THE BABAR EXPERIMENT - COMPUTING ISSUES FOR THE YEAR 2000

D. R. QUARRIE

Lawrence Berkeley National Laboratory, MS 50B-3238,
1 Cyclotron Road, Berkeley, California 94720, USA
DRQuarrie@LBL.Gov

The *BaBar* experiment at the PEP-II asymmetric e^+e^- Collider at SLAC is due to commence data taking early in 1999. This talk will briefly describe the detector and its physics goals in order to set the scale of the computing problem. Approximately 10^9 events, corresponding to 80 TBytes of information, are expected to be accumulated per year. In addition the collaboration is large and geographically dispersed - almost 500 physicists from 80 institutions in 10 countries. Thus the computing issues break down into two categories - technical and sociological/managerial. Both of these will be discussed. The major focus will be on the work that is underway to establish a distributed software development environment, the computing model that is the baseline plan for dealing with the onslaught of data, and more speculative avenues that are expected to be explored in parallel with this baseline.

1 Introduction

The *BaBar* experiment [1] at the PEP-II asymmetric e^+e^- Collider at SLAC is due to commence data taking in 1999. The primary physics goal is the systematic study of *CP* asymmetries in the decays of B^0 and $\overline{B}^0$, and secondary goals include studies of charm, τ, two-photon and Υ physics. Some characteristics of the experiment are summarized in Table 1.

Table 1: Characteristics of *BaBar* Experiment

Characteristic	Size
No. of Detector Subsystems	7
No. of Electronic Channels	~220,000
Raw Event Size	~25 kBytes
DAQ to Level 3	2000 Hz 50 MBytes/sec
Level 3 to Storage	100 Hz 2.5 MBytes/sec
Storage Requirements (includes Monte Carlo)	10^9 events/year ~85 TBytes/year
CPU Requirements	~20,000 MIPS
Collaboration	~480 physicists ~77 Institutions 10 Countries

2 Preparing for the onset of data

Computing issues that are of critical importance are both technical and sociological in nature, the former resulting from the complex detector and the high physics interaction rate, the latter from the large and geographically dispersed collaboration. The rapid establishment of a computing management structure, a productive collaborative software development environment and a model of a computing infrastructure that will provide the storage, bandwidth and processing power requirements is essential.

2.1 Management Structure

The *BaBar* Computing System is treated on an equal footing to the other detector systems such as the Vertex Detector and Calorimeter. It is managed by a team of a System Manager (actually two co-Managers in this case) and four Deputy System Managers for Reconstruction, Simulations, Online and Tools & Infrastructure. A Project Computing Engineer has overall technical oversight. The philosophy is that anyone writing software is a member of the Computing System and therefore many people will have a dual allegiance to another detector system and to Computing.

2.2 Software Methodology and Programming Language

An object oriented methodology has been adopted by *BaBar*, based on the Booch notation [2] and using the Rational Rose/C++ CASE tool. The C++ programming language has been selected as the recommended programming language. However, there is recognition of the widespread expertise in Fortran and some continued development in this language will have to be accommodated. Note that the choice of programming language has been phrased as a recommendation rather than a decision. This is a subject where people have very strong views over a wide spectrum. The goal is to achieve overall a higher level of productivity and to produce a more robust system. The adoption of the OO methodology in conjunction with C++ is intended to achieve this, albeit at the expense of a short-term loss of productivity and the need for additional training.

A significant effort has gone into producing a set of guideline documents, code skeletons, example programs and tutorials as an encouragement for large-scale adoption of this recommendation. A quality assurance (QA) strategy is still being established, but it will recognize the need for different levels of QA appropriate for the different types of software. Physicists' code for their own private analyses will require less QA than code in the production environment or within the online data acquisition and trigger environments. Migration of code from one environment to another also has to be addressed.

2.3 User and Developer Environments

There is a tension between the needs of a code user and code developer. The user wants the illusion of a stable system; even if there are bugs they should be *stable* bugs. The developer wants unrestricted access for enhancements and bug fixes. This is complicated by the distributed nature of development. Most people are simultaneously users and developers in different areas of the overall software and their role will change as a function of time throughout the duration of the experiment. The software development environment must also provide stability in this highly dynamic situation.

The Concurrent Version System (CVS) has been selected as the basis for the software development environment. The rCVS system has been developed to extend access to CVS from a distributed developer community. However, many sites are currently investing in the AFS distributed filesystem and this has been adopted as a standard within the Collaboration. A migration from rCVS to a combination of AFS and the client-server version of CVS is planned. Poor network connectivity between the USA and Europe is the main limitation of this environment.

In parallel to this, a software release environment has been established. This is based on the concept of independent software *packages* that are tagged by their developers at convenient intervals, thus forming a *version*. Snapshots spanning all packages form a *release*, and are made at a frequency that is typically lower than that at which package versions are being made. A release forms an island of stability for users and several releases can be supported simultaneously.

2.4 Chosen Technologies

Technologies that have been selected by *BaBar* include the following:

- Unix and VxWorks operating systems. A limited set of versions of Unix will be supported, including IBM AIX, DEC Unix, HP-UX and Solaris.
- Tcp/ip networking. A FDDI switch is used as the baseline Event Builder architecture. Ongoing R&D is evaluating alternatives (ATM, FibreChannel).
- The EPICS controls system will be used within the online environment.
- X-Windows, Motif and Tcl/Tk for user interfaces. The use of PEX and OpenGL and User Interface Builders is still under investigation.
- Databases. Oracle is in use for the Collaboration database, but an OODBMS solution is being investigated for the geometry, alignment and calibration databases. Financial concerns are an issue here.
- The CORBA distributed architecture is being evaluated as a component within the online system. A R&D project has demonstrated C++ to FORTRAN 90 integration based on code generation from an OMG Interface Definition Language compiler [3].
- Electronic mail and mailing lists are used extensively, as are tools based on the World Wide Web (WWW). WWW tools are used for documentation, discussion groups, email archives, the minutes of meetings, advertising job openings within the Collaboration, online registration for Meetings, remote access to databases and more.
- Significant use is made of dial-up video conferencing and conference phone calls. There has been much less use of desktop video so far.

