SLAC–PUB–7991
November, 1998

Event Logging and Distribution for the BABAR Online System [1,2]

S. Dasu

Department of Physics, University of Wisconsin, Madison, WI 53706.

J. Bartelt, T. Glanzman, T. J. Pavel

Stanford Linear Accelerator Center, Stanford, CA 94309.

For the BABAR Computing Group [3]

Abstract. Event logging and distribution needs for the BABAR online system are described. The hardware and software technologies selected to implement these needs are presented. An initial performance report and a technology assessment are provided.

Introduction

The BABAR experiment [1] at the Stanford Linear Accelerator Center is being built to study the particle and anti-particle asymmetries. These small asymmetries due to a phenomenon called the CP violation, are hitherto observed only in the K mesons are also expected to be observable in the B mesons as well. Both validation of the Standard Model explanation for the CP violation phenomenon and searches for beyond the Standard Model explanations require the CP violation data in the B meson system. The observation and study of the CP violation phenomenon in the B meson system requires production of at least 10^9 electron-positron collision events per year at the Υ(4S - 10.58 GeV) resonance. These events are measured by the BABAR experiment with high precision, using its silicon vertex detectors, drift chamber, ring imaging Cherenkov detector, CsI crystal calorimeter and resistive plate chambers. The raw data for typical events from these sub-detectors adds up to about 30-50 kB per event. During the data taking the collected stream of data from the detector is expected to run at about 3-5 MB/s. A robust and efficient computing system is required to receive this data-flow and reconstruct the events before their final storage, with raw and reconstructed data, in an object database. These data, accumulated over years, are analyzed by physicists to study the B meson physics.

---

[1] *Presented at the International Conference on Computing in High-Energy Physics (CHEP98), Chicago, IL, USA, August 30, to September 4, 1998*

[2] This work was supported by the Department of Energy contracts DE–FG02–95ER40896 (University of Wisconsin) and DE–AC03–76SF00515 (SLAC).

[3] We thank our summer students R. White and S. Bonneaud for help in implementing some of the software described here.

The data from the BABAR sub-detectors are collected by custom VME electronics and presented to the data acquisition system using a set of generic read out modules via VME based PowerPC processors. These processors running VxWorks operating system communicate outside the crate using Fast Ethernet interfaces. This data-flow system and its software are described in detail elsewhere in this proceedings [2]. The data from various subsystem processors are assembled into a full event on a farm of Unix computers. These data arriving at a combined 2 kHz rate into the farm are reduced to a 100 Hz flow by online event processing framework [3] that supports level-3 trigger software [4]. These 100 Hz data comprise of the events that need to be processed and archived.

Although each event is independent of the others, the environmental information, e.g., for calibrating the drift time in the chambers, requires events contiguous in time. Therefore, it is important to collect these streams of events from various computers into a single stream to make collections of events spanning a certain time period, e.g., 30 minutes. These event collections are called "ConsBlks" because they are used to extract various constants that are used in full reconstruction of the events. The "prompt reconstruction" of these events requires running large and complex software programs [5]. Prompt reconstruction provides physicist early access to the fully reconstructed events enabling timely publication of results. Although the reconstruction is expected to be prompt, i.e., latency of at most two hours, it is prudent to provide a data buffer to separate these large programs from the data-flow software. The size of these 30 minute data sets is several giga-bytes strongly suggesting disk file buffers. Additionally, this disk file buffer decouples the data taking from the offline storage. In this paper we describe the software, the logging manager, that collects the events from several Unix computers to make these "intermediate" event files (ConsBlk files) and distributes those events to the processor computers for final archival.

Technology Selection

The hardware platform for the BABAR online system is illustrated in Figure 1. The VME computers and the individual farm nodes have Fast Ethernet (100Base-T) connections to the Cisco Catalyst 5500 switch. The Online server computer, where the logging manager runs, supports an array of high performance RAID disks for intermediate file storage, and is connected to the online switch using a Gigabit Ethernet (1000Base-SX) link. A separate farm of Unix computers in the SLAC Computing Services department support the BABAR experiment for running both reconstruction software and final data archival in a HPSS based Objectivity database. The SCS farm is connected to the BABAR online switch using a Gigabit Ethernet (1000Base-LX) link. The choice of computers and IP interconnects are based on careful evaluation of the commercial technology [6] considering both performance and cost issues.

