A Step Towards a Computing Grid for the LHC Experiments: ATLAS Data Challenge 1

Sturrock, R.; Eerola, Paula; Konya, Balazs; Smirnova, Oxana; ATLAS Collaboration

Published in:
CERN-PH-EP/2004-028

2004

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN-PH-EP/2004-028
30-April-2004

A Step Towards A Computing Grid For The LHC Experiments:

ATLAS Data Challenge 1

The ATLAS DC1 Task Force*)

Abstract

The ATLAS Collaboration at CERN is preparing for the data taking and analysis at the LHC that will start in 2007. Therefore, a series of Data Challenges was started in 2002 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 for the final offline computing environment. 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 as a worldwide distributed activity.

It should be noted that it was not an option to "run the complete production 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. The great challenge of organising and then carrying out this large-scale production at a significant number of sites around the world had therefore to be faced. However, the benefits of this are manifold: apart from realising the required computing resources, this exercise created worldwide momentum for ATLAS computing as a whole.

This report describes in detail the main steps carried out in DC1 and what has been learned from them as a step towards a computing Grid for the LHC experiments.

(To be submitted to Nucl. Instr. Meth.)

*) See next pages for the list of authors

The ATLAS DC1 Task Force

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
Laboratoire de Physique Nucleaire, Université de Montréal, ***CANADA***

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

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

S. Armstrong, N. Benekos, V. Boisvert, M. Boonekamp[b], 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[c], S. Rosati, T. Schoerner-Sadenius[d], 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[e]
LPSC, CNRS-IN2P3, Université Joseph Fourier, Grenoble, ***FRANCE***

---

[a] Now at the Institute of Particle Physics of Canada
[b] Now at CEA-Saclay
[c] Now at Crakow
[d] Now at Hamburg
[e] 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
*Universität Freiburg****, GERMANY***

A. Putzer
Kirchhoff -Institut für Physik, Universität Heidelberg, ***GERMANY***

A. Khonich
Universität Mannheim, ***GERMANY***

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

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

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

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

G. Negri[f]
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[g], 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***

[f] At Milano before Dec. 2002
[g] 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
*Università di Roma Tre 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***

S.Bentvelsen, A.Fornaini, G.Gorfine, D.Groep, J.Templon
*NIKHEF**, NETHERLANDS***

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

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

T. Bold[i], A. Kaczmarska, P. Malecki, T. Szymocha, M. Turala
*The Henryk Niewodniczanski Institute of Nuclear Physics, Krakow**, POLAND***

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

A. Minaenko[k], R. Rudenko, E. Slabospitskaya[k], 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***

[h] Also at IMSAR, Vilnius University
[i] Also at Faculty of Physics and Nuclear Techniques, UST-AGH Krakow, POLAND
[k] Supported by a grant under contract CERN-INTAS-0052-4297

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

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. Lee, S.C. Lin, 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[m]
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, N. McCubbin
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***

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

[†] Deceased
[m] Now at CERN

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***

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***

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

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

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

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

L. Grundhoefer, F. Luehring
Indiana University, ***USA***

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

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

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

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

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

1.) Introduction

In the year 2007, the Large Hadron Collider (***LHC***) is due to come into service at the European Particle Physics Laboratory (*CERN*) in Geneva. Proton beams of 7 TeV are steered to collide head-on in the middle of complex detectors. The debris of these collisions will reveal fundamental particle processes.

ATLAS[1] (A Toroidal LHC ApparatuS) is one of collaborations that have been formed with about 1800 physicists participating from more than 150 universities and laboratories in 34 countries. After on-line data reduction and data compression by a factor of several million, there will still remain a total data count of many petabytes to be analysed every year. by institutes across the globe. To ensure that the requisite resources for are available, each experiment will need a worldwide distributed data bank and computer system.

The LHC Computing Review[2] recommended that the LHC experiments should carry out **Data Challenges** (DC) of increasing size and complexity. 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.

In the future worldwide ***Grid***[3], calculations will be made at the most suitable sites on the Net, with the so-called middleware organizing access and assigning computing capacity. Grid technologies offer several advantages for a multinational and geographically distributed project: they allow for a uniform infrastructure of the project computing-wise; they simplify the management and coordination of the resources while potentially decentralizing such tasks as software development and analysis; and last, but not least, they provide an affordable way to increase the effective computing power.

In this report the ATLAS Data Challenge 1 (DC1) is described. The resources available in the different production phases and the amount of data that was processed are described in Section 2. The different activities that made up DC1 event processing (Generation, Simulation, Pile-Up and Reconstruction) are discussed in Section 3. This is followed by a detailed description of the various tools developed and used by ATLAS during DC1: Bookkeeping and Databases (Section 4), Production Tools (Section 5) and Software Distribution (Section 6). The DC1 activities using the different Grid testbeds are discussed in Section 7.

2.) DC1 Phases

During the LHC preparation phase, all experiments have large needs for simulated data in order to optimise the design of the detectors and to measure the physics performance of the experiment. These "Monte Carlo" simulations (Data Challenges) are 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);
- The particles of the generated final state are transported through the detector elements 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 to recover particle trajectories and energy depositions from the raw data;
- The (Monte Carlo) generated information (sometimes called *truth*) is saved for comparison with the reconstructed information.

The ATLAS collaboration intends to perform these DC's using as much as possible Grid tools provided by the LHC Computing Grid (LCG) project[4], the NorduGrid[5] and the USGrid[6]. DC1 saw the first usage of these technologies in ATLAS in limited-scale tests.

---

[1] http://www.cern.ch/Atlas/
[2] http://lhc-computing-review-public.web.cern.ch/lhc-computing-review-public/Public/Report_final.PDF
[3] The term 'Computational Grid' (or Grid for short) has been coined by analogy with the electrical power grid.
[4] http://lcg.web.cern.ch/lcg/
[5] http://www.nordugrid.org
[6] More details : See paper "DC1 Production in the U.S."; in preparation

For all Data Challenges it is essential to have physics content in order to engage the physicist community with the exercise, leading to a more thorough validation of the software. For DC1, in 2002-2003, a major goal was to provide simulated data needed for the preparation of the ATLAS High Level Trigger TDR[7].

