Chapter 14 Computing

14.1 Data-flow model

The data produced by the detector and by Monte Carlo simulation undergo a number of transformations before reaching the desktops of the physicists performing the analysis. Each transformation places specific requirements and constraints on the system. For example there are requirements on processing power, data storage and network bandwidth, and constraints from the availability of calibration and alignment data, or on the quality and reliability of the algorithms.

The data flow model of the LHCb experiment is shown in Figure 14.1. The model is similar to that of other HEP experiments. There are however certain features which are particularly relevant to LHCb:

- The Level-2 and Level-3 software triggers must achieve rate reduction factors of 8 and 25 respectively. This will require sophisticated algorithms similar to those employed in the full reconstruction, but which must be fast enough to cope with the expected input rate and highly reliable since only the data passing these triggers will be written to permanent storage. Calibration and alignment constants will need to be of sufficient quality to allow the triggers to work efficiently, implying that at least some part of the calibration and alignment must be performed in real time. The full reconstruction will make use of the results of the trigger algorithms.

- The raw data is written to permanent storage at a rate of 20 MB/s. The reconstruction program adds a similar amount of data to the data store, at a similar rate. Assuming a data-taking year of 10^7 seconds, the permanent data store will have to be capable of accepting in excess of 500 TB of data per year at a rate of 40 MB/s. This is 2-3 orders of magnitude greater than current experiments, and similar to the other LHC experiments [1].

- Nearly all processes in the diagram access at least some fraction of the data store. This is a database of enormous size and complexity, which will need to be designed, maintained and populated in such a way that all processes will have transparent access to the necessary information.

14.2 Computing requirements

The tasks identified in the data flow model are analysed by breaking each of them down into subsystems, such as individual subdetectors for the simulation, subdetectors plus tracking, calorimetry and particle identification for the reconstruction, and individual physics channels for the analysis. For each subsystem, studies are made of existing simulation, reconstruction and analysis algorithms to measure processing power requirements, to understand the types and volume of data required as input and produced as output, and to evaluate the dependence on calibration and alignment data.

The measurements are extrapolated to the final system by making reasonable assumptions on the evolution of the algorithms and on the expected luminosity and trigger efficiency. Where suitable algorithms do not yet exist, such as for pattern recognition in the tracking system or for all but the simplest analyses, estimates are based on extrapolations from existing experiments. This approach should ensure that the estimates of data storage, processing and networking needs are realistic. It should give a detailed understanding of access patterns to the data and of the data flow logistics, allowing reasonable constraints to be placed on the computing model.

The current processing power and data storage requirements of LHCb are summarized in Table 14.1 and Table 14.2 respectively, and compared with estimates of the annual requirements during data taking. The detailed breakdown of the estimates by computing task can be found in [2]. The figures for reconstruction assume two passes through the real data in order to extract calibration constants etc. One complete re-processing of all the real data per year is also assumed. It should be noted that the total CPU requirements do not

