ATLAS Computing

Presentation at CHEP97, Berlin

Jürgen Knobloch

CERN/ECP, 1211 Geneva 23, Switzerland

A summary of the ATLAS Computing Technical Proposal [1] is given, outlining the plans of the ATLAS collaboration for computing in terms of software development, computing infrastructure, manpower and cost. We can only lay down the requirements, as they are known today, technical directions that we currently follow, and a rough estimate of the resources needed. Computing is instrumental to the success of the experiment. In many of the areas, we do not yet have solutions that are adequate to the scale of the problem in terms of complexity, data rate and data volume, organisational problems due to the world-wide dispersion of resources, and duration of the project.

The Challenges

Computing for the ATLAS experiment at the LHC proton-proton collider at CERN will pose a number of new, yet unsolved technological and organisational challenges:

- The CPU power required would currently not be affordable. For the offline processing, it is estimated that 2.5x10^5 SPECint95 (10^7 MIPS) will be needed. To collect this compute power today, it would take 50,000 of the most powerful PCs. We count on the continuing improvement of the price-performance ratio of CPU power and of the corresponding disks and memory.
- The data volume produced by the experiment of about 1 Pbyte (10^{15} bytes) per year requires new methods for data reduction, data selection, and data access for physics analysis. The basic consideration is that every physicist in ATLAS must have the best possible access to the data necessary for the analysis, irrespective of his/her location. The proposed scheme consists of archiving the raw data (1 Pbyte/year) selected by the

Level-3 trigger system. A first event reconstruction will be performed at CERN, on all data a few hours after data taking. For this processing, basic calibration and alignment have to be available. The purpose of this processing is to determine physics quantities for use in analysis and to allow event classification according to physics channels. The data produced in the processing have to be accessible at the event level and even below that at the physics object level. We are considering an object-oriented database system for this purpose. One copy of the data will be held at CERN. We also consider replicating some or all of the data at a small number of regional centres.

- The world-wide collaboration will require performant wide-area networks for data access and the physics analysis. Big improvements in the price-performance evolution of networks are hoped for in view of the de-regulation of the PTTs, the evolution of the Internet and the wide-spread use of networks for multi-media applications such as video on demand.
- The effort required to develop and maintain the ATLAS software will be enormous. Because of the dependence of the whole experiment's success and due to the long lifetime of about 20 years of the project, the software quality requirements will have to be very high. For the whole ATLAS software development, up to 1000 person-years will be required. It appears that the overall manpower is available within the collaboration. A complication is that the workforce is very much spread geographically and that many developers will be students who can spend only a few years in the project.

The ATLAS Software Process

It is expected that about 85% of the ATLAS software effort will be from small separated groups not based at CERN. In order to optimise the quality, to guarantee the long-term maintenance, and to minimise the necessary resources, a well-defined 'ATLAS Software Process' (ASP) [2] has been developed in the framework of the RD41 (MOOSE) project [3]. In the software development we will adhere to accepted international standards; wherever possible we will seek common developments with other experiments and employ commercial solutions. We plan to implement the software following the object-oriented paradigm. Currently, we are studying the implementation using the C++ language. For a project the size of ATLAS we must adopt appropriate engineering techniques for the construction of the software. The key elements of the ASP are

- **Deliverables.** Documents such as requirement analysis dococuments, design documents, project management plans and not least the code are defined in the ASP. Rules for the format of these documents have been defined in order to provide uniformity and maintainability thoughout the different domains.

- **Evolutionary development.** We have currently adopted a working rythm of eight working week cyles. At the end of each cycle there is a new working system developed in a controlled way. After a cycle plan has been established, developers can work independently on implementing and testing classes during a cycle, and then incorporate them for system testing at the end of the cycle.
- **Project organisation.** If the project is to run smoothly, responsibilities must be defined. The chief architect has the task of organizing a small group to define the overall structure of the software and dividing it up into domains corresponding to coherent work packages, some of which might be expected to correspond to the subdetector hardware. Domain architects have the responsibility for organizing a domain team and designing and building the software in their individual domains. The project manager who is responsible for setting objectives, planning the organization of the work, and allocating resources, and a number of more technical ones such as taking responsibility for project documentation or configuration management.
- **Quality assurance.** Three important mechanisms for ensuring and improving software quality are envisaged. The first is the review of deliverables. Starting from the document defining the user requirements, proceeding through various design documents to the final source code, it must be possible to show that each new deliverable satisfies the requirements implied by the preceding document in the sequence. The second mechanism is regression testing ensuring that the performance of the software at the end of a cycle is at least as good as at the previous release. Thirdly, we plan to employ software metrics giving an indication of quality. As the cycles succeed each other, it should be possible to get an increasingly good idea of these measures.