The DC1 production was split into three different phases. The first phase (event generation, simulation) 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. In the third phase distributed reconstruction of the most demanding high-statistics samples was carried out at the 9 largest sites (April-June 2003); the output data of this phase was stored in 8 sites (Alberta, BNL, CCIN2P3 Lyon, CERN, CNAF Bologna, FZK Karlsruhe, Oslo, RAL).

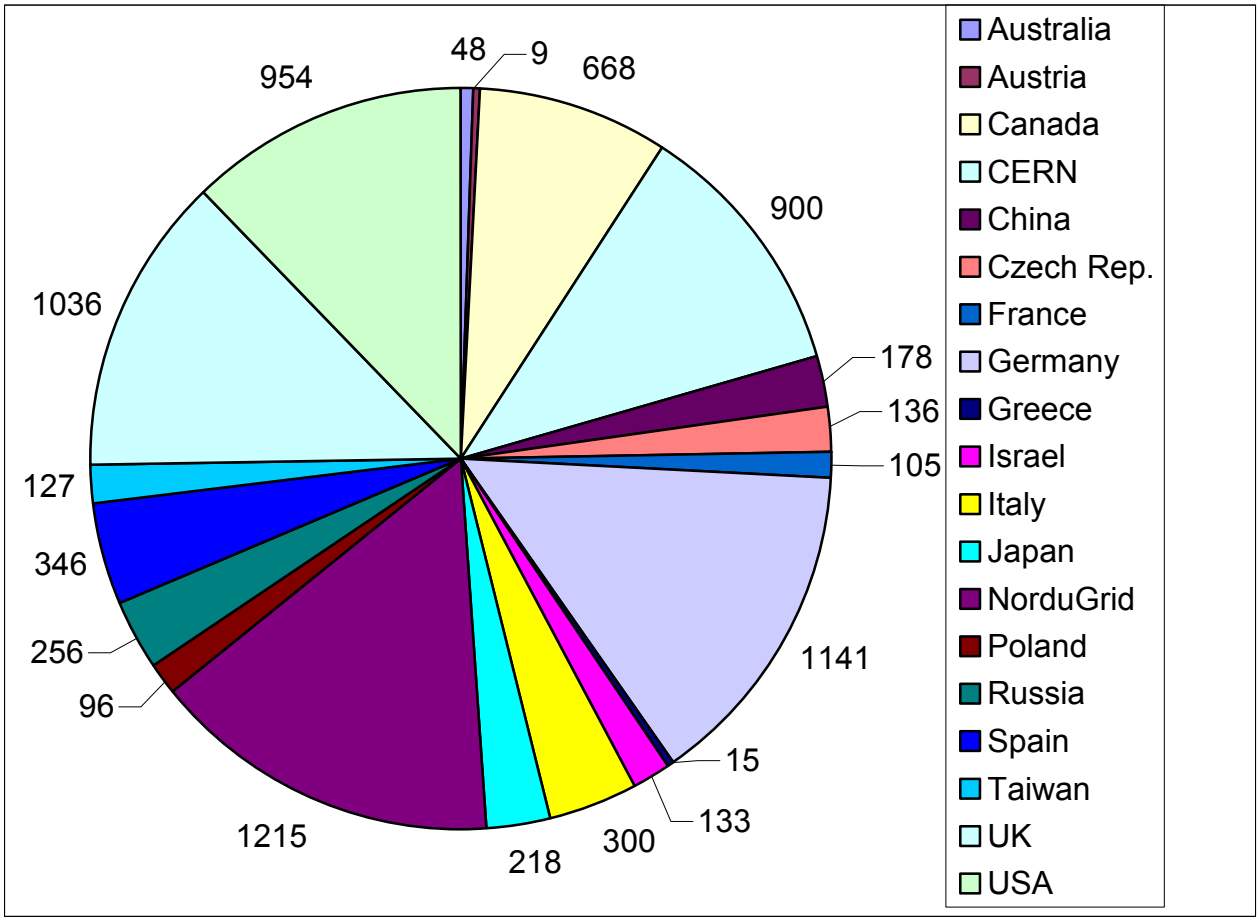

Fig. 1 Number of normalised processors[8] per country accessible to DC1 (Countries are displayed clockwise, starting with Australia (48)). The numbers of processors per site varied between 9 and 900 resulting in a total of ~5000 CPUs at peak time corresponding to a processing power of ~2.2 MSI2k.

The following 56 institutes in 21 countries[9] participated in the DC1 activities.

1.) Australia (Melbourne**)**
2.) Austria (Innsbruck**)**
3.) Canada (Alberta, Montreal, Simon Fraser, Toronto, Victoria**)**

[7] ATLAS TDR 016; CERN/LHCC/2003-022; ISBN 92-9083-205-3
[8] A normalised processor corresponds to 1 Pentium III/500 MHz.
[9] The list of the NorduGrid sites used by the three Nordic countries (Denmark, Norway, Sweden) can be found on http://www.nordugrid.org/hardware.html

4.) CERN
5.) China (Beijing, Nanjing)
6.) Czech Republic (Prague)
7.) France (Grenoble, Marseille; using CCIN2P3 in Lyon)
8.) Germany (Heidelberg, München; using GridKA in Karlsruhe)
9.) Greece (Thessaloniki)
10.) Israel (Weizmann)
11.) Italy (CNAF Bologna, Frascati, Milano, Napoli, Roma)
12.) Japan (Tokyo)
13.) NorduGrid (NBI, Odense, Bergen, Oslo, Linkoping, Lund, Stockholm, Umea, Uppsala)
14.) Poland (Cracow)
15.) Russia (Dubna, ITEP Moscow, MSU Moscow, Protvino)
16.) Spain (Valencia)
17.) Taiwan (Taipei)
18.) UK (Birmingham, Cambridge, Glasgow, Lancaster, Liverpool, RAL, Sheffield)
19.) USA (ANL, Arlington, BNL, Boston, Dallas, Indiana, LBNL, New Mexico, Oklahoma)

Fig. 2 Map of the sites (in red) taking part in the ATLAS DC1 activities. The countries with ATLAS institutes are marked in yellow.

