CMS Computing Model

The CMS Collaboration

presented by M. Pimiä, CERN, Geneva, Switzerland

The CMS Computing Model represents the architecture of a system of four interconnected resources: computing and network hardware; data; software; and people. These must function effectively in the context of more than a billion physics events per year, more than a thousand physicists located at over one hundred institutes, with a detector and physics complexity unprecedented in High Energy Physics. The model described is flexible, in order to be able to cope with the rapid pace of technological advances and the long lifetime of the experiment. The scale of the model underlines the fact that computing and software constitutes a major subsystem of CMS.

1 Introduction

Computing and software are of paramount importance to the Compact Muon Solenoid (CMS) experiment [1]. As both the detectors and events recorded in high energy physics experiments have become more and more complex, larger and higher levels of collaboration have become necessary. Computing has been the enabling technology for widely spread collaborations as well as being a major detector subsystem of its own. Appropriate and timely software investment is essential for the optimisation of the detector hardware design and the realization of the experiment's potential for physics discoveries and precision measurements. Networks are vital in allowing physicists located throughout the world to be full participants in the experiment, involving such activities as communication through tele-conferencing, synchronisation of software development efforts, high bandwidth access to experimental data, and access to shared databases, documentation, and physics results.

The computing models we present are architectural level models describing methods for managing our resources and meeting our hardware, data storage, software, and physics analysis needs. The models will be flexible enough to respond to changes in technology and to availability of resources, allowing us to choose the optimum deployment of computers, storage systems and networks.

Preprint submitted to Elsevier Preprint *2 June 1997*

Clearly models for hardware alone are not sufficient – many years of intensive software development will be needed to deal with the complex problem of detecting the sought-after signals in the extremely demanding background conditions of the LHC.

The overall complexity and scope of the task makes it crucial that CMS have a cost-effective and flexible plan to manage its computing activities. The CMS Computing Technical Proposal [2] presents the elements of this plan.

2 Overview

The CMS computing model presents the physicist with a simple logical view of all objects needed to perform physics analysis or detector studies. Objects can respond to queries and must thus have access to both data and code (methods). Raw data or DAQ objects have access to trivial methods able to answer questions such as *"what are the ADC counts in the ECAL endcap crystals for this event?"*. Reconstructed objects can answer questions such as *"what is the 4-momentum of this muon candidate?"* or perhaps *"what are the track segments that belong to this muon candidate?"*

Most physicists require access only to high-level reconstructed objects such as the muon candidate, provided that the reconstruction was performed with the best methods and calibrations for the physics under study and that no detector malfunctions or idiosyncrasies are suspected. The CMS computing model should ensure efficient access to high-level objects while recognising that only DAQ objects and calibration-alignment-status objects are not re-computable and thus are the only objects which *must* be stored.

Answers to questions such as 'should the DAQ objects be replicated?', or 'should the muon-candidate objects be stored on disk at my university?' should not be input to the computing model. The model and its implementation should be sufficiently flexible to allow such decisions to be made and changed, perhaps even automatically, during the running of CMS. This flexibility will permit an optimised use of resources even if the relative costs of the networking, storage and CPU components of the worldwide CMS computing environment vary greatly during the preparation and operation of the experiment. The fundamental requirements of the computing model are very simple:

(i) objects which cannot be recomputed must be stored somewhere;
(ii) physicists at any CMS institute should be able to query any objects (re-computable or not re-computable) and retrieve the results of the query;
(iii) the resources should be used as efficiently as possible.

For over two decades the ideal data analysis environment, in which all data are randomly accessible and any useful query completes in seconds, has been unthinkable for any large HEP experiment. More detailed 'requirements' or real implementations have taken into account technical and financial realities. For CMS also, any progress beyond the three bland requirements above requires guesses about the technologies that will be available and affordable.

3 The CMS Computing Model

The CMS computing model should allow CMS physicists to take full advantage of all progress in networking, hardware and software technology which occurs in the next 20 years. This necessarily means that today's tentative form of the model must be forward looking and judiciously vague.

The driving force behind the current form of the model is recent advances in commercial object database management systems. We believe that these advances will allow CMS physicists to be freed not only from needing to know where objects are stored, but also from needing to know whether the objects exist at all, provided they are computable from other existing objects. The infrastructure required to support physicists desktops will be a geographically distributed object store whose coherency is guaranteed by the ODBMS. Servers will perform on-demand computation of objects which do not already exist or which are more cheaply re-computed than accessed over wide-area networks.

It will be very important to define classes of objects for use in reconstruction and physics analysis. However, rigidly defined physical collections of these objects called 'DST', 'mini-DST', 'micro-DST' need not be planned. Effectively, the contents of the virtual CMS 'micro-DST' will change dynamically in response to the queries made by physicists.