As we gain experience in applying the ASP, in particular in our environment of very small teams working in many different locations, we will have to adjust the software process in order to ensure that it is a benefit and not a burden. The organisation of computing is part of the overall ATLAS organisation. The ATLAS Computing Steering Group (ACOS) deals with offline computing matters and more global computing aspects such as software engineering and computing infrastructure. The chairperson of ACOS is the ATLAS computing co-ordinator. He represents offline computing in the ATLAS Executive Board. In ACOS, the computing representatives of the ATLAS systems (Inner Detector, Liquid Argon Calorimeter, Tile Calorimeter, Muon System, Trigger/DAQ) provide direct contact to their respective communities. The detector communities organise their software work relatively autonomously. The co-ordinators of the major packages such as simulation, reconstruction and trigger simulation integrate the software prepared in these sub-domains. Additional members represent specific geographical regions or computing tasks within the ATLAS collaboration.

The Software Development Environment

The Software Development Environment (SDE) is everything needed on the developer's desktop (CASE tools, testing tools, compilers, linkers, debuggers etc.) in order to participate in the orderly development or modification of a software product. It should be a fully integrated operational software environment, and not just a collection of individual task-oriented tools. A working group has discussed the requirements and has produced a first document listing the functional requirements and making, some initial choices for the Software Development Environment. It has been decided that ATLAS will follow the Unified Notation as soon as the standard has been published officially and tool support is acceptable. Up to that time the OMT notation for diagrams showing static associations between classes (the Object Model in OMT terminology) will be used along with a diagram which shows message flow using the notation supported by the Object Interaction Editor of the StP CASE tool.

Transition from FORTRAN to Object-Oriented Software

We have a detector simulation program based on the GEANT 3.21 detector simulation package and a reconstruction program for the simulated data. All this code is written in FORTRAN 77 and uses ZEBRA for memory management. These programs have been used in the past to study the detector behaviour and to optimise its parameters and have produced all results for the ATLAS Technical Proposal [4] and are being used for the Technical Design Reports of the various sub-detectors. It is foreseen to use and upgrade these programs for at least the next two to three years. The ATLAS detector has been described in the simulation program in great detail, resulting in 11 million GEANT volumes. The simulation of a single di-jet event takes on average 20 minutes on a HP-735 workstation. Between November 1996 and May 1997, about two million jets have been produced. A major challenge will be to design and develop new software while still maintaining the FORTRAN software. We plan to start the development from the proven algorithms, improving them where necessary. We will follow an OO design, implementing in C++. The data, now described in common blocks and stored in ZEBRA banks, will then be encapsulated in objects together with the functions acting on that data. To introduce OO and C++ we propose to follow two lines; one will start from the existing simulation and reconstruction code and make extensions and modifications in C++, and the other starts from scratch with a new OO design. The first line allows implementing new algorithms written in C++ in an established environment, allowing the users to continue to use the services such as input/output and histogramming provided by the framework. In the second line we start to develop a framework and a set of tools for the long-term use in ATLAS. It is hoped that most of the effort invested in the first approach can be reused in the second. It is felt that pursuing these two parallel lines of development minimises the risk of the changeover from procedural to object-oriented programming. It does not disturb significantly the necessary detector studies based on simulation. It allows the development of new FORTRAN code, which follows closely the development of ideas on the ATLAS detector itself.

Current Object Oriented Projects

We have started several projects providing object-oriented software for practical applications in ATLAS. The purpose of these projects is to gain experience with the ASP and to provide hands-on experience with the new style of programming for the ATLAS physicists. The projects have a relatively short timescale of about one year such that they can be used to try out the ASP and provide examples for the ATLAS community to start more software developments in the new style. Examples of such projects are:

- First implementations of the simulation framework and toolset GEANT-4 [5] for ATLAS test-beam configurations and for the full ATLAS detector.
- A tracking package providing pattern-recognition for the inner detector using e.g. Kalman filter algorithms.
- Test beam analysis packages including the use of an object oriented database for event storage.

