G G—

ANL-HEP-CP-94-29

SW 3430 CS

A Multi-level Object Store and its Application to HEP Data Analysis *

Edward May, David Lifka and David Malon
High Energy Physics Division
Argonne National Laboratory
and
R.L. Grossman, X. Qin, D. Valsamis and W. Xu
Laboratory for Advanced Computing
University of Illinois at Chicago

Abstract

We present a design and demonstration of a scientific data manager consisting of a low overhead, high performance object store interfaced to a hierarchical storage system. This was done with the framework of the Mark1 testbeds of the PASS project.

INTRODUCTION

The work described in this report is part of the PASS project. The motivation and a status report for the PASS project are given in a paper by Quarrie [1]. The basic notion of the present work is to use an Object Oriented (OO) approach to the modeling, storage, and accessing of HEP data. Thus the user should only be concerned with representation of HEP data in-memory (as C++ classes), while the storage and access of the HEP data structures are handled transparently in a multilevel cache based hierarchical storage environment. We use the PTool system [2], which provides an efficient and flexible method of making C++ classes persistent. It is, in effect, an object manager which manages physical collections of objects between memory and disk.

We have extended this model with separate managers to manage the physical collections between disk and tertiary storage devices. This follows the basic design described by the IEEE Mass Storage Model [3].

PTool: Design and Architecture

This section is based in part upon reference [4]. The physical design of PTool is based upon three concepts: objects, folios, and stores. A *folio* is a contiguous range of virtual memory that is managed by PTool. A folio may contain one or more objects or a piece of a large object. A *store* is a physical collection of folios. A folio is implemented as a UNIX file.

The PTool Management System consists of several software tools which the user sees through well defined APIs. The most important are the Persistent Object Manager which creates, accesses, and updates persistent objects; and the Persistent Folio Manager which manages folios between virtual memory and local disk, and between local disk and networked or

*This work is supported by the DOE OSC-HPCC and HEP programs.

hierarchical storage.

The user of PTool works in a C++ programming environment. The user may declare any C++ class (the objects) as persistent in a full 64bit addressable virtual memory space. The user accesses the objects via a "overloaded" pointer. When this pointer is dereferenced during program execution, the PTool system determines if the object is in the local system real memory. If it is not PTool consults the Persistent Folio Server to determine the location of the object within the multilevel cache storage system, then moving the object through the hierarchy by the fastest means available. For local or NFS-mounted disks this can be memory-mapped I/O, for network attached RAID or tape robots this could be lightweight message-passing protocols to remote servers, etc.

The Mark1 Evaluation Tests

The Mark 1 evaluation tests are an extension of the Mark 0 tests reported at the CHEP'92 meeting [5] designed to evaluate the use of various database like technologies for HEP data analysis. A set of test suites using typical HEP data were constructed for use in a hierarchical storage environment. Testbeds at the scale of 10 GB have been developed as an initial point to study the scaling of the performance of database technology over the range of 10 GB to 100GB with the later extension to 1TB. The plans included the testing and evaluation of a commercial relational database, a commercial object oriented database, the PTool object manager, and a typical HEP Fortran-based access method.

The ANL testbed was used for both development work with the Sun Fortran and C++ compilers; and for initial tests. It consisted of a SparcStation 2 (4/75) with local SCSI attached disks (1 GB for data storage) and a 5 GB 8mm tape drive. A SGI 4D/35 server was available via 10 Mb/s ethernet for data storage and access via FTP and NFS remote file system mounts.

The SSCL testbed consisted of a Sun 670MP server with 10 GB of SCSI-2 attached disk and IPI3 attached Ampex DD2 tape drive capable of using the 25GB cartridges.

We obtained a CDF 1989 J/Psi DST data sample in YBOS format stored on 8mm tapes. It contained muon, electron, photon, jet, central drift chamber tracks, and digitized hit information. The YBOS Fortran-based access library was obtained and compiled on the Sun architecture.

Data Model

An object-oriented data model was constructed for the data using the Rumbaugh design notation[6] and an X11-GUI based design tool (OMTool from GE Research, Inc). This tool can output the data model in either SQL or C++ header files, which can be used as the schema for relational or (C++ based) object-oriented databases. In all three cases, the schema produced required some hand editing to fit the specific requirements of the target database.