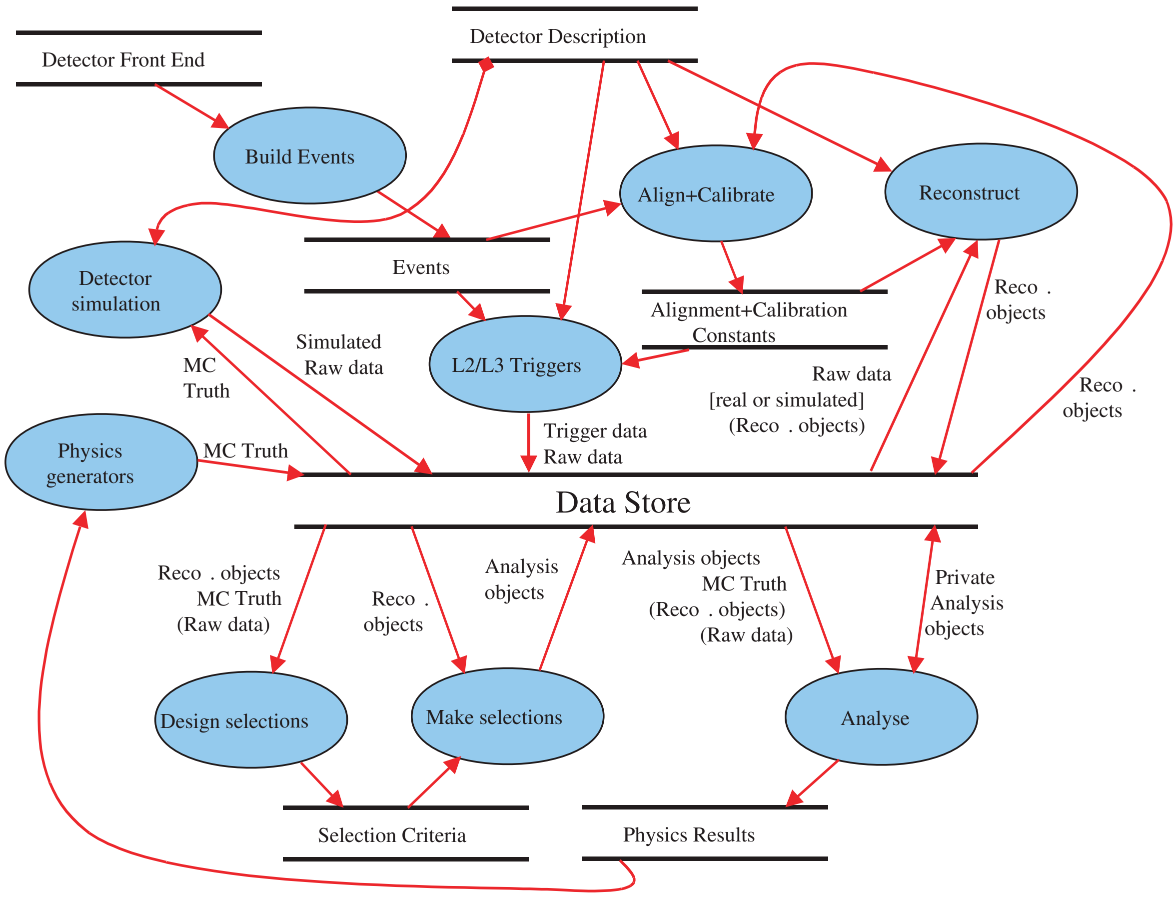

Figure 14.1: LHCb data flow model.

necessarily map to processors installed. Only a detailed computing model will tell how exactly the mapping of requirements to processors will be done. For example, some of the CPU power installed for triggering could be used for re-processing outside data-taking periods.

14.3 Computing infrastructure

Until very recently LHCb relied almost entirely on public Unix services at CERN for its computing needs. In the last few months access has been gained to regional facilities outside CERN and to the public PC simulation farm (PCSF) at CERN for large Monte Carlo productions. Although processing requirements are expected to grow by $\sim 50\%$ per year until the end of 2000, it should be possible to satisfy these needs by use of these public facilities during this period. The simulation program will produce sequential data until the end of 1999, and therefore investment in object oriented

Table 14.1: Processing power requirements

Processing type	CPU used (MIPS, 1997)	CPU needed (MIPS, 2006)
Triggering		
Level-2	-	4×10^5
Level-3	-	1×10^6
Reconstruction		
Real Data	-	1.6×10^6
Sim. Data	200	2×10^5
Simulation		
B-meson	600	1.2×10^6
Background	-	1.2×10^3
Analysis		
Real Data	-	5×10^4
Sim. Data	150	5×10^5
Total	950	$\sim5\times10^6$

Table 14.2: Data storage requirements

Storage type	Storage used (TB, 1997)	Storage needed (TB/year, 2006)
Raw	0.01	200
Monte Carlo Raw	1.0	100
Calibration & Alignment	-	< 1
Reconstructed (Raw+MC)	1.0	200
Analysis	0.05	$\sim$2
Total	$\sim$2.0	$\sim$500

databases is not expected before 2000.

As test beam activity increases and Technical Design Reports are prepared, a more significant increase in CPU needs is anticipated. This will be the appropriate time to begin investing in private facilities. Investment in full scale facilities will be left as late as possible, to benefit fully from continuous improvements in the price/performance ratio of computing equipment.

The main issues relating to infrastructure that must be addressed during the preparation phase are:

- The ability to build and operate large scale computer farms containing several hundred CPUs.
- The handling of petabytes of data stored in a central database.
- The access to the data from HEP institutes outside CERN.

These issues are of common interest to all LHC experiments, and are discussed in detail elsewhere [3]. Current thinking is that experimental data in the LHC era will be logically stored in a single, experiment-wide Object database, with the physical storage managed by a mass storage system, as currently being investigated by the RD45 [4] project. The distribution of storage and computing resources at CERN and amongst regional centres will be optimized, in the light of networking, political and sociological constraints.

The mass consumer market will have a big impact on the provision of the necessary computing resources at an affordable cost. PC technology is evolving rapidly, and industry standards must be followed closely to avoid making inappropriate choices that might result in a premature commitment to an obsolescent platform, language or operating system. None of the requirements can be considered outside the limits of current technology trends.

