANADA***

S. Armstrong, N. Benekos, V. Boisvert, M. Boonekamp[55], S. Brandt, P. Casado, M. Elsing, F. Gianotti, L. Goossens, M. Grote, J.B. Hansen, K. Mair, A. Nairz, C. Padilla, A. Poppleton, G. Poulard, E. Richter-Was[56], S. Rosati, T. Schoerner-Sadenius[57], T. Wengler
CERN

G.F. Xu
Institute of High Energy Physics, Chinese Academy of Sciences, ***CHINA***

J.L. Ping
Nanjing University, ***CHINA***

J.Chudoba, J.Kosina, M.Lokajicek, J. Svec
Institute of Physics, Academy of Sciences of the Czech Republic, Praha , ***CZECH REPUBLIC***

P. Tas
Charles University in Prague, Faculty of Mathematics and Physics, IPNP, Praha, ***CZECH REPUBLIC***

J. R. Hansen, E. Lytken, J. L. Nielsen, A. Wäänänen
Niels Bohr Institutet for Astronomi, Fysik og Geofysik, Copenhagen, ***DENMARK***

S. Tapprogge
Helsinki Institute of Physics, Helsinki, ***FINLAND***

D. Calvet
Université Blaise Pascal de Clermont-Ferrand, ***FRANCE***

S. Albrand, J. Collot, J. Fulachier, F. Ledroit-Guillon, F. Ohlsson-Malek, S. Viret, M. Wielers[58]
LPSC, CNRS-IN2P3, Université Joseph Fourier, Grenoble, ***FRANCE***

[54] Also at the Institute of Particle Physics of Canada
[55] Now at CEA-Saclay
[56] Now at Crakow
[57] Now at Hamburg
[58] At TRIUMF until 01/02/03

K. Bernardet, S. Corréard, A. Rozanov, J-B. de Vivie de Regie
CPPM, CNRS-IN2P3, Université de la Méditérannée, Marseille, ***FRANCE***

C. Arnault, C. Bourdarios, J. Hrivnac, M.Lechowski, G. Parrour, A. Perus, D. Rousseau, A. Schaffer, G. Unal
LAL-Orsay, CNRS-IN2P3, Université Paris XI, Orsay, ***FRANCE***

F. Derue
LPNHEP, CNRS-IN2P3, Université Paris 6/7, Jussieu, Paris, ***FRANCE***

L. Chevalier, S. Hassani, J-F. Laporte, R. Nicolaidou, D. Pomarède, M. Virchaux
CEA/DAPNIA, Saclay, ***FRANCE***

N. Nesvadba
Rheinische Friedrich-Wilhelms Universität Bonn, ***GERMANY***

Sergei Baranov
*University of Freiburg****, GERMANY***

A. Putzer
University Heidelberg, ***GERMANY***

A. Khonich
University Mannheim, ***GERMANY***

G. Duckeck , P. Schieferdecker
LMU Munich, ***GERMANY***

A. Kiryunin, J. Schieck
MPI für Physik, Munich, ***GERMANY***

Th. Lagouri
Nuclear Physics Laboratory, Aristotle University of Thessaloniki, ***GREECE***

E. Duchovni, L. Levinson, D. Schrager,
Weizmann Institute of Science, ***ISRAEL***

G. Negri[59]
CNAF, Bologna, ***ITALY***

H. Bilokon, L. Spogli
LNF, Frascati, ***ITALY***

D. Barberis, F. Parodi
Università di Genova e INFN, ***ITALY***

G. Cataldi, E. Gorini, M. Primavera, S. Spagnolo
Università di Lecce e INFN, ***ITALY***

D. Cavalli, M. Heldmann[60], T. Lari, L. Perini, D. Rebatto, S. Resconi, F. Tartarelli, L. Vaccarossa
Università di Milano e INFN, ***ITALY***

M. Biglietti, G. Carlino, F. Conventi, A. Doria, L. Merola,
Università di Napoli "Federico II" e INFN, ***ITALY***

G. Polesello, V. Vercesi
Sezione INFN di Pavia, ***ITALY***

[59] At Milano before 01/12/02
[60] Now at Freiburg

A. De Salvo, A. Di Mattia, L. Luminari, A. Nisati, M. Reale, M. Testa
*Università di Roma "La Sapienza" e INFN , **ITALY***

A. Farilla , M. Verducci
*University of Roma Tre "E. Amaldi" e INFN, **ITALY***

M. Cobal, L.Santi
*Università di Udine e INFN, **ITALY***

Y. Hasegawa
*Shinshu University, **JAPAN***

M. Ishino, T. Mashimo, H. Matsumoto, H. Sakamoto, J. Tanaka, I. Ueda
*International Center for Elementary Particle Physics(ICEPP), the University of Tokyo, **JAPAN***