Computing Model

The ATLAS computing model describes the global architecture of how we plan to use software, processing power, storage and networks to do the offline computing at LHC. The term offline computing encompasses detector calibration and alignment, event reconstruction, Monte Carlo generation, and physics analysis of both real and simulated data. The basic inputs to the model are:

- 100 Hz event rate out of the event filter (Level-3 trigger), i.e. 109 events per year;
- 1 Mbyte event size;

- $\sim$1 Mbyte/h of calibration and alignment data,

leading to $\sim$1 Pbyte (10^{15} bytes) of raw data per year. Our model must take into account both low-luminosity running (10^{33} cm^{-2}s^{-1}), expected at beam turn-on, as well as high luminosity running (10^{34} cm^{-2}s^{-1}). It is assumed that the above data rate remains essentially constant as the luminosity increases by a factor of 10. The event reconstruction must handle the 100 Hz rate out of the event filter. We propose to reconstruct quasi-online allowing for a short ($\sim$ few hours) delay for the generation of the alignment and calibration constants from the data themselves. For the output of the reconstruction, we target an event size of$\sim$100 kbyte, i.e. a reduction of a factor of 10 in data volume. The set of objects produced by the reconstruction is labelled Event Summary Data (ESD). It is anticipated that the reprocessing of events, to account for changes in the calibration, alignment, or reconstruction algorithms, can begin from either the ESD or the raw data. We propose to allocate sufficient resources to reprocess events: a few times per year starting from the ESD, and once per year starting from the raw data. There are five activities requiring access to the data which are of interest for the computing model:

1. Monitoring the detector performance and the data quality.

2. Understanding the detector response: calibration, alignment.

3. Developing and testing of reconstruction algorithms.

4. Studies of rare processes such as high-p_T physics.

5. Studies of processes requiring high statics in the final sample such as B physics or QCD. The monitoring will run in parallel with the event reconstruction to provide rapid feedback during data-taking. The detector response studies need access to a variety of events from calibration to specific physics channels and, as well, need access to both raw and reconstructed data. Similarly, the development and testing of reconstruction algorithms will to a certain extent be done with Monte Carlo events before the beginning of data taking. However, when confronted with real data these algorithms will require tuning, and there will be further developments as new ideas arise. For the physics studies, the computing model must allow efficient access to select and study relatively small event samples embedded in large samples of mostly background. For example, many physics channels consist of 10^7 to 10^8 events, i.e. 1 - 10% of the annual event sample. One can imagine that several groups apply different selection criteria to define 'analysis samples', which are several orders of magnitude smaller. These are then extensively studied; resulting in a new set of selection criteria which is used to repeat the exercise. The information required for physics studies is generally just simple 'physics objects', i.e. electrons, muons, jets, tracks, etc., requiring only a small amount of data per event (relative to the initial 1 Mbyte), estimated to be less than $\sim$10 kbyte. This set

