The BaBar Computing Model

Neil Geddes,

Rutherford Appleton Laboratory, Chilton, Didcot, England, OX11 0QX

The BaBar experiment will begin data taking in 1999. We will record and fully reconstruct up to 10^9 event decays of the upsilon(4s) resonance per year. We present our current status and plans for acquiring, processing and analysing this data.

Key words: babar; computing model; software development; OO

1 Overview

The BaBar [1] experiment, currently being constructed at SLAC, will study CP violation in the B meson system at the Stanford Linear Accelerator Centers PEP-II accelerator. Due to the rarity of the CP violating decays these studies require large event samples and to acquire these the PEP-II accelerator and BaBar detector will run continuously for over half of each year ("factory" operation). We expect to record 10^9 events per year, totalling approximately 25 TB of data. A large fraction of this data will require detailed analysis to isolate the small signal samples.

The BaBar collaboration is geographically widespread and we have developed a computing model to accommodate this far flung expertise in a workable and cost effective way, both during the construction and running of the experiment. We already make extensive use of wide area networks for communication and software development. We also define and support a standard BaBar development environment which allows full integration of remote collaborators.

The data will be recorded at SLAC, giving SLAC a unique position. Demands for detector monitoring and maintainance, physics analysis by collaborators resident at SLAC, and simplifications in the data handling will mean that the bulk data processing will be performed online at SLAC. There also exist within the collaboration several sites with current or planned computing facilities (data storage, computing capacity and support) similar to those at SLAC. These "Regional Centres" will serve as code development and analysis centres, providing convenient data access to remote collaborators. These centres should also provide a significant fraction of BaBars Monte Carlo needs and could also be used for data processing, if required. It is envisaged that data analysis will be based around the data store at SLAC or a regional centre, whichever is most convenient. Other institutions *may* keep local copies of small data samples.

2 Software Development

As detailed elsewhere in these proceedings [2] BaBar has adopted an object-oriented approach for its software development, with C++ as the principal programming language. Both of these were relatively new to widespread use in HEP and this has produced practical difficulties in retraining and acquiring experience. To support the dual needs of software development and training, an iterative design and implementation process has been adopted. Developers are encouraged to strike a balance between "learning by doing" and "doing while learning". This has been supported by home–grown and commercial courses on C++ and object–oriented analysis and design. We now have a small but growing number $(10 - 20)$ of competent principal developers. C++ experience is spreading further through the collaboration as we get closer to a functionally complete reconstruction program enabling "physics analysis" type tasks to be undertaken. We are also developing a unified fast/detailed simulation of the BaBar detector based on GEANT4, to replace our current GEANT3 based programs and much of the detailed detector simulation is already written in C++.

To support our software development we have created a code management and release system [3] which allows integration of a distributed developer community. Source code management is provided by CVS. The main BaBar CVS repository is based at SLAC but is made accessible to collaborators worldwide via AFS. BaBar software is organised into "packages". Each package contains groups of classes/functions performing a well defined task. Each package is maintained as a separate CVS module and is managed by a package coordinator. The package coordinator is responsible for the development of the package, for testing of the code, and for tagging new self-consistent and tested versions as and when appropriate. Access to CVS modules can be controlled via AFS ACLs if required and package coordinators are notified automatically when modifications are made to a package.

A full "release" of the BaBar software is built at least every two weeks, based on the then current versions of all packages. A release is a snapshot of a consistent set of package versions, and includes compiled libraries and binaries for each of the supported BaBar platforms (currently, HP-UX10, AIX4, and OSF1.3). If the build is successful the release will become the "test" release and subsequently, if testing proves successful, the "current" or "production" release. Releases are tagged by a sequential number and logical links are used to reference "latest", "test", "current" and "production" releases. Releases will remain on disk for as long as is required. Creation of a release is increasingly automated, including code checking, compilation and testing. The latter is based on specific tests provided by package coordinators which can be run automatically, producing output to be compared with a supplied reference copy. Errors from all stages of the release build are emailed automatically to package coordinators. Outside release builds, bug reporting and tracking is provided using the GNATS package.