The Unix processes and their IP based communication links involved in orchestrating the online platform to enable the intermediate file storage and data processing before the final storage in Objectivity database are illustrated in Figure 2. The reliable server processes run on the online server computer and command the daemon processes on the rest of the farm to perform specific tasks. The logging manager processes collect and distribute the data using TCP/IP sockets for high performance, whereas the monitor and control data is exchanged using CORBA. When the processes are co-resident in a memory space, e.g., PR daemon and PR framework, shared memory is used to make the best use of the system resources.

The software is built using object oriented design. The development phase involved modeling the objects using Unified Modeling Language. Class diagrams and the use case diagrams were particularly useful for communication among the development team members. Up-to-date UML models also serve as a good documentation for future maintenance. The C++ Standard Template Library provided many useful constructs that would have otherwise involved considerable amount of programming on our part. In order to exploit the multi-thread and socket programming features while maintaining object oriented design we selected the ACE wrapper libraries and TAO CORBA implementation [7]. Use of CORBA allowed us to select Java with IIOP to build the monitoring processes. The cross platform Java technology is useful in allowing easy desktop access.

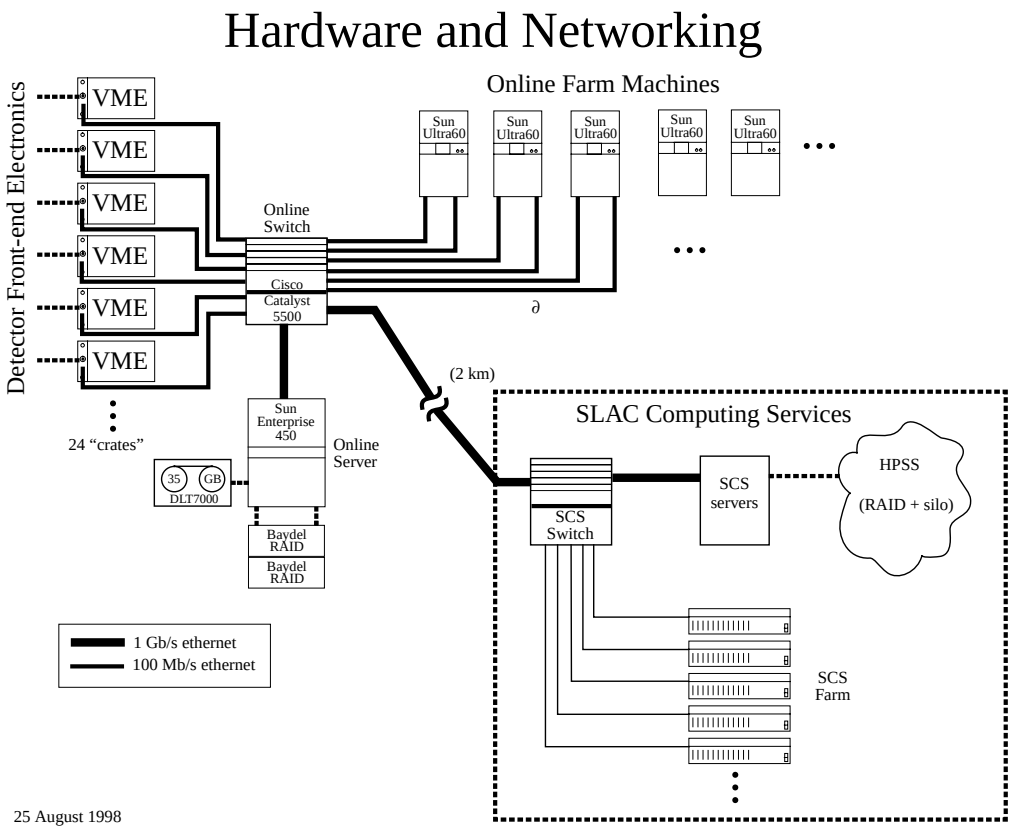

Figure 1: Diagram of the BABAR online computer farm architecture showing the VME PowerPC front-end computers, the online farm of Sun UltraSPARC computers, the Sun Enterprise 450 server with the intermediate store disk array and the SLAC computer center farm connected using Cisco Catalyst 5500 switches.