We expect that the ODBMS will provide most or all of the tools needed to optimise the use of the distributed object store but that the ODBMS itself will not be able to make all the decisions. For example, CMS physicists know that the raw data objects will be valid for all time whereas track fits performed by the level-3 trigger may be superseded within hours. The decisions on the parts of the database which should be systematically replicated, such as the Event Tag Database (ETDB), may well be made by physicists.

3.1 The Physicist's Desktop

CMS desktops will be equipped with the commodity devices typical of any desktop from which access to information technology is a vital component of the job. CMS should orient itself firmly in the direction of mainstream commodity hardware and operating systems. This strategy will provide the hardware and operating system support for an excellent user interface at a cost small compared with the cost of employing even a graduate student.

Interactive devices on the desktop will certainly contain processors that are powerful by today's standards. However, it is not certain that desktop devices will contribute significantly to the computation of physics objects, either for CMS as a whole or even for the user of the desktop device.

3.2 The Network and the Role of Regional Centres

The CMS computing model must assume the existence of physicists' desktops and of data storage and processing facilities at CERN. These two components of the model will have vital roles whatever happens to wide area networking. In contrast, the roles or even the existence of regional centres are dependent on the bright or bleak future of networking.

In the limit of very cheap networking, CMS would be free to locate the components of the computing environment wherever they could be most efficiently operated and funded. There would be no technical justification for regional centres, but equally there would be few technical barriers to the siting of computing resources for CMS at regional computer centres.

In the limit of very expensive networking, the valid model would be centralisation of data-storage and processing with only results being transmitted to physicists' desktops. In this extreme also, regional centres bring no advantage although they could be exploited with minimal damage if each contained a complete 'vertical slice' from raw data to physics objects for some agreed fraction of the data and offered to manipulate these data for all of CMS.

Both these extremes are considered unlikely. Our guess is that by the start of data taking CMS will be able to transfer about 100 Mb/s over each major network path. Within a few years this should rise to over 400 Mb/s which is sufficient to transmit 1 PB in a year. These predictions may easily be wrong by a factor of ten in either direction, but such variations are within the elasticity of the CMS computing model. Regional computing centres will bring significant benefits whenever the cost of desktop-regional centre communications would be much less than that of desktop-CERN traffic.

3.3 Level 3 Trigger Processing and Data Storage

The mission of the level 3 trigger is to reject unwanted or less interesting events. Although the level 3 trigger farm will often have spare capacity, its mission necessarily makes it an unsuitable resource for other time-critical activities. We expect to use the trigger farm outside data-taking periods for Monte-Carlo simulation and tactical re-processing.

The CMS data acquisition must write the event data and alignment-calibration-status information into the CMS data store since these data are impossible to re-create. Other reconstructed objects generated by the level-3 farm do not *have* to be stored since they are re-calculable in principle.

3.4 The CMS Data Store at CERN

The physical components of the object store are likely to be robotic 'tape' storage with a capacity of several PB and a disk cache of several hundred TB. Physicists will access objects via the Object Database Management System that will be automatically available from CMS software. Neither the tapes nor disk files should be accessed explicitly.

Whenever possible, the CMS object store should ensure that needed objects are cached on disk. If retrieval of a tape-resident database is needed, the tape should be a rich source of wanted objects. To achieve these goals, disk-resident and tape-resident objects will have to be 're-clustered' and perhaps replicated. The ODBMS is expected to support coherent replication of the 'hottest' objects at several centres.

3.5 CPU Servers at CERN

Although the CMS computing model is intended to maximise flexibility through its support for on-demand processing, the perceived responsiveness of the system will rely partly on the prior creation, by serial processing on relatively cheap CPUs, of objects which are certain to be required. The Event Tag Database (ETDB) is the prime example of a collection of objects which must be created before further analysis can proceed.

On-demand or formally planned computation of reconstructed objects to satisfy physicists' queries will require computing systems with excellent access to the CMS object store and having a CPU capacity at least as great as the Level-3 trigger. A major logical function of the offline CPU servers will be to cooperate with servers at regional centres or universities to keep objects which may be needed in parts of the object store which are close to the likely users.

3.6 Regional Centres

Regional centres, as part of the distributed computing system, will be most effective if they are invisible. Physicists will be delighted if a regional centre doubles the speed of response to a query, but they will be dismayed if use of the centre requires a different approach from use of the offline CPU servers at CERN.

Although the bandwidth of an airliner loaded with tapes is known to be very large, the latency of such a data distribution process is high. Perhaps the most useful tapes to ship are those containing selected raw data objects, which have the outstanding advantage of being valid for all time regardless of improvements in calibrations and reconstruction algorithms.

A limited number of regional centres will be encouraged to attain high visibility within the worldwide effort to develop and support CMS software. No single site will be able to provide sufficient manpower to develop even the framework of the CMS software which must therefore be the responsibility of CERN together with at least one regional centre.

3.7 Systems at Universities

At any university where more than one physicist works on CMS analysis it will probably be advantageous to provide a local shared object cache with exactly the same behaviour as the object cache at a regional centre. The existence of this component of the distributed CMS object store will make it attractive to connect a CPU server directly to the cache rather than relying on slower network access from desktops.