3 Data Acquisition and Reconstruction

The "physics" event rate at BaBar is ~ 30Hz. However, large machine backgrounds are expected, associated with the high luminosity of the PEP-II accelerator. A first level trigger, based on information from the main tracking chambers and the calorimeter, will restrict the read out rate to less than 2kHz, while ensuring high efficiency for events of interest ($> 99\%$ for B events and $> 95\%$ for most other channels). A software filter then uses information from other subdetectors, principally the vertex detector and tracking chambers, to further reject machine backgrounds and reduce the data rate to ~ 100Hz.

The online system is based around VME and UNIX. The data is read out from the front end electronics through a uniform VME interface. The full readout spans some 20 VME crates and the aggregate output rate is expected to be around 50MB/second, corresponding to the 2kHz trigger rate. Data are read out in parallel from the VME crates into a farm of UNIX processors which perform the event filtering. At this stage events can also be tagged to simplify subsequent processing.

The readout from VME to the online farm is accomplished through use of a none blocking network switch. The data transport uses standard tcp/ip protocols to isolate the software from the underlying fabric, which may be 100BT fast ethernet or ATM OC-3. Event building also occurs during this transfer stage, with the relevant event fragments from each readout crate being collected in a single farm CPU.

Events from the Online farm are automatically forwarded to the prompt reconstruction system. The aim is to process all events within a few hours of their being recorded. The principle aims of the prompt reconstruction are

- To efficiently tag events for subsequent analysis or reprocessing
- To provide rapid and effective monitoring of the detector and accelerator performance, based on quantities relevant to the physics goals of the exper-

iment.

- To make efficient use of manpower through effective automation of the data recording and reconstruction process.
- To provide rapid access to physics quantities for the analysis groups.

We currently expect that the reconstruction will be performed on farms of commercial unix processors. A prototype farm has been under development at SLAC for some time. Developments to date do not tie us to a single architecture and we continue to monitor developments in SMPs and commodity hardware (PC's). Output from the prompt reconstruction will be written to the primary data store.

4 Data Model

With the adoption of object-oriented reconstruction and analysis software a solution must be found to managing the input/output of the produced data (objects). An Object Database Management System provides a natural storage model. A full database system is well matched to the problem and also provides tools for management of large data volumes in a distributed heterogeneous environment. An ODBMS can also provide good integration with C++, while not limiting object retrieval to this language.

BaBar have chosen the commercial ODBMS Objectivity/DB for our initial implementation of the data store, following the choice of this same database system for our conditions database. Following initial evaluation studies of Objectivity/DB, we currently have a prototype version of the conditions database under evaluation and a prototype event store under development.

The Event Store database design assumes that several types of data may exist for each event (some of the terminology is borrowed from Atlas) [4]:

SIM (Simulated Truth) Underlying physics event data generated by Monte Carlo simulations (the Monte Carlo "truth").
RAW (Raw Data) The raw data output from the experiment (or Monte Carlo simulation). This data can not be regenerated. This could already be output from the reconstruction programs (25kB per event)
REC (Reconstructed data) Output from the reconstruction programs, can be regenerated from the **raw** data (100 kB per event).
ESD (Event Summary Data) A compressed form of the reconstructed data, corresponding approximately to a traditional DST (10KB per event).
AOD (Analysis Object Data) End user physicist data, based on 4-vectors, particle types, etc. (1KB per event).
TAG Highly compressed summary data for classifying events and allowing

single selection queries to be performed (< 100 bytes per event).

HDR (Event Header) The event header containing references to the other information in the event, designed to be as compact as possible (< 64 bytes per event).

Objects in the data store are accessed through “collections” which contain references to the events within the event store. New collections, e.g. corresponding to some specific event signature, may be created as required and can be accessed a simple name which is stored in an event “dictionary”.

Data present in the database may be replicated to improve access efficiency, or for export. We anticipate exporting complete or partial copies of the database to several Regional Centres supported by the collaboration. Objectivity/DB, already provides tools for integrating a distributed database into a single logical entity. We will investigate these features in the future in relation to their usability with currently available networking, however, we do not assume that this will be required.

It is unlikely that we will be able to store our complete event database (100Tbytes) on disk. It is, therefore, essential that we effectively integrate the database with the SLAC mass store. This is based upon the current SLAC STK tape silos, upgraded to use StorageTek helical scan tape technologies. This will provide tape storage for up to 600TB. A large fraction, if not all of the most compressed and most frequently accessed data will remain on disk. Less frequently accessed, more detailed, data will be staged from the tape store as required.