2.1 Phase 1 (Generation and Simulation)

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. At peak time ~3200 processors located in 40 institutes in 19 countries were used. This corresponds to ~1400 kSI2k 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 one year corresponds roughly to 50 % of the yearly hardware investment needed from 2006 onwards for the non-CERN part of the ATLAS Offline Computing.

During Phase 1

- about 50 million events in total were generated via PYTHIA;
- about 51 million events 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 total data volume produced during Phase 1 was about 24 TBytes and about 8 TBytes for generated events; the total CPU time necessary to generate all the events was about 200kSI2k-days, the time to simulate all the events about 1.4 MSI2k-days.

2.2 Phase 2 (Pile-up production)

In DC1 Phase 2 pile-up was added to a sub-set of data samples as described in Section 2.4. New countries, China, Greece and new institutes from Canada, Italy, NorduGrid, UK and USA joined the effort in the course of the Phase 2 so that 56 institutes in 21 countries are participating in DC1 phase 2 giving a total of 1.5 MSI2k's. About 3900k events were produced for low luminosity and 2650k events for high luminosity. This part of DC1 took about 3.2 MSI2k-days and produced a total data volume of about 34 Tbytes in 32000 partitions.

2.3 Phase 3 (Reconstruction)

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. Therefore, the processing of the data was mostly done in those countries. About 6400 k events were processed during the reconstruction phase. This part of DC1 took about 4.4 MSI2k-days and produced a total data volume of about 200 GBytes in 25000 partitions.

3.) The Different Phases of Event Processing

3.1 Event Generation

The event generation can use several event generators and can run either in the Fortran Atlsim[10] framework or in the official ATLAS Athena/Gaudi[11] framework. The fast simulation, Atlfast[12], was used in the quality control process for the generated data, as described in section 3.2. The generation of all event samples was done at CERN using Pythia 6.203[13].

3.2 Event Generation Monitoring

The quality of the generated events produced was monitored by histogramming various characteristic properties of those events. In addition an 'ntuple' was also produced by the HistSample algorithm containing quantities

[10] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/ATLSIM/atlsim.html
[11] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html
[12] http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/HIGGS/Atlfast.html
[13] http://www.thep.lu.se/~torbjorn/Pythia.html

related to the jet structure of the event. Jet finding was performed by running Atlfast with the smearing, which would normally account for detector and reconstruction imperfections, turned off, and then using Atlfast utilities to perform the jet finding at the particle level. The ntuple was then used in a secondary job, running in the Physics Analysis Workstation (PAW[14]) environment, to produce histograms of the number of reconstructed jets, their transverse momentum (p_T) spectra and pseudo-rapidity distributions. These are normalised in various ways as appropriate: to the number of events; to the number of jets; and to the total cross section. Finally, the two histogram samples (event characteristics and jet properties) 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.

3.3 Event Simulation

The ATLAS detector simulation code used for DC1 is Fortran-based. It uses GEANT 3.21[15] to track the events through the detector and runs in the Atlsim framework. During the simulation phase the events were analysed by a filtering routine that looked for a predefined minimum energy deposition in the detector elements. Only events selected by this filter were passed to the simulation step and then written out in the form of ZEBRA[16] banks.

3.4. Pile-Up Procedure

The cross-section for inelastic, non-diffractive pp interactions at the LHC is expected to be around 67 mb. At the design luminosity (10^{34} $cm^{-2}s^{-1}$) of the LHC, the average number of minimum-bias events is 23 per bunch crossing, varying according to a Poisson distribution. Any collision recorded in the ATLAS detector therefore contains a superposition of particles coming from several interactions. In general the particles from a single "interesting physics" event will have triggered the readout, and additional particles will come from other uninteresting pp collisions.

The total number of observed particles per recorded event depends on the signal collection time. In the Liquid Argon calorimeters the signal is measured shortly after the trigger, so that it is affected only by previous bunch crossings. In contrast, measurements in the Transition Radiation Tracker or the Muon Drift Tubes continue for the maximum signal collection time (about 700ns) so that they are sensitive to bunch crossings after the triggering event.

Neutrons fly around the ATLAS cavern for a few seconds until they are thermalised, thus producing a kind of permanent neutron-photon “bath” resulting in a steady 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". To take care of this effect, special minimum-bias files were produced that 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.

3.5 Quality Assurance and Data Validation

The aim of the ATLAS DC quality assurance and validation procedure[17] was threefold:

- to ensure the compatibility and reproducibility of samples produced at different sites (site validation);
- to monitor the changes and improvements to the ATLAS detector geometry;
- to check the physics content of the generated samples (physics validation).

The validation test suite consists of a modular analysis structure based on PAW, which runs off a general-purpose ntuple from the ATLAS reconstruction framework (Atrecon[18]), and which contains information on Monte Carlo event generation and the reconstruction for all ATLAS sub-detectors.

[14] http://paw.web.cern.ch/paw/
[15] CERN Program Library W5013
[16] CERN Program Library Q100/Q101
[17] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DC/Validation/www/
[18] http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/reconstruction.html

The analysis procedure consists of two steps. First, an open-ended list of sub-detector-specific macros is run from a master process to produce the two sets of validation histograms. Second, 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 **site validation** 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 check of the simulation infrastructure at the contributing DC sites. For example, it made it possible 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.