Data Sample

A group of codes (in Fortran, C and C++) were written to read the CDF YBOS data, and to reorder and populate the test evaluation databases according to the schema obtained for the data model. Due to the cancellation of the SSC project, the building and testing of the commercial relational and OO databases have not been completed by the time of this report. The remainder of this paper will thus describe experience and test results for the YBOS and PTool-based access methods. Three data set sizes were prepared: 805 MB, 1.8 GB, and 6.5 GB in the case of the PTool-based stores. These sizes were set to conveniently fit on local disks, 8mm tapes, and the DD2 tape cartridge.

Test Results

We comment here only briefly on the test results; a full report[4] is in preparation for publication. A comparison of the YBOS and PTool access for typical physics-based queries shows a speedup of approximately four in the wall clock elapsed time. From the technical queries, we note that a study of the effect of accessing objects of increasing complexity (higher multiplicity of instances) shows an elapsed time dependence which is approximately a power law on the complexity with exponent of 1/3. The access to PTool stores which are remotely NFS mounted (via ethernet) are only a few per-cent different from locally mounted disks. Access times to folios stored in a simple hierarchy based upon 8mm or DD2 tape drives, are only limited by the speed of large block transfers from the media to a local disk. These and other results for the OO approach and its implementation in the persistence model provided by PTool within a caching hierarchical storage environment are very encouraging.

Current Activities – Future Plans

The PTool code system has been moved to the GNU g++ compiler for more portability. A cooperative R&D program between ANL and IBM has recently resulted in the siting at Argonne of an IBM 128 node SP1 computer and advanced high capacity I/O system for file systems. The I/O system includes a hierarchical storage system based on a HIPPI connected 220 GB RAID, a 6 TB three read-head tape robot for DD2 tape cartridges, and an NSL Unitree hierarchical storage manager. We have extended the PTool code system to work in this environment and are currently preparing the test suites described above for a detailed evaluation.

During the same time frame a similar R&D project has sited a 24 node SP1 at Fermilab as part of the Fermilab computing division CAP project. Their strategy for high performance I/O and file systems is different and complementary to that of the ANL facility. It is based on multiple directly node-attached SCSI disks and the use of the IBM Research Vesta Parallel File System. We have begun a series of cooperative implementations and tests with the CAP and D0 team that is exploring the use of the SP1 for D0 DST analysis.

We intend to extend the tests of the PTool-based access methods to larger size data stores at a scale typically of 100 GB on the RAID disk and 1 TB in the tape robot at ANL. In addition, the use of NSL Unitree (and later the HPSS) hierarchical storage manager will be explored. Some of the important issues to be studied are: the effect of varying the caching size to match the physical devices in a more optimal manner; the use of different clustering models of which objects (or sets of objects) to store together in a physical storage volume; techniques for reclustering existing stores to better match typical HEP query strategies.

ACKNOWLEDGEMENTS

The authors thank A. Gauthier, U. Nixdorf, J. Marstellar of the Superconducting Super Collider Laboratory for their help with the data model design, manipulation of the CDF data, and use of the SSCL testbed.

REFERENCES

1. D. Quarrie, "The PASS project, A Status Report" CHEP 94, this publication.
2. R.L. Grossman and X. Qin, "PTool: a low overhead, scalable object manager," Proceedings of Sigmod 94, to appear.
3. R.A. Coyne and H. Hulen, "An Introduction to the Mass Storage Reference Model, Version 5", Proceedings: 12th IEEE Symposium on Mass Storage Systems, S. Colman, ed., IEEE Computer Society Press, 1993.
4. R.L. Grossman X. Qin, Valsamis, W. Xu, D. Lifka, E. May, D. Malon, and L. Price, "The Architecture of a Multi-level Object Store and it Application to the Analysis of High Energy Physics Data" In preparation, May, 1994.
5. C.T. Day, et. al., "Database Computing in HEP—Progress Report", Proceedings of the International Conference on Computing in High Energy Physics '92. CERN Report CERN-92-07, 1992.
6. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, "Object-Oriented Modeling and Design", Prentice Hall, 1991.