CONF-9509237--1

Fermi National Accelerator Laboratory

FERMILAB-Conf-95/357-E

DØ

Object Based Data Access at the DØ Experiment

S. Fuess
For the DØ Collaboration

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

RECEIVED
FEB 06 1996
OSTI

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

November 1995

Proceedings of *Computing in High Energy Physics 1995, (CHEP '95)*, Rio de Janeiro, Brazil, September 18-22, 1995.

MASTER

Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CHO3000 with the United States Department of Energy

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DT

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

OBJECT BASED DATA ACCESS AT THE DØ EXPERIMENT

Stuart Fuess

Fermi National Accelerator Laboratory
Batavia, Illinois 60510, USA

For the DØ Collaboration

The DØ Experiment at Fermilab is currently participating in the FNAL Computing Division's "Computing for Analysis Project" (CAP) to investigate object based data storage and access. Following a short description of the CAP system architecture, the DØ data model is explored. A brief discussion of the method of operation of the CAP system leads into a concluding section.

1 Introduction

1.1 The CAP Project

The Computing for Analysis (CAP) project at Fermilab was initiated for the purpose of exploring new hardware and software techniques for *data-mining* operations. In High Energy Physics analyses there are frequent needs to scan through large data sets in order to classify and select certain types of events.

There are two principle aspects to the CAP project: the hardware architecture and the software tools. The system is centered on a 24-node IBM SP2 processor with a high-speed, low-latency switch interconnect. Approximately 200 Gbytes of disk are mounted, supported by a IBM 3494 robotic tape library providing the bulk of the mass storage repository.

The persistent object manager on the system is ptool, a product developed by Robert Grossman of the University of Illinois at Chicago [1,2,3], and then adapted as the persistent manager by the PASS project [4]. CAP uses a slight modification of ptool to adapt to the specific needs of the SP2 architecture. The CAP software system also includes a hierarchical storage manager, supplied by the NSL-Unitree package.

1.2 DØ Interest in CAP

In Run 2 of the Fermilab collider program, expected to begin in 1999, DØ expects to accumulate at least 100 Tbytes of raw data *per year*. The event reconstruction process will produce an equivalent amount of data. With such huge data sets, efficient and timely access becomes very problematical. A potential solution to the data access problem is presented by CAP, with a combination of hardware providing high data bandwidths and software tools enabling efficient access to structures within an event record.

DØ is currently in the process of redefining its computing paradigm for Run 2 efforts. The areas of data persistency and object oriented data access are being

explored. The ptool product may be able to satisfy some of the DØ software requirements, or may lead to the development of an even more powerful methodology.

Of equal interest to DØ is the examination of the optimal architectural configuration for a central analysis and data serving engine. As disk capacity is not expected to be sufficient for the enormous data sets, a robotic tape store with controlling hierarchical storage is required. Access to these storage devices is also crucial, and the CAP system demonstrates the features of a set of processors with a high bandwidth network interconnection.

As a test data set, DØ imported approximately 260 Gbytes of data from the 1994/95 data taking period. This set contained high p_T triggers in ZEBRA format, after reconstruction, with summary and working banks (but not raw data banks) included.

2 The DØ / CAP Data Model

For the purposes of CAP, DØ chose a data model which would closely parallel the existing ZEBRA data structure. This allowed an easy mapping between ZEBRA banks and objects managed by ptool. This mapping was used when importing data into the ptool store and also when exporting selected events back into ZEBRA. The latter step was important for the validity check of the complete cyclical process.

The first step in defining the data model was to classify the objects created from one (or more) ZEBRA banks as either *cardinal* or *auxilliary*. Cardinal objects, expected to be disk resident, contain information deemed necessary for event classification. Auxilliary objects, generally tape resident, contain the more basic banks used as input to reconstruction algorithms. Since auxilliary objects were intended to be accessed only on final rebuilding of the ZEBRA event, they were collected in a single store for efficient access.

Example cardinal objects are event headers, trigger records, and summary objects. The auxilliary class might include detector hits and tracks.