The **physics validation** of the data was carried out in parallel with the other checks. A comparison of the number of jets, b-jets, c-jets, electrons and photons in each event, with a similar sample produced in a previous large-scale production, was performed. In addition, new fully simulated samples were inspected and were used for detailed detector calibration and similar purposes. New samples were also used extensively to study b-tagging. The b-physics group validated the DC1 simulation-reconstruction software chain for several detector layouts [19].

3.6 Event Reconstruction

The new ATLAS reconstruction software is based on the Athena/Gaudi framework. In short, the Athena framework embodies a separation between data-specific members and methods and those primarily concerned with algorithms. Data objects are handled through the Transient Event Store for event information and a Transient Detector Store for condition information. Data (specified by object type and string key) are 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 the specification of algorithms and services parameters and sequencing. The Athena executable itself is very small, as all the significant software is dynamically loaded at run time, with typically one library per package.

Since the Data Challenge was carried out before the new C++ database system (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 content into an ntuple through the Gaudi ntuple converter. The algorithm is fully configurable, so that only the needed information is written out, which is especially important for the large truth information. The main drawback was that the downstream analysis could only be done in PAW (or ROOT after conversion), as the C++ design of the original objects was not preserved.

In a separate step, algorithms running in the HLT Selection Software environment reconstruct objects and extract features from event data; these features are used to derive the trigger decision.

4.) ATLAS Bookkeeping and Databases

Essential components required for ATLAS Monte Carlo production are the associated bookkeeping and meta-data services. In addition, a well-defined strategy for how to replicate and access the large worldwide-distributed datasets[20] 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.

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

[20] In the context of DC1 the term "dataset" was taken to mean a collection of events. Each dataset has a logical dataset name, unique within ATLAS, and assigned following a nomenclature established by the production team. Datasets have to be divided into partitions because of file-size limitations. A partition is a file which contains a part of a dataset. Dataset partitions are assigned Logical File Names (LFN). The LFN contain by convention, at least the key part of the logical dataset name, and a partition number. Logical File Names are also unique within ATLAS.

Several bookkeeping tools were developed or adapted for use in DC1. The bookkeeping tools *AMI*[21] *(ATLAS Metadata Interface*) and *MAGDA*[22] *(MAnager for Grid-based DAta* used on the full set of data are described in this section. The production tools are described in Section 5, and the Grid tools used to perform a small subset of the production are described in Section 7.

4.1 ATLAS Metadata Interface (AMI)

AMI[23], developed at LPSC Grenoble, comprises a database containing metadata on produced datasets and partitions (name, size, processing time, physics contents, transformations, etc.), together with command-line and web interfaces, and various search possibilities. Thus it provides a framework to make relational databases self-describing and a software layer to exploit this additional information. It was used in the context of DC1 to manage the bookkeeping database that stores descriptive information (meta-attributes) about the data (binary files).

The attributes of the binary data file that are stored by the bookkeeping database are "logical". These are application specific properties that could not be guessed by any outside system, such as a Grid application. An example of a logical attribute is the type of event contained in the data file. Such logical attributes never change when data files are replicated.

The aim of the bookkeeping catalogue is twofold:

- to make it possible to understand the contents of a file of binary physics data without actually having to open it;
- to search for data given a set of logical attributes.

The set of attributes defined for DC1 datasets was established in close collaboration with the production management team. It also included a few parameters that are not strictly logical attributes, for example the data file sizes. These have been included to ease the production of statistics.

AMI provides several interfaces for use by physicists, and an API for developers.

AMI Architecture

AMI is written in Java. As a consequence, 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 as shown in Figure 3. The core packages manage the remote connection to the database, and the transmission of SQL commands. While MySQL was chosen for DC1, any database that understands SQL, and for which a Java JDBC driver is available, may be used.

Each AMI compliant database must contain a certain number of tables that describe it. The middle layers of AMI provide generic classes for accessing these databases, making use of these internal descriptions. This generic mechanism hides details of database implementation from clients. If clients query an AMI database, they are not expected to have knowledge of the name of the database, the name of any database table, or the relation between them. The architecture allows clients to express queries in terms of the application semantics. Thus a user of the DC1 AMI production bookkeeping database should be able to work with a schema of datasets, partitions or files and events, whereas a client of another AMI-based application would work with a different semantic. The core and middle layer software are common to the two sets of users.

The top layers of the software are specific to the particular project. In the case of DC1 bookkeeping, the top layer contains classes to manage dataset provenance, dataset definition, dataset nomenclature protocol and production statistics. Some specific web interfaces have been developed which make use of this layer.

The architecture allows geographic distribution of databases; 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 are physically at the LPSC Grenoble, and are situated on the same server.

[21] http://atlasbkkl.in2p3.fr:8180/AMI/
[22] http://www.atlasgrid.bnl.gov/magda/info
[23] http://arxiv.org/abs/hep-ex/0304029

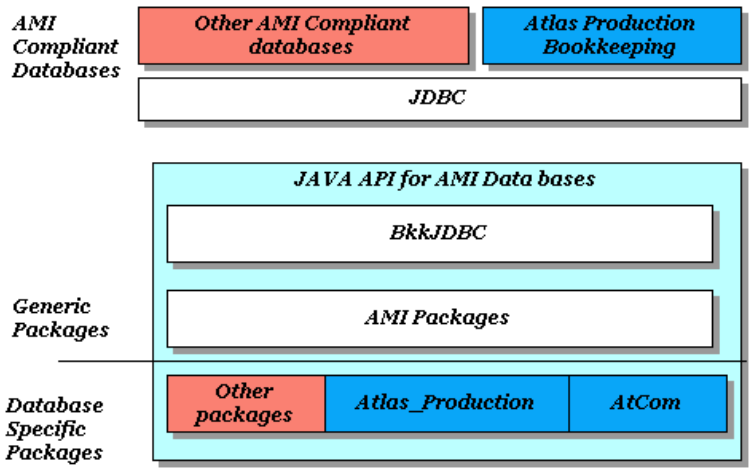

Fig. 3 The 3-tier architecture of AMI

The Command Line Interface

Physicists use this interface to input and update information in the databases. For DC1 the client software was available either as a stand-alone JAR file, or as part of the ATLAS software release. The client software makes use of a configuration file containing the server and port details of the database, and other parameters common to all the commands. The first client session must configure the software using a database password distributed by a separate mechanism.