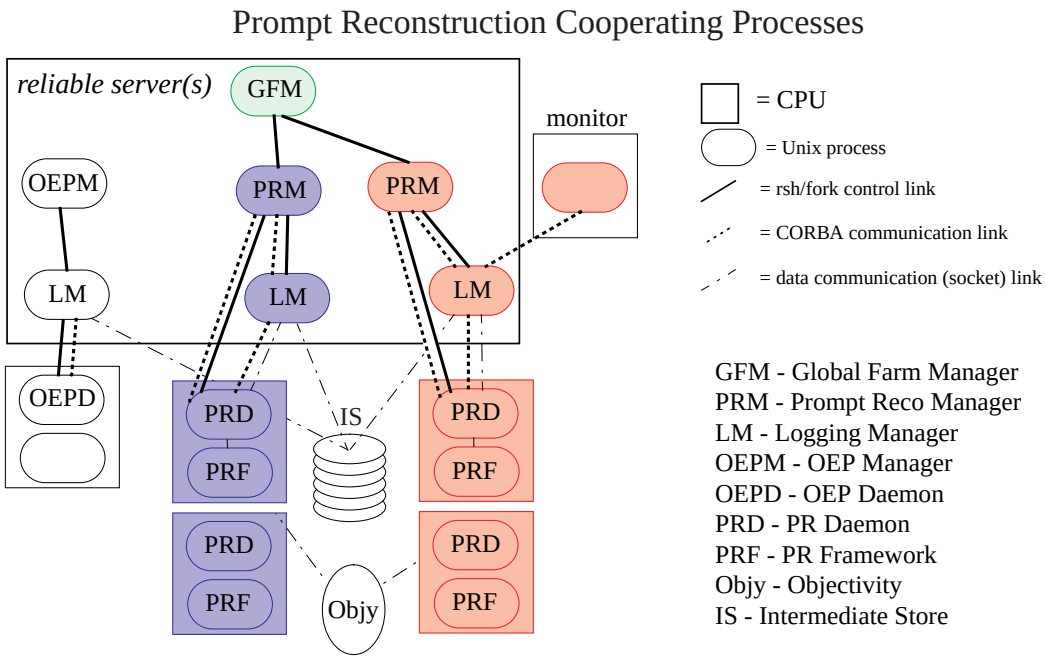

Figure 2: Online event logging and prompt reconstruction cooperating processes and their interaction with storage systems.

The details of the prompt reconstruction [5] are described elsewhere. Here, we report only on the Logging Manager process.

Logging Manager

A simplified class diagram showing the logging manager application is shown in the Figure 3. This multi-threaded program is implemented with one thread that provides CORBA interface for monitor and control, and a separate thread for each input and output client. This program receives events through input objects, maintains a queue of current and pending events and delivers events out through output objects. Three types of input and output classes are available: socket, CORBA and ConsBlk file. The design is such that any number and type of input or output objects can be connected at any time. The setup of the program is accomplished through CORBA calls. The TCP/IP socket or the file setup information is exchanged through this mechanism. The mutex-locked queue status and the input and output status lists are available for monitoring. The status of ConsBlk files is maintained in an online database [8] for tracking reconstruction progress. This persistent database also maintains the information about the input and output clients for disaster recovery.

Event Logging

For event logging, several online event processing daemons will connect to the logging manager and send events through socket interface. These events are written to the ConsBlk files. There is only a small amount of processing involved to handle a selection of special input marker events. The file opening and closing at appropriate time intervals is controlled through these special marker events in the data stream itself.