A more detailed view of the data objects and their linkages is given in Fig. 1. This figure illustrates the important position of the event handler object, which contains links to the headers of the event component objects. Each event component structure is allowed to be a variable length linked list of variable length objects. Event and component headers of successive events are also linked, allowing a complete scan to be made accessing only the object of interest. Finally, the event header contains a link to the auxilliary data object. Like objects are collected in a store referenced by the object name.

To implement the data model, a set of FORTRAN and C++ routines was automatically generated. The starting points of the automatic process are *bank definition files*, the contents of which comprise a data definition language. A file exists for each object in the model, and gives its ZEBRA bank equivalent(s), reference links to other objects, and object member data. A suite of programs then takes the bank definition files and generates code for ZEBRA and object translation, access, and manipulation. The bank definition files are an evolutionary step beyond the current text documentation files produced for each DØ bank.

Links in the model are implemented using the ptool persistent pointer, a 64-bit

entity encoding information on the store, folio, segment, and offset of the persistent object member data. Each object is associated with a store, which may be composed of many folios (files). Each folio is split into fixed size segments, chosen to optimize I/O operations. Within a segment is the particular data of interest.

3 Query Operation

A mechanism, outside the scope of this paper, exists for taking a script-like query of the data set and generating C++ code, compiling, and linking an executable to be run on the CAP system. The execution of the query involves successive dereferencing of the persistent pointer, which then fetches the object data from the persistent store. In the Fermilab ptool implementation, the data store is distributed over multiple I/O nodes and disks. A segment request on the execution node requires the appropriate I/O node to deliver the data. The high performance of the CAP system arises from ptool's efficient memory access, the low latency high bandwidth SP2 switch, and effective distributed disk striping and cacheing.

A query may produce several possible outputs. Most basically, a list of persisent pointers to the headers of selected events can be produced. Using this list, the ZEBRA structures for the selected events can be rebuilt. The user can selectively include specific ZEBRA banks in the output event; most efficiently only the cardinal banks are included, but at the cost of making tape accesses the entire event can be built. Another option allows the user to omit the rebuilding of any ZEBRA structure but instead fill an ntuple with selected information. The implementation of the ntuple scheme allows a dynamic selection of event attributes to be included.

4 Comments and Future Activities

Beyond the creation of the data model and the demonstration that the query process works, the DØ operational experience on CAP is limited. Our future plans include study of the system operation with various data access patterns. A modification of the data model is under way which will allow a greater number of cardinal objects, relegating some to storage on tape with the option of making them programmatically available for use in queries. The auxilliary store is replaced by a store containing the complete event, since auxilliary objects generally represent the bulk of an event.

One important factor in a data model is its ability to evolve. A limitation of the persistent pointer is that any link to a store becomes obsolete if the store is restructured. A critical issue in data management is the manipulation of *hot* events, those which have the most analysis interest. For these events we wish to retain disk storage for all objects within, whereas for less interesting events most objects could be migrated to tape. There will likely be active restructuring of object stores; we'd like to avoid changing the complete data store to update persistent pointer links in such operations. One possible solution under investigation is a combined use of persistent pointers and identifiers resolved by database lookup. The latter could be used to flexibly represent links between objects.

The DØ experiment sees the hardware and software advances represented in CAP as keys to future analysis computing. We will continue to take active interest

and contribute effort to the advances in this area.

Acknowledgments

The work of Seung-chan Ahn and Hailin Li from DØ was essential in the success of this project, as were the talents of the members of the Fermilab Computing Division High Performance and Parallel Computing Department.

References

1. R. L. Grossman and X. Qin, "PTool: a scalable persistent object manager," *Proceedings of SIGMOD 94*, ACM, 1994, page 510.
2. R. L. Grossman, N. Araujo, X. Qin, and W. Xu, "Managing physical folios of objects between nodes," *Persistent Object Systems (Proceedings of the Sixth International Workshop on Persistent Object Systems)*, M. P. Atkinson, V. Benzaken and D. Maier, editors, Springer-Verlag and British Computer Society, 1995.
3. N. Arajuo, R. Grossman, D. Hanley, and W. Xu, K. Denisenko, M. Fischler, and M. Galli, D. Malon and E. May, "Some Remarks on Parallel Data Mining Using a Persistent Object Manager, Proceedings of Computing in High Energy Physics 1995 (CHEP 95), to appear.
4. C.T.Day et al., The PASS Project Architectural Model, Proceedings of Computing in High Energy Physics Conference 1994 (CHEP 94), LBL-35822, April 1994.