4 The CMS Software Model

4.1 Overall Software Strategy

During the last three decades High Energy Physics has often been at the leading edge of computing technology. This has led to the development of in-house solutions for a large range of software components including data handling, memory management, and even database systems. These products are, however, rarely robust enough to survive for the lifetime of a LHC experiment.

The large commodity market for both hardware and software computing has made the price of industrial-strength software components comparable or even much cheaper than in-house solutions. Therefore, except for some specific applications, such as simulation and reconstruction, we anticipate that most of the CMS software environment will be composed of commercial solutions including widely used and well-supported freeware or shareware.

Modern languages like C++ make it easier to share and reuse code, and enable the use of software development methodologies that can meet our goals for software reliability. CMS has adopted C++ as our current standard programming language and object orientation as our preferred methodology, while recognising that even better tools may emerge before LHC turn-on.

Finally, by organising our software into a hierarchy of different categories, we can support different styles (and even languages) for different software components. Simple single user programs can be developed without constraint, and only when programs are intended for the central CMS reconstruction must they conform to rigorous software standards.

4.2 Software Architecture

The overall design of the CMS Software Architecture is motivated by the following underlying principles:

- Multiple Environments: Various software modules must be able to run in a variety of environments as different computing tasks are performed;
- Migration between environments: a particular software module may start out being developed for one environment, then later used in other unforeseen environments as well;
- Dispersed code development: the software will be developed by organisationally and geographically dispersed groups of part-time non-professional programmers, primarily graduate students and other physicists. Only portions of the code will be written by computing professionals;
- Flexibility: not all software requirements will be fully known in advance. The software systems must readily adapt to changing conditions;
- Ease of use: the software systems must be easily usable by collaboration physicists.

These principles result in the following overall structure:

- An application framework customisable for each of the computing environments;
- Physics software modules with clearly defined interfaces that can be plugged into the framework;
- A service and utility Toolkit that can be used by any of the physics modules.

4.2.1 Application framework

The framework will do the program control, module scheduling, and I/O. It will be customised and optimised for the task at hand, and most likely will be written by computing professionals (or physicists acting in that role) using disciplined software engineering standards. The tasks that the framework should be able to handle are:

- program development: emphasis on interactivity and access to small numbers of events;
- calibration and monitoring: moderate interactivity, high performance serial access to full runs of calibration or special data samples;
- event analysis: reasonable interactivity, rapid access to random selections of data drawn from large numbers of runs, excellent data presentation tools;
- event reconstruction: batch processing, high performance serial I/O for entire runs, well controlled versioning and software engineering;
- level 3 reconstruction: real time processing, high performance I/O, rigidly controlled versioning and software engineering;
- simulation: limited interactivity, highly compute bound jobs.

4.2.2 Physics modules

These will be written by detector groups and physicists. The modules can be plugged-in to any of the application frameworks at run time. One can easily choose between different versions of various modules. The physics modules will not communicate with each other directly but only through the data access protocols that are part of the service layer below. The level of software engineering required will depend on in which environments the modules are certified for use.

4.2.3 Service and utility Toolkit

It will consist of two major categories of services: physics type services (histogrammers, fitters, physics calculation routines) and computer services (data access, inter module communication, event input and output, *etc.*).

5 Summary

The main features of our proposed solutions in the CMS Computing Model are summarised below:

- Hardware: Our hardware solution is characterised by large collections of CPU, disk and robotic mass storage at CERN and a small number of regional centres. We do not give precise specifications for the division of hardware now, but rather a recipe for optimising the distribution of hardware at a later date based on hardware availability at different CMS institutions, network costs, and available bandwidth. The hardware arrangement will be based on the principles of executing computing tasks where the data reside and of providing equivalent access to CMS physicists independent of their location.
- Data: Our data access solution is based on a new paradigm. Instead of relying on serial access to increasingly refined data sets (DST, mini-DST, etc) we will access all data (raw, reconstructed, and physics data) from a common object store through common mechanisms. Our data storage systems will automatically optimise data storage, based on access patterns, to retrieve the data most efficiently.
- Software: Our software solution is based on a hierarchy of types of software, with a professionally engineered framework into which physics modules may be inserted; on a greater reliance on software engineering to cope with the increasing complexity; and on modern programming methods, languages, and tools. In particular this implies, at least for now, the use of C++ and an Object Oriented paradigm.
- People: We must optimise the productivity of our highly dispersed software development effort and maximise the quality of the software produced. This will be achieved using formal methods for code design, production and distribution, and with tools that facilitate collaborating on software projects from a distance.

References

[1] CMS Collaboration, *Technical Proposal CERN/LHCC 94-38* (1994)

[2] CMS Collaboration, *Computing Technical Proposal CERN/LHCC 96-45* (1996)