5 Analysis Interface

To use the full power of the C++ objects output from the reconstruction. A high level toolkit is being developed for physics analysis. The Beta package from BaBar [5] provides a ”physicist-compatible” level of abstraction for using reconstructed information to do four-vector analysis, vertexing, particle identification, etc. Although extensible by sophisticated users, it is primarily intended to provide an accessible analysis environment for those new to C++. It is designed to be very easy to use, with simple metaphors for common tasks, and without requiring the user to master the archania of the C++ programming language. It is intended to be complete, in the sense that it permits the analyst to examine anything available within the reconstruction processing and output. Beta provides the analyst with a common interface to fast or detailed Monte Carlo or real data, regardless of how sparsified and compressed, provided that the required information is available. To do this, Beta defines several basic concepts:

Candidate A candidate is the Beta representation of what is known or assumed about some particle that may have existed in the event. Charged tracks can be candidate pions, neutral clusters can be candidate photons. The output of vertexing two candidates is another candidate, representing the "particle" that might have decayed to those two tracks. All candidates, from different sources and with different implementations, provide the same, complete interface, so they can be used interchangeably.

Operator An operator can combine one or more candidates to form new ones. Vertexing, mass constrained fitting, and simple 4–vector addition are all operators. The analyst selects the operator needed at particular stages in the analysis, trading off accuracy and completeness for speed and flexibility.

Finders Finders search for decayed particles (K^0, D^0, π^0 ...)

associators an associator is used to associate candidates with each other (e.g. Monte Carlo truth and reconstructed tracks, charged tracks and clusters etc.).

Beta is being actively developed now and is being used to performs realistic and complex analyses as part of a Mock Data challenge currently being undertaken by BaBar.

6 Regional Centres

There are currently two sites functioning as regional centres in BaBar at RAL and CCIN2P3. It is planned that there will be another site in Italy and possibly one in the US. The centres currently act as foci for software development, simplifying the distribution and maintanence of BaBar software and the coordination of commercial licenses. These sites also provide significant facilities for Monte Carlo generation. The Three European centres currently plan to copy slightly different versions of the data, based upon anticipated local demand together with the available or planned mass storage infrastructure:

CCIN2P3 The full reconstruction output and any raw data that may be required to fully reprocess the events.

RAL The Event Summary Data. Full reprocessing will not be possible from this sample, although refitting of tracks etc. should be feasible.

Italy Analysis Object Data.

Collaborators will perform most analyses on the data sets at convenient regional centres (SLAC for US collaborators), possibly copying output Ntuples or small event samples back to their home institution. When requiring access to more detailed data then analysis must be performed at another centre, e.g. SLAC, where the more detailed data is available. ODBMSs in principle allow for full integration of such a distributed system with the data at all centres forming a single large database. User queries would automatically be routed to where the data reside. We expect that international networking will be at best unpredictable over the next 5 years and, consequently, this full integration is not a requirement in our data model. We anticipate that, initially at least the regional centres will function as autonomous copies of the master database at SLAC. We will investigate the automation/federation of these databases next year (1998).

7 Conclusions

We have developed a computing model which allows widely dispersed developers to collaborate on software development. We have made good progress in adapting to OO programming and are developing an analysis model, based on an ODBMS and a common analysis interface, which can take advantage of flexibility and power of this approach. Perhap most significantly many of our developments to date are already being studied with interest by other colaborations.

References

[1] D. Boutigny *et. al.* "BaBar Technical Design Report", SLAC-R-95-457.

[2] R. Jacobsen, "Applying Object-Oriented Software Engineering at the BaBar Collaboration." paper presented at this conference.
S.F. Schaffner and J.M. LoSecco, "Object-oriented Tracking and Vertexing at BaBar"
paper presented at this conference.

[3] R. Jacobsen, "The BaBar reconstruction system's distributed development/ centralised release system ", paper presented at this conference.

[4] Atlas Collaboration, "ATLAS Computing Technical Proposal", CERN/LHCC 96-43

[5] R. Jacobsen, "Beta: a high level toolkit for BaBar physics analysis", paper presented at this conference.