14.4 Software strategy

The LHCb approach to managing the production of software is outlined below. More detailed information can be found in [5].

14.4.1 Objectives

The strategy adopted for ensuring good software quality will have important consequences for the success of the experiment. Intensive use is made of software in the trigger, and the event reconstruction and selection algorithms must be implemented correctly and be fully tested by the time first colliding beams are delivered. Bulk processing of TByte data samples require large CPU facilities, and reprocessings will be costly and must be minimised.

Software quality can best be ensured by using modern engineering practices, modern technologies and by putting in place sound management practices for controlling the development process. Effort will be put into building high quality software components, and the extra investment in manpower that this represents will be recuperated by extracting as much use as possible out of these components in all application domains. Commercial products will be used whenever appropriate technically, and providing they can be licensed to all collaboration institutes.

14.4.2 Technology

The software industry has invested heavily in the development of Object Technologies. A good level of maturity has been reached in the definition of standards and commercial products, such as databases and development tools, are becoming widely available. Supported features, such as patterns, components and frameworks, will provide opportunities for sharing concepts, designs and code across projects and will therefore help to achieve the re-useability goals already mentioned.

The most widely used OO language today is C++. However C++ is known to have deficiencies and other languages could be more appropriate providing they become widely used and attract industrial support. Java is an example today of a new language with this potential. The two most popular design methodologies, OMT and Booch, are converging to define a single method, called UML, incorporating the best features of both. UML is supported, at least partially for the moment, by the major CASE tools, such as Rose and StP.

At this moment a combination of C++ and UML offers a robust and well supported software development environment. However it is important that the architecture and design of the software is independent of the language and design tool, since changes in the choice of methods and tools may become necessary.

14.4.3 Migration policy

The process of understanding an object-oriented language is typically quite time-consuming and it can take several months to become an effective OO developer. The transition to OO programming is one of the most important challenges we face in the immediate future and its introduction requires careful planning.

At present the LHCb development environment is based on FORTRAN. The simulation program is the basic tool for making detector optimisation and physics performance studies, and this uses GEANT3 as a framework. Tools for reconstruction and analysis written in FORTRAN also form part of the maintained suite of software. It is our intention to re-engineer these codes using Object Technologies. There is an incentive to make this transition as soon as possible in order to minimise the amount of "legacy code" that has to be maintained.

In the first instance, the language, tools and programming environment will be selected and a suitable infrastructure (i.e. user interface, data management tools etc.) assembled such that application development can begin. Much of the impetus for this work will come from the release of the LHC++ library[6], which implements some of the key HEP specific toolkits for data analysis, a new detector simulation framework (GEANT4) and object-persistency services based on a commercial ODBMS (Objectivity). Using these components, new simulation, reconstruction and analysis frameworks will be developed. Once the development tools and frameworks are in place, a rigorous training programme will be organised for collaboration members. At this time, further developments of the existing FORTRAN code will be frozen, and it will be discarded altogether as soon as the new OO based programs are put into production. A detailed plan exists showing details of the tasks and their timescales[7].

14.5 Project organisation

All computing related activities will be managed within one project framework (Figure 14.2). The scope includes on-line and off-line systems and covers both hardware and software aspects. The aim of this approach is to minimise barriers between different activities and encourage the sharing of concepts, tools and software. Teams dedicated to support software engineering and computing facilities will provide the environments used to develop the software and the computing infrastructure. The development of software of common interest to all applications will also be done by a dedicated team. The Computing Project will be managed by a Steering Group, comprising the Computing Coordinator and the leaders of the various subprojects.

14.6 Project management

The lifetime of the LHCb experiment, from its conception to its completion, spans more than 15 years. The main phases and activities during this time are summarised below. More detailed information on current planning, including task schedules and manpower estimates can be found in [7].

Preparation phase (now – 2000) Functional specifications of all components will be developed. Knowledge and experience of relevant technologies must be accumulated, and in some cases detailed evaluations made through prototyping and simulation. This will be an intense period of learning and will involve formal training. Public computing facilities will be used for the processing of simulation and testbeam data during this period.

Implementation phase (2001 – 2003) Final choices will be made on software and hardware technologies, and all sub-systems developed. Reviews will be made of the architecture and design of software in the light of experience obtained during the preparation phase. Hardware components will be built and tested, and installed in the experimental area. During this phase, data processing needs will increase sharply, due to increased test beam activity, and investment in private CPU capacity will be needed.