A large number of commands are available, including commands to query the database, and to obtain information about the attributes.

For jobs that did not use the ATLAS production tools, AtCom or GRAT (described in Section 5), the information for updating AMI was contained in batch job log files. Scripts were provided to parse the log files and to generate a set of AMI commands for insertion or update of AMI records.

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 constructed SQL is visible to the user, and can be directly edited if the user desires. Users can navigate from the search results to a graph showing the 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 approve the dataset, with the possibility of editing some fields, or refuse it.
- A third interface is available to give an 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.

4.2 Manager for Grid-based Data (MAGDA)

MAGDA[24], developed at Brookhaven National Laboratory (BNL), provides automated file registration and replication tools. Thus it is a manager for grid-based data. It is built upon a database, which describes *where* the data reside; thus it complements AMI, which describes *what* the data are.

[24] http://arxiv.org/abs/physics/0306105

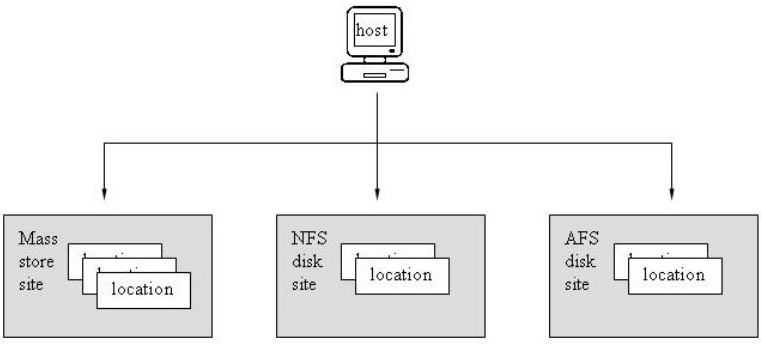

Fig. 4 Logical relations among site, location and host in MAGDA

The ATLAS experiment data are distributed among globally dispersed storage facilities. In MAGDA the concept of a 'site' is used to abstract the storage facility. The term 'location' is used to denote data locations within a site – typically directory trees. Locations carry attributes characterizing the data stored there, such as whether the data are a master or replica copy. The concept of a 'host' is used to represent a set of computers that have access to a defined set of sites. Thus a MAGDA service, having knowledge of the host it is running on, is automatically aware of all the data locally accessible to it.

MAGDA makes use of MySQL, Perl, Java, and C++ to provide file cataloguing, retrieval and replication services. For data movement, gridFTP, bbftp and scp can be chosen depending on available protocols.

The principal components of the system, implemented as MySQL tables, are:

1. A file catalogue with logical and physical file information and metadata. File metadata describe the file attributes and content. The file catalogue supports the notion of master and replica instances.
2. Site, location and host catalogues.
3. The catalogue of logical file collections.
4. The task catalogue. Automatic file replication operations are organized into reusable tasks.

The bulk of the system is to provide surrounding infrastructure for the following:

1. Setting up and managing distributed sites with associated locations, and locations within those sites, and the hosts on which data-gathering servers and user applications run.
2. Gathering the file information from the various sorts of data sites. A file spider cron job can be set up to automatically scan several storage sites and reports to the catalogue database.
3. Interfacing to users via web interfaces for presenting and querying catalogue information and for modifying the system.
4. Replicating and serving files to production and end-user applications.

5.) ATLAS Production Tools

A number of tools were developed and used to ease the production and the monitoring of the DC1 data processing. Among them:

AtCom[25] (short for ATLAS Commander), developed in Europe, an automated job definition, submission and monitoring tool, directly working with AMI;

GRAT[26] (short for the GRid Application Toolkit), developed in the context of the US Grid projects.

[25] http://atlas-project-atcom.web.cern.ch

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. The technical details are described in separate papers (references see below).

5.1 ATLAS Commander (AtCom)

The purpose of AtCom[27] 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. The interactions with the various computing systems are defined by means of separate 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 ATLAS sites, AtCom allows several of them to be used transparently at the same time.

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.

Figure 5 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 is the set of plug-ins that interface AtCom to the various flavours of computing systems.

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 into a number of partitions, each corresponding to a separate logical file. At 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.

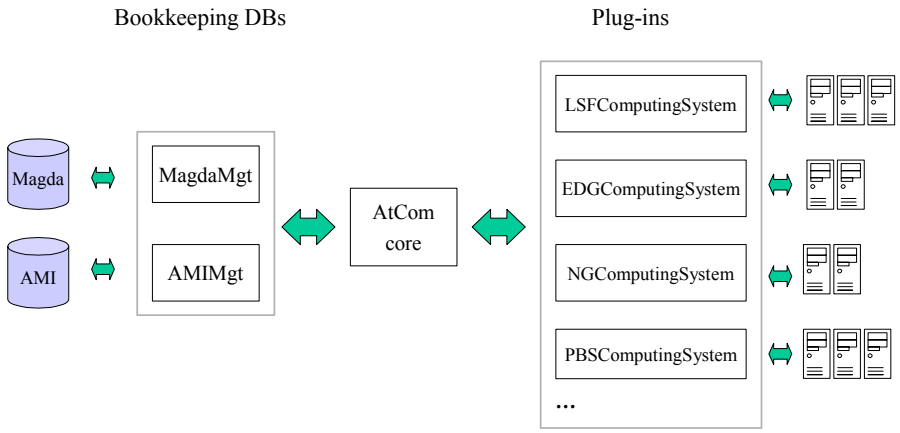

Fig. 5 The AtCom architecture

[26] http://heppc1.uta.edu/atlas/software/grat
[27] http://arxiv.org/abs/hep-ex/0305089

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.