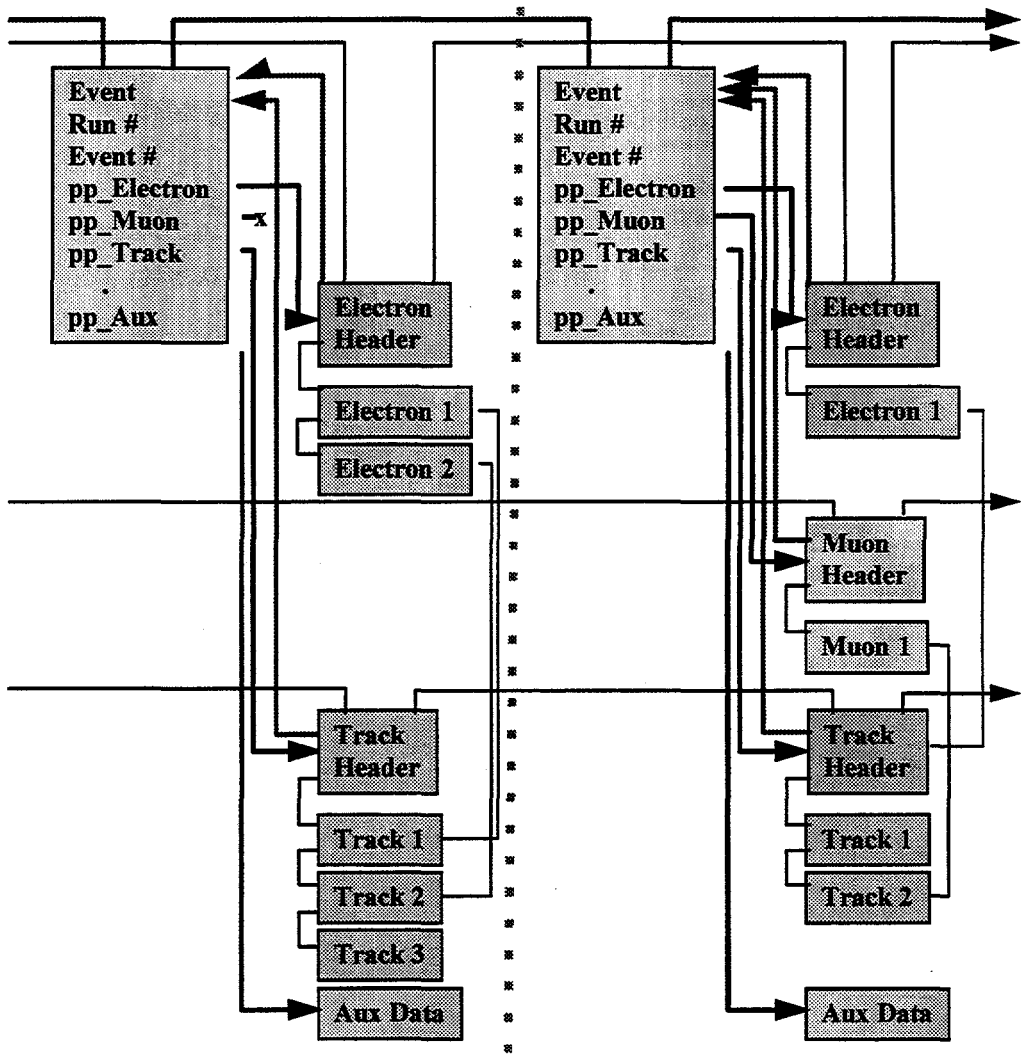

Figure 1: D0 / CAP Data Model