Commissioning Phase (2004) A significant period of time must be allocated for the assembly

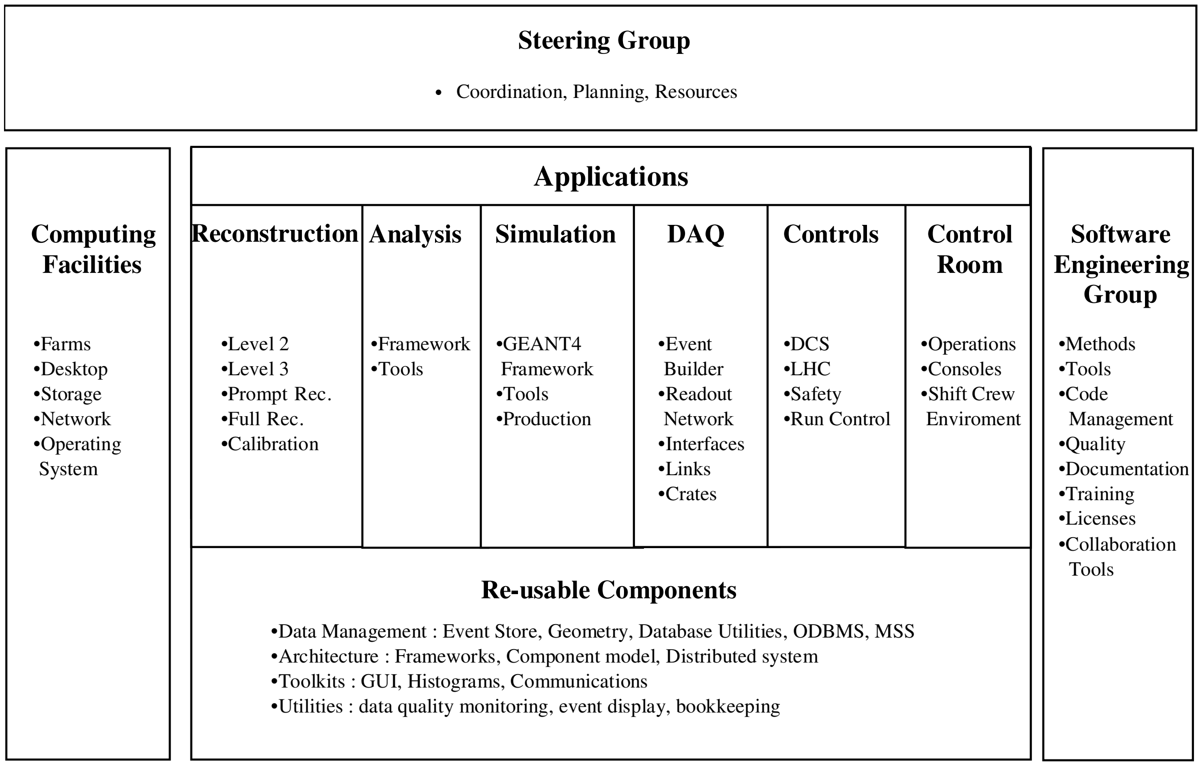

Figure 14.2: Breakdown of Computing activities into subprojects.

of all the various computing subsystems, and for thoroughly testing them under realistic data taking conditions, since the detection and elimination of faults can take a considerable time. The software must also be tested in advance to ensure that the event selection mechanisms and reconstruction algorithms are working correctly, and that the expected data volumes and processing loads can be sustained. Operational experience must be obtained with large scale CPU farms. The commissioning phase must therefore start towards the beginning of 2004.

Operation Phase (from 2005) This starts when beams are delivered and involves maintenance of the computing environment to match evolving requirements and to keep up to date with technology changes.

References

[1] ALICE Technical Proposal, CERN/LHCC/95-71; ATLAS Computing Technical Proposal, CERN/LHCC/96-43; CMS Computing Technical Proposal, CERN/LHCC/96-45

[2] P. Binko,“LHCb Computing Tasks”, LHCb/98-042 COMP

[3] M. Frank, A. Pacheco,“LHCb Computing Model”, LHCb/97-046 COMP

[4] RD45 Collaboration,“Object Databases and their Impact on Storage-Related Aspects of HEP Computing”, CERN/LHCC 97-7

[5] E. van Herwijnen,“LHCb Software Strategy”, LHCb/98-025 COMP

[6] LHC++ , Version 1.0 Users Guide, 3 Nov 1997

[7] J. Harvey, LHCb Project Plan for Computing Technical Note, LHCb/98-026 COMP