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 be edited when needed. 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 to 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, etc.

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.

5.2 Grid Application Toolkit (GRAT)

GRAT was developed to facilitate automated ATLAS Monte Carlo production in a Grid environment. It consists of bash and python shell scripts, and was frequently modified and updated over the course of DC1, both to add new features and to adapt to the evolution of Grid middleware.

GRAT provides software utilities to handle all phases of Monte Carlo production, including job definition, job submission, verification of results, data storage management, as well as production site management. Job definition tasks include adding new datasets to the production database and incorporating new steps into the production chain as required. During actual job submission, GRAT can launch either single or multiple jobs at a remote site, or optionally a given dataset can be subdivided for submission to several sites in parallel. Post-execution, results are verified by performing data quality checks (usually via analysis of log files), and errors are automatically corrected where possible by restarting jobs and moving output files to the final storage location. Failed jobs that cannot be recovered are cleaned up and the jobs database modified accordingly. Management of storage resources includes the movement and/or deletion of verified input files, cleaning up temporary storage areas once jobs are completed, and disposing of replica file copies from intermediate processing steps. In the area of production site management, GRAT monitors job manager queues and running jobs, using both dynamic lookups and database queries. Information about disk storage resources at a site is provided, and the availability of required production software is verified. In the case of "pile-up" production, pre-staging and management of the pile-up input files is performed.

Data management tools are provided to facilitate interactions with the various databases. New information can be added, for example by creating or updating entries in AMI. Database queries allow one to obtain information regarding single entries in the production database, to view summary information, to make a decision whether a job is in a "hung" state, and to view the characteristics of jobs waiting to process. Other utilities enable consistency checks such as: (1) scanning for and correcting bad records; (2) ensuring the accuracy of replica copies; (3) verifying the existence of generated files in MAGDA; and (4) ensuring the integrity of common data distributed among multiple databases.

Phases of Execution

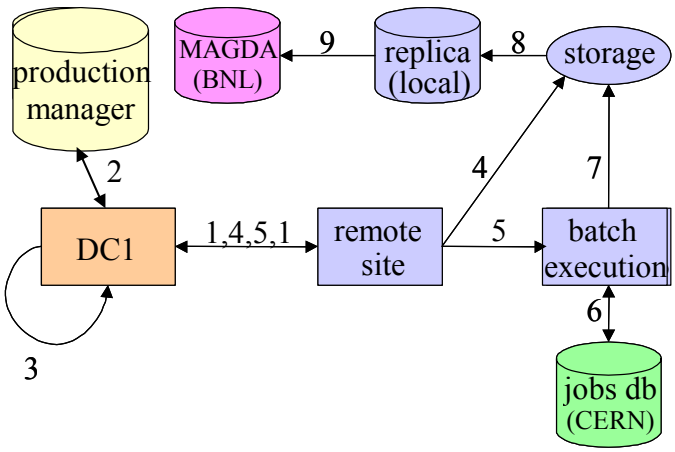

Fig. 6 Execution flow for a typical simulation job.

An example of the execution flow for a typical simulation job is shown in Figure 6. 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 database is interrogated to determine which dataset should be simulated. The next available set is reserved and job-specific information is registered.
3. *Job Customisation* - 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[28] gatekeeper service.
6. *Execution/Parameterisation* - At simulation start-up, the Virtual Data Cookbook (VDC) database is interrogated 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. *Cataloguing* - The job schedules a transfer, via GridFTP using MAGDA, from the replica location to the master location at BNL for permanent storage in the HPSS system.
10. *Monitoring* - Monitoring the state of the job, through query of the production database, Globus 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 will be available for the next production run.

6.) Software Distribution

The ATLAS software is split into more than 500 packages residing in a single repository at CERN. The Configuration Management Tool, CMT[29], manages package dependencies, libraries and executable building. New releases are built at CERN approximately every three weeks, following a predefined plan for the introduction of new features.

The compilation process is done on Linux machines. Users with a good network connection and access to AFS could use executables and data files directly from CERN. This approach is of course not suitable for remote sites

[28] The Globus project, http://www.globus.org/
[29] http://www.cmtsite.org/

with a limited network bandwidth or without access to AFS. Therefore, the relevant ATLAS software releases have been packaged into RPM format. The kits, along with the installation scripts, are available for download[30] via secure web connection or, otherwise, from the various Grid sites, EDG [31], USGrid or NorduGrid.

The general criteria, followed during the package architecture development phase, have been to build a self-consistent distribution procedure, not dependent on the Linux version. The installation has been designed to keep the same directory structure as in the CERN repository. To be consistent with the reference software produced at CERN, all the executables and libraries shipped with the kit are copied from the official ATLAS AFS software repository.

The packages are organized in a set of base tools that are required for all installations, and several additional components. The minimal installation provides the following items:

-the set-up and management scripts;
-the official ATLAS compilers;
-the required libraries not part of the ATLAS software (external packages);
-the ATLAS executable, libraries and data needed at runtime.

Other packages are provided to enable code development.

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 some sites for the DC1 production have also adopted other types of installations. In particular a procedure based on full mirroring of the distributions directly from the CERN AFS repository, and the production of an alternate set of RPMs, have been used on the NorduGrid testbed.

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.

7.) 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 such as High Energy Physics experiments, 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 optimising the usage of distributed resources, the Grid will naturally offer to all members of collaboration a uniform way of carrying out computational tasks. This is essential for large production tasks, which need plenty of worldwide distributed resources, both hardware and human.

A significant fraction of DC1 was performed in the Grid environment, involving about 21 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. The concept of Virtual Data, put forward by the U.S. GriPhyN Project[32] was also used in a prototypical way in a part of the DC production.