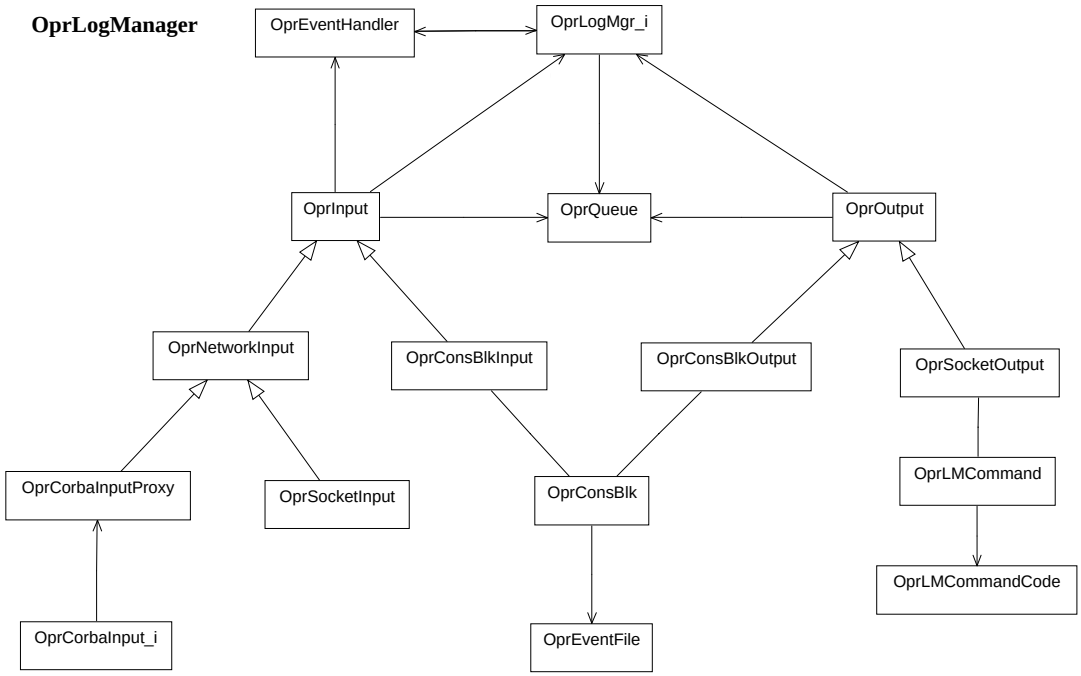

Figure 3: Logging Manager simplified class diagram.

Event Distribution

For prompt reconstruction the events are distributed from a selected ConsBlk file to several prompt reconstruction daemons via socket interface. On this socket link there is bidirectional communication to ensure that every event is processed and safely logged to the object database. The objects needed for the constants calculation are accumulated from event data on several prompt reconstruction computers. These object databases [8] need to be merged to obtain a single database for the ConsBlk time period. Only a small amount of processing is involved to drive the prompt reconstruction framework state machine to allow processing these database accumulations from various daemons and constants calculation. All of this support is handled by special marker events inserted into the data stream by the logging manager.

Performance

Although we have not tested the programs on the final system, we are satisfied with the performance of the network links themselves. We expect that the final system illustrated in Figure 1 will have more than adequate performance. The software overhead placed by the ACE in transporting the events using TCP/IP sockets is also more than satisfactory. We are pleased to report that even TAO CORBA implementation has good performance for transporting event data as simple sequences of unsigned integers.

Technology Assessment

The commodity Fast Ethernet technology suffices for a majority of our online links and is a good choice considering cost issues. The choice of Gigabit Ethernet for our most demanding links is also appropriate. The object oriented paradigm that we adhered to in building our software enabled its implementation in

Use Case	Server	Clients	Data rate (MB/s)
OEP Logging	Sun Enterprise 100Base-T	Sun Sparcs 4 100Base-T 3 10Base-T	10
CORBA Event Logging	Sun Enterprise 100Base-T	Sun Sparcs 4 100Base-T 3 10Base-T	5
Prompt Reco Event Distribution	Sun Enterprise 100Base-T	Sun Sparcs 4 100Base-T 3 10Base-T	8

Table 1: Performance of the logging manager in a test configuration.

a timely fashion with good reliability. Unified Modeling Language descriptions of the software project helps communication between the group members in development and in documentation. The ACE and TAO packages make an excellent freeware choice in building object oriented applications in a client-server environment. We also found that the C++ Standard Template Library to be valuable.

References

1. The BABAR Technical Design Report, SLAC-R-95-457, 1995 http://www.slac.stanford.edu/BFROOT/doc/TDR.
2. The BABAR Data Acquisition System, I. Scott et al., Paper No. 19, this proceedings.
3. Flexible Processing Framework for Online Event Data, G. Dubois-Felsmann, Paper No. 525, this proceedings.
4. Architecture of the BABAR Level-3 Software Trigger, E. D. Frank, Paper No. 117, this proceedings.
5. BABAR Prompt Reconstruction, T. Glanzman, Paper No. 52, this proceedings.
6. Network Performance Testing for the BABAR Event Builder, T. J. Pavel et al., Paper No. 31, this proceedings.
7. An Architectural Overview of the ACE Framework, D. C. Schmidt, http://www.cs.wustl.edu/ schmidt, USENIX login magazine, November, 1998. and The Design of the TAO Real-Time Object Request Broker, D. C. Schmidt et al., Computer Communications, Elsevier Science, Volume 21, No. 4, April, 1998.
8. Databases for Tracking BABAR Online Processes, J. Bartelt, Paper No. 47, this proceedings.