such applications and provided significant improvements in throughput. Careful consideration has been given to scalability in all aspects of the system design. Parallel I/O and execution streams, asynchronous processes, high performance networking and data paths, and data management techniques including intelligent read-ahead strategies are all part of the system design to achieve both peak performance for single jobs as well as excellent aggregate performance for multiple jobs running simultaneously .

The project has developed and /or integrated a number of major components to achieve its performance goals. Use of C++ data structures to represent physics quantities along with a light-weight persistent object manager substantially reduces I/O by focusing this bandwidth on the subset of data needed for event selection. This also enables event data to be quickly and efficiently reloaded from disk to memory for subsequent accesses. Integrated mass storage including high performance robotics and tape drives are implemented to provide good data access performance to multi-terabyte data stores. Lastly, a PAW-like query language, EQL, along with a set of tools, SQS, has been implemented to enable physicists to easily create, manage and execute event selection and query jobs. Tools and procedures for converting between ZEBRA-structured data and the system's native object format have also been developed.

The CAP prototype system currently consists of a 24-node IBM SP2 system with 300 GB of SCSI-attached disk and is connected to a NSL-Unitree managed mass storage system with an IBM 3494 automated tape library. Ultranet is used to provide high speed data paths between several of the SP2 nodes and the Unitree storage servers. Use of the new IBM 3590 tape technology enables the 3494 to hold 30TB of data while providing high aggregate bandwidth when accessing this data. Fermilab's collider experiments, CDF and D0, are beginning to use this system for analysis of datasets in the hundreds of gigabytes class. These early tests are providing feedback to ongoing design improvements that will enable this system to handle the multi-terabyte datasets expected for Fermilab collider experiments running in the next 5 to 10 years.

BR9737146

ABS_65

An Analysis Framework and Data Model Prototype for the BaBar Experiment

F. PORTER (CALTECH)

D. Quarrie (LBL)

The BaBar experiment is a high energy physics experimnet to do physics with e+e- colliding beams in the 10 GeV center-of-mass energy at the PEP-II accelerator at the Stanford Linear Accelerator Center. The principal objectives are to study CP violation and rare processes in decays of B mesons. The experiment is under construction, with first data anticipated in 1999.

The BaBar experiment is expected to accumulate of order 10^9 events per calendar year, with over 10^8 interesting hadronic events. The data must be stored efficiently, but must be easily accessible in order for multiple and frequent physics analyses to be carried out. The analysis framework must be flexible enough to accommodate a variety of analysis modules and multiple input/ouput streams.

The BaBar collaboration has developed a prototype for the analysis framework and data access, written in C++ using an object-oriented design philosophy. The data access is based on the Farfalla package. The base class is a "node". Various types of node objects can be associated into a "tree" organization to form the record of an event. Because of an existing Fortran code base and expertise, access from Fortran 90 as well as C++ is a requirement, and various implementations have been studied.

The analysis framework is intended to be a system which can accomodate code from a variety of sources in both online and offline environments. The prototype is based on the idea of forming sequences of modules which can be linked together to obtain the desired processing of data streams. There are standard modules, which perform tasks such as input, output and filtering. In addition, the user can create modules for particular analysis needs. A complete execution sequence, called a path, can be active or inactive, or can be terminated prior to completion based on the event data in the filtering process. Multiple paths can be specified.

BR9737147

ABS_63

Some Remarks on Parallel Data Mining Using a Persistent Object Manager

R. GROSSMAN (ILLINOIS)

N. Araujo, D. Hanley, and W. Xu (Illinois), K. Denisenko, M. Fischler, M. Galli, and S. C. Ahn (Fermilab), D. Malon and E. May (Argonne)

High performance data management will be as important as high performance computing by the beginning of the next millennium. In this environment, data mining will take on increasing importance. In this paper, we discuss our