of objects is labelled Analysis Object Data (AOD). An important point for the computing model is that these 'physics objects' evolve with time as the calibration constants and reconstruction algorithms improve. This is particularly true during the early phase of the experiment where the first physics data will be used to understand the detector response. Monte Carlo studies have already been extensively exploited for the design of the detector and trigger and the understanding of test-beam results. Studies will continue during the construction phase, and as well, Monte Carlo generated events will be used to test the offline event reconstruction and analysis software. As the experiment begins taking data, the Monte Carlo will need to be tuned and then used to calculate corrections for physics results. It is estimated that the required number of Monte Carlo generated events is approximately 10% of the number of real events, and corresponds to $\sim$5x10^4 SPECint95 processing power. The low I/O bandwidth required for Monte Carlo generation allows it to be distributed across the collaboration. This effort will need to be organised collaboration-wide, and most likely only the ESD and/or AOD information will need to be made available for general use. Technology is an important ingredient in the computing model, since the offline system which will eventually be designed relies on the capabilities of an underlying technological layer. The extrapolation of cost estimates for networks, storage and computing power is difficult over a time-scale of several years. However, for the requirements of the computing model presented in this chapter, it is reasonable to expect from recent trends that the cost of storage and computing power will have decreased sufficiently for the requirements of ATLAS to be satisfied. The largest uncertainty lies in the affordability of wide area network (WAN) bandwidth, in particular because of the deregulation of the European telecommunications industry and the recent rapid growth in Internet usage. The importance of WAN bandwidth becomes clear when one understands that, in order to analyse the large volumes of data produced at LHC, the processing power is required to be close to the data, and thus analysis facilities will be localised at CERN and possibly a few regional centres. The key software elements, which directly concern the computing model, are the management and the storage of data. Commercial ODBMS and MSS capabilities are currently under study by RD45 [6]; their preliminary results are promising. An ODBMS would serve as a front-end tool where one organises and manages the data from a logical perspective, i.e. one directly manipulates runs of events, individual events, tracks of an event, different samples of events, etc., and the ODBMS manages the physical location of the information, i.e. which part of an event is stored on which file. A MSS would serve as a large bandwidth back-end file server allowing hierarchical storage management of the data, which is transparent to the front-end user. The current view is that the combination of a commercial ODBMS and MSS will manage all of the data for both the event reconstruction and the physics analysis. We expect that there will be $\sim$500 'equivalent physicists' performing some analysis task with $\sim$150 users simultaneously accessing the data. We assume that from start-up all physicists will have adequate access to the data to perform analysis from their home institute. It will be important that there is a coherent view of the data independently of where the data physically resides, i.e. at CERN, at a regional centre or at one's home institute. The question of the rôle of regional centres in the ATLAS computing model has not yet been resolved. It is generally agreed to perform the event reconstruction at CERN. Also, the bulk of the raw data will remain at CERN. Thus, any reprocessing of a large fraction of the raw data will be done at CERN. The rôle of regional centres would be to concentrate on the areas of physics analysis, MC generation, and possibly some of the reprocessing which begins from the ESD information. The information provided in the following sections is intended to begin the preparation for a decision that will be taken by the end of 1998. The participating institutes in ATLAS provide the basic support for their physicists. This includes desktop support and a certain amount of computing power and storage. The rôle of the institute within the ATLAS computing model will also need to be understood. The key point is to provide the resources so that one can perform the required analysis tasks from the home institute. This may include some data that is physically transferred. However, it should be stressed again that the majority of the data will have to remain at the large facilities, i.e. CERN and possibly regional centres.

Cost

A precise cost estimate for the computing hardware is impossible due to the uncertainties in both the requirements and the evolution of the technology and the market. A rough estimate puts the cost of the central installation of data storage and processing power at CERN for ATLAS at about 20 million Swiss francs, to be spent over several years. The cost is dominated by the CPU requirements of $2.5\text{x}10^5$ SPECint95 (10^7 MIPS). Subsequently about 9 million Swiss francs will be needed to expand the facilities to the increasing requirements and for maintenance. To enable physics analysis in a world-wide collaboration, good networking is a necessity. Today it is impossible to predict the evolution of the cost and the performance of international networks at the time of LHC running. As these are important parameters for the precise planning of an analysis scenario, we have to follow the developments and adjust our planning accordingly. Already during the construction phase, we need international networks for document and code exchange as well as for communication such as video-conferencing in order to minimise travel. Currently, in some areas the networks are still insufficient even for code exchange.

Conclusion

Computing for ATLAS is an important and challenging task. We have started to define solutions for the software development using a well-defined software process based on the object-oriented design and implementation. First ideas for a computing model concentrate on the event storage in an object oriented database. Because of the long lifetime of the project, we have to build into our strategy the flexibility to cope with the rapid evolution in the field of computing and with the changing requirements of the experiment.

References

[1] The ATLAS Collaboration, ATLAS Computing Technical Proposal, CERN/LHCC/96-43

[2] S. Fisher, The ATLAS Software Process, Presentation at CHEP97

[3] K. Bos, RD41 or MOOSE Collaboration, Presentation at CHEP97

[4] The ATLAS Collaboration, ATLAS Technical Proposal, CERN/LHCC/94-43, LHCC/P2

[5] S. Giani, GEANT4: a world-wide collaboration to build Object Oriented HEP simulation software, Presentation at CHEP97

[6] J. Shiers, RD45 Status Report, Presentation at CHEP97