7.1 Prototyping Virtual Data Approach

In HEP computing, preparation of the recipes for data production (the “human data”) requires significant effort and encapsulates a considerable knowledge. The experience in ATLAS so far has demonstrated that development of the production recipes typically involves several feedback cycles in order to assure the correctness of the generated data. The necessity to verify and reproduce these results makes the development of the production recipes a laborious iterative process.

The GriPhyN project emphasises this perspective on recipes and virtual data:

[30] https://classis01.roma1.infn.it/atlas-farm/atlas-kit
[31] http://datagrid.in2p3.fr/distribution/applications/wp8/atlas
[32] http://www.griphyn.org

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

It is useful to distinguish (both conceptually and in design) the data required before the invocation of a transformation from the history information collected during and after the data transformation[33]. In that regard the Virtual Data Cookbook (VDC) catalogue encapsulates the specific data transformation knowledge and the validated parameters settings that must exist before any transformation. The fully implemented DC1 workflow is rather complicated. For each data transformation step in the DC1 processing pipeline, the essential content of the verified data production recipes was captured and preserved in a Virtual Data Cookbook database. The collection of production recipes – VDC – complements ATLAS Grid tools deployed in ATLAS Data Challenge production as shown in Figure 7.

Because Virtual Data technologies were in the prototyping stage at the start of DC1, the data volume allocated for the production test of the system was limited to about one fifth of all the DC1 data. A production system, utilizing the VDC prototype, implemented the scatter-gather data-processing architecture to enable high-throughput computing. The parameters for simulations were catalogued in the VDC database, with attribute collections normalized according to their non-overlapping domains: data reproducibility, application complexity, and grid location.

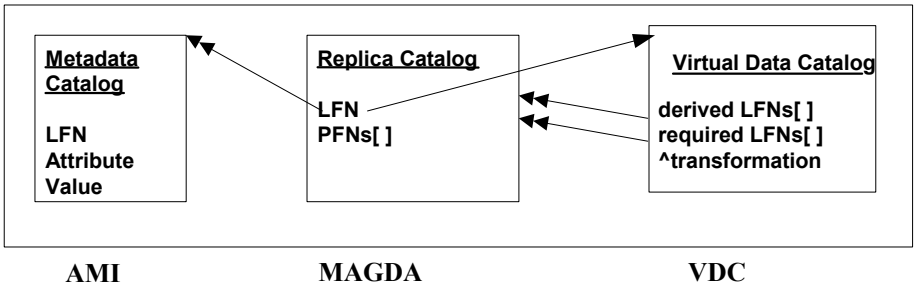

Fig. 7 Architectural view of the relationships between the three catalogues present in the Virtual Data System[34] and the corresponding ATLAS Grid tools that were deployed and used in ATLAS DC1 as the components of data management architecture supporting the processing workflow.

To provide the local-remote transparency during DC1 production, the VDC database server delivered in a controlled way both the validated production parameters and the templated production recipes for thousands of event generation and detector simulation jobs around the world, simplifying the production management solutions. Given that the production system relied on the VDC server running at one central location (CERN), the reported failure rate due to such a 'single point of failure' architecture was remarkably low (better than 1 per thousand) over the whole DC1 production period. Further improvement in the VDC services robustness will be achieved by deploying catalogue replicas at different geographic locations.

The major benefit of VDC database technology was demonstrated by simplifying the management of the parameter collections that were different for each of the more than two hundred datasets produced in DC1. Significant reduction in the parameter management overhead enabled successful processing of about half of all the DC1 datasets, representing 20% of the total data volume, using the VDC services.

7.2 DC1 Production on NorduGrid

The aims of the NorduGrid project have been, from the start, to build 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

[33] A. Vaniachine et al., Prototyping Virtual Data Technologies in ATLAS Data Challenge 1 Production. http://arxiv.org/abs/cs.DC/0306102
[34] J. Vöckler, M. Wilde, I. Foster, The GriPhyN Virtual Data System, GriPhyN 2002-2, January 2002.

were able to participate in the overall DC1 exercise using solely the NorduGrid environment[35]. During the whole DC1, more than 2 TB of input data were processed and more than 2.5 TB of output data were produced by more than 4750 Grid jobs[36].

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 but used by all sciences.

In Phase 1 of DC1, all the input files were pre-staged (replicated) at all the sites and output files were stored at a designated 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 not feasible, 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. Input data for this phase were stored in many storage facilities, including servers at BNL, and were located by the NorduGrid services with the help of the Replica Catalogue.

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 USGrid via PACMAN.

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

7.3 DC1 Production on the U.S. Testbed

DC1 production in the U.S. was carried out using both batch and grid facilities[37]. Batch processing was done at BNL. Grid processing took place in the U.S. ATLAS grid testbed, a widely distributed computational grid comprising eleven institutions. 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 RedHat Linux. The executables were initially installed on Globus gatekeeper machines at three U.S. testbed sites. Production was eventually expanded to seven testbed sites. All simulation and pile-up production in the U.S. testbed was carried out using the GRAT system described earlier in this paper.

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

[35] http://www.nordugrid.org/documents/atlasdc1.html
"Building a Production Grid in Scandinavia". P.Eerola et al., IEEE Internet Computing, 2003, vol.7, 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.
[36] More details: See ATLAS Note : ATL-SOFT-2003-002
[37] More details: See paper "DC1 Production in the U.S.; in preparation

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 two weeks of production were typical of distributed systems spread out over four 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. Seven sites were used for pile-up production, which had much more complex requirements since hundreds of inputs files were merged together randomly. The output files also had to be split because of maximum file size limitations.