3 Computing Model

The overall computing model is of a three-tiered hierarchy:

• *Collaboration Center at SLAC*. This is obviously the location of the experiment and the online system. It contains the primary repositories for code, documentation, the raw data and databases. It is foreseen that bulk processing of the data to an intermediate DST form will take place here.
• *Regional Centers*. Several of these will exist (*e.g.* RAL in the UK and CCIN2P3 in France) and will act as secondary repositories for DST data and cloned copies of the code, documentation and databases. They will act as possible sites for bulk Monte Carlo processing.
• *Home Institution*s. Many of these will exist and may chose to use a Regional Center or SLAC as their primary repositories. Much software development and physics analysis will take place at these locations.

3.1 SLAC Collaboration Center Architectural Model

The architectural model for the SLAC Collaboration Center is shown in Figure 1. Highlights are:

• Sufficient processing power and storage to perform a pseudo real-time reconstruction pass with a latency of minutes.
• Mass Storage based on helical scan technology (20-50GB per cartridge in 3480 format) but with cheaper distribution media (*e.g.* DLT, 4mm, 8mm).
• A Reconstruction and Monte Carlo Farm based on multiple Unix processors and relatively low bandwidth links.
• An Analysis Farm based on multiple Unix processors and high bandwidth links.

A prototype of this Model is being already established at SLAC. By the end of 1995 it will have a total processing power of about 8,000 MIPS and access to a 24TB robotic tape silo.

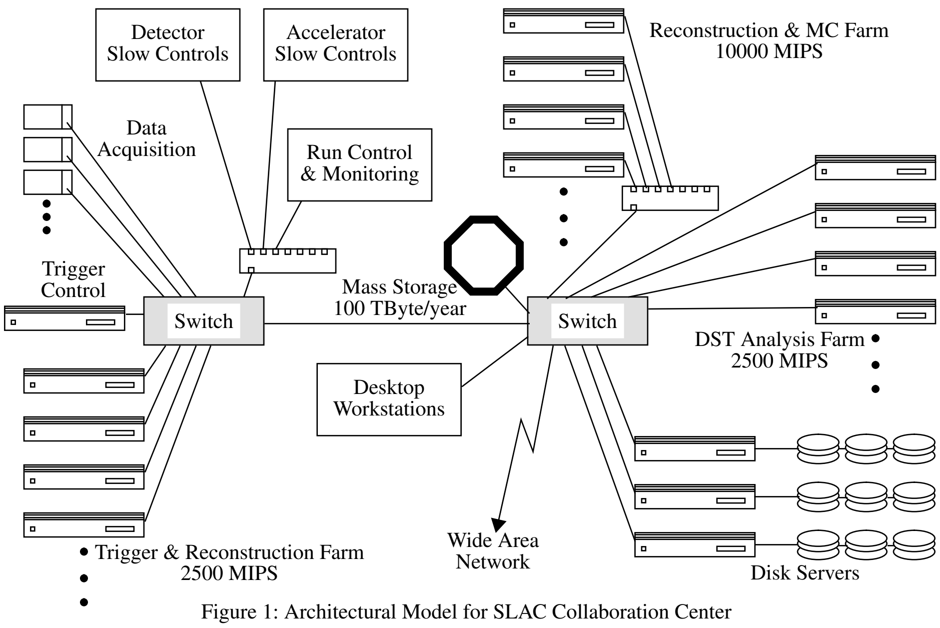

Figure 1: Architectural Model for SLAC Collaboration Center

4 Possible Enhancements to the Baseline Plan

Several R&D projects are either already underway or a being planned that will significantly enhance the baseline plan. These include:

- Database storage of event data instead of conventional sequential files. This reflects the concern that conventional techniques will be unable to cope with 10^9 events per year.
- More widespread use of CORBA as a possible component of distributed database access and in order to provide a distributed application framework.
- A Hierarchical Storage Manager (HSM) providing a virtual file system with automatic file migration instead of a conventional disk staging system.
- Alternative Unix cpu technologies such as the SGI Challenge or IBM SP-2.
- The personal viewpoint of the author is that the impact of Windows-NT should be evaluated. Such systems are generally extremely cost-effective, especially when software costs are taken into account. A low level R&D project has already demonstrated portability between Unix and Windows-NT of some *BaBar* developed code.

5 Summary

The *BaBar* experiment is scheduled to come online just prior to the next millennium. It is of sufficient scope to be extremely challenging from the computing viewpoint, with significant demands on processing power, bandwidth and storage. The large and geographically dispersed software developer community is also challenging. The rapid establishment of a management team and distributed development environment is crucial.

A decision has been made to based the software development on the object oriented methodology and, primarily, the C++ programming language. Much effort has been devoted to provide a productive development environment in support of this decision.

The overall computing model is fairly conventional, but it is being augmented by several R&D projects aimed at addressing areas of significant concern.

Acknowledgments

This work was supported by the Department of Energy, contract DE-AC03-76SF00098.

References

1. *BaBar* Collaboration, *BaBar Technical Design Report*, SLAC-R-95-457, March 1995.
2. G. Booch, *Object Oriented Analysis and Design with Applications*, Benjamin/ Cummings, 1994.
3. D. R. Quarrie, *A Framework for Distributed Mixed Language Scientific Applications*, to be presented at this Conference.