A. Fornaini, G. Gorfine
*NIKHEF**, NETHERLANDS***

J. Koster,
*Parallab / UNIFOB, University of Bergen, **NORWAY***

A. Konstantinov[61], T. Myklebust, F. Ould-Saada
*University of Oslo, Department of Physics, **NORWAY***

T. Bold, A. Kaczmarska, P. Malecki, T. Szymocha, M. Turala
*Cracow University, **POLAND***

Y. Kulchitsky, G. Khoreauli, N. Gromova, V. Tsulaia
*Joint Institute for Nuclear Research, Dubna, **RUSSIA***

A. Minaenko, R. Rudenko, E. Slabospitskaya, A. Solodkov
*Institute of High Energy Physics, Protvino, **RUSSIA***

I. Gavrilenko
*P.N. Lebedev Institute of Physics (FIAN), Moscow, **RUSSIA***

N. Nikitine, S. Sivoklokov, K. Toms
*Skobeltsyn Institute of Nuclear Physics, Moscow State University, **RUSSIA***

A. Zalite, I. Zalite
*St.Petersburg Nuclear Physics Institute, **RUSSIA***

B.Kersevan
*University of Ljubljana and Iozef Stefan Institut, Ljubljana, **SLOVENIA***

M. Bosman
*Institut de Física d'Altes Energies (IFAE), Barcelona, **SPAIN***

S. Gonzalez, J. Sanchez, J. Salt
*Instituto de Fisica Corpuscular (IFIC, Centro Mixto CSIC-UVEG), Valencia, **SPAIN***

N. Andersson , L. Nixon
NSC, Linköping University, ***SWEDEN***

P. Eerola, B. Kónya, O. Smirnova
*Particle Physics, Institute of Physics, Lund University, **SWEDEN***

[61] Also at IMSAR, Vilnius University

Å. Sandgren
HPC2N, Umeå University, ***SWEDEN***

T. Ekelöf, M. Ellert, N. Gollub
Department of Radiation Sciences, Uppsala University, ***SWEDEN***

S. Hellman, A. Lipniacka
Department of Physics, Stockholm University, ***SWEDEN***

A. Corso-Radu, V. Perez-Reale
Laboratory for High Energy Physics, Bern, ***SWITZERLAND***

S. C. Lin, S.C. Lee, Z.L. Ren, P.K. Teng
Institute of Physics, Academia Sinica, ***TAIWAN***

P. Faulkner, S.W. O'Neale, A. Watson
University of Birmingham, ***UK***

F. Brochu, C. Lester
Cambridge University, ***UK***

S. Thompson, J. Kennedy
University of Glasgow, ***UK***

E. Bouhova-Thacker, R. Henderson , R. Jones, V.Kartvelishvili, M. Smizanska
Lancaster University, ***UK***

A. Washbrook
Liverpool University, ***UK***

J. Drohan, N. Konstantinidis
University College London, ***UK***

E. Moyse[62]
Queen Mary and Westfield College, University of London, London, ***UK***

S. Salih
Manchester University, ***UK***

J. Loken
University of Oxford, ***UK***

J. Baines, D. Candlin, R. Candlin, R. Clifft, W. Li
RAL, ***UK***

S. George, A. Lowe
Royal Holloway College, University of London, Egham, ***UK***

C. Buttar, I. Dawson, A. Moraes, D. Tovey
University of Sheffield, ***UK***

D.L. Adams, K. Assamagan, R. Baker, W. Deng, V. Fine, Y. Fisyak, B. Gibbard, H. Ma, P. Nevski, F. Paige, S.Rajagopalan, J. Smith, A. Undrus, T. Wenaus, D. Yu
Brookhaven National Laboratory, ***USA***

S. Youssef, J. Shank
Boston University, ***USA***

[62] Now at CERN

J. Huth
Harvard University, ***USA***

P. Loch
University of Arizona, Tucson, ***USA***

S. Gonzalez
University of Wisconsin, Madison, ***USA***

S. Goldfarb
University of Michigan, ***USA***

H. Severini, P. Skubic
Oklahoma University, ***USA***

K. De, M. Sosebee, P. McGuigan, N. Ozturk
University of Texas, Arlington, ***USA***

L. Grundhoefer, F. Luehring
Indiana University, USA

D. Engh, E. Frank, A. Gupta, R. Gardner, F. Merritt,Y. Smirnov
University of Chicago, ***USA***

J. Gieraltowski, E. May, T. LeCompte, A. Vaniachine
Argonne National Laboratory, ***USA***

P. Calafiura, S. Canon, D. Costanzo, I. Hinchliffe, W. Lavrijsen., C. Leggett, M. Marino, D.R. Quarrie, I. Sakrejda, G. Stavropoulos, C. Tull
*Lawrence Berkeley National Laboratory****, USA***

Y. Gao, T. Ryan
Southern Methodist University, ***USA***