More than two million events were fully simulated, piled-up and reconstructed in the U.S. The majority of the jobs during the simulation phase was done with the batch system at BNL, while we learned how to use the grid systems and developed the GRAT software. All the complex jobs in the pile-up phase were done on the grid testbed. A majority of jobs during the reconstruction phase was done with the batch system. A new grid system based on Chimera was used also for reconstruction. During the various phases, over 30 Terabyte of data was stored on disk and HPSS systems in the U.S., about 40 years equivalent CPU cycles were used, and over 50 thousand files were generated and catalogued using MAGDA. Approximately half the jobs were done using the GRAT software, demonstrating the usefulness of the grid in exploiting ten sites distributed in the U.S.

7.4 DC1 Production using the EDG testbed

The European DATAGRID (EDG)[38] project aims to develop a complete Grid solution, which includes the Globus -based middleware, as well as tools for fabric and mass storage management, and network monitoring.

In July 2002, at the start of DC1 Phase1, it was decided to start a focused effort by forming a Task Force involving ATLAS and EDG personnel, with substantial support by the EDG management and especially by members of the HEP Applications Work Package. To evaluate the EDG middleware, a small subset (~1% of the complete data) of the DC1 was chosen. The first tests (July-Sep 2002) showed several major problems in various areas of EDG middleware. This triggered developments during September 2002 in the areas of Information Systems, Data Management and Job Submission. More details on the ATLAS EDG tests in DC1 Phase 1 are available on the web[39].

During the tests, EDG releases 1.2.0 and 1.2.2 were used first in Phase1; the final Phase 1 tests were performed with release 1.4.3. These releases were installed at the 5 core EDG Testbed sites[40] and one external site[41] involved in the CrossGrid Project[42], constituting the Applications Testbed infrastructure. All the sites were listed in the Information Index of the dedicated Applications Testbed Resource Broker. Only these Resource Brokers were used in the tests.

[38] http://eu-datagrid.web.cern.ch/eu-datagrid/
[39] https://edms.cern.ch/file/375586/1.2/wp8-D8.3-0119-3-1.pdf
[40] CERN (Geneva, Switzerland), NIKHEF (Amsterdam, The Netherlands), CC-IN2P3 (Lyon, France), RAL (Oxford, United Kingdom), and CNAF (Bologna, Italy).
[41] FZK (Karlsruhe, Germany)
[42] The CrossGrid project, http://www.crossgrid.org/

All the resources and users had certificates issued by one of the EDG Certificate Authorities[43]. Registration in the ATLAS Virtual Organization was performed for all ATLAS users participating in the EDG tests. The final Phase 1 tests showed a global efficiency of only about 50%, mainly due to local problems (site configuration, lack of disk space on the Storage Element, etc.). On the basis of these results, it was decided to use the EDG testbed in May for the production of a small part of the ATLAS reconstruction (DC1 Phase3). Thus 250 reconstruction jobs processing about 50 000 events have been run in spring 2003. The sites involved were Cambridge, Lyon, Milan, Roma and Bologna, with Resource Brokers set up at Bologna and Lyon.

The ATLAS reconstruction software required RH 7.3, which was not yet officially supported by this version of EDG middleware. Additional work had to be done to create new LCFG[44] profiles to install both the operating system RH 7.3 and the EDG software on the Worker Nodes, while keeping the gatekeepers running RH 6.2. This required certain merging with EDG release 1.5.0, which was forthcoming at the time.

The input data were copied on the Storage Elements of the participating sites. The job submission was made as transparent as possible, specifying only the required input file (which was always local) and the job type. Output files were manipulated manually, by storing them at a local SE and registering them into the ATLAS RC. The task of the matchmaking of the resources has been assigned to the EDG Resource Broker (RB), which performed it successfully.

The Resource Broker and the whole EDG middleware have shown good stability over a period of about 2 weeks, requiring only slight interventions of the site managers. It should be noted that this mini-production did not constitute a stress test: the ATLAS job rate was modest and few activities from other users were going on in parallel.

Approximately 15% of the jobs failed for various reasons (mainly not due to EDG middleware)[38] and had to be re-submitted. This mini-production demonstrated that the EDG middleware was actually very capable of handling ATLAS production jobs. Although it was not of a scale to verify the stability and the scalability of the middleware in case of a real large production, it has provided evidence that the EDG performance was greatly improved during the first quarter of 2003 and that many the problems, previously identified by ATLAS, were solved.

8.) Conclusions

ATLAS Data Challenge 1 ran from spring 2002 to spring 2003. For several reasons it was divided into 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 13.5 million of SI2k-days were used to produce 10 million physics events and 40 million 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 in the effort.

- The pile-up production (Phase 2) ran smoothly. 3.2 MSI2k-days were needed to produce about 34 TBytes of data.

- A large fraction of the data has been reconstructed (Phase 3) in offline and/or trigger reconstruction mode. 4.4 MSI2k-days were needed to produce about 200 Gbytes of data.

The numbers for all three phases together are approximately:

- 21 MSI2k-days
- 70 Tbytes produced
- 100000 partitions

[43] EDG Certification Authorities, http://marianne.in2p3.fr/datagrid/ca/
[44] A Large Scale UNIX Configuration System, http://www.lcfg.org

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 Grid tools. Therefore ATLAS has been extremely active in Grid matters since mid-2002. During DC1 we have seen the emergence of the production on the Grid. Grid tools were used intensively on NorduGrid and U.S. testbeds. We are confident that their use will continue to grow.

In summary, ATLAS DC1 has proved to be a very fruitful and useful enterprise, with much valuable experience gained, providing feedback and triggering interactions between various groups, for example groups involved in ATLAS computing (e.g., HLT, offline-software developers, Physics Group), Grid middleware developers, and CERN IT. Much has been learned from DC1, and much more will doubtless be learned over the next months, when more Grid tools will be used in DC2. 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.

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 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 our Data Challenge successfully 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.

It is with the deepest regret that we must note the untimely death of our colleague Steve O'Neale. Steve made a huge contribution to ATLAS computing over many years, not least to DC1, and he will be sorely missed.