Optimizing Mass Storage Organization and Access for Multi-Dimensional Scientific Data

Robert Drach, Susan W. Hyer,
Steven Louis, Gerald Potter, George Richmond
Lawrence Livermore National Laboratory
Livermore, California

Arie Shoshani, Doron Rotem
Lawrence Berkeley Laboratory
Berkeley, California

Arie Segev, Sridhar Seshadri
University of California, Berkeley
Berkeley, California

Hanan Samet, Pedja Bogdanovich
University of Maryland
College Park, Maryland

Abstract

A critical issue for scientific investigators is ready access to the large volume of data generated by large scale supercomputer simulations and physical experiments. This paper describes the current status of a collaborative effort which focuses on managing data produced by climate modeling applications. The project is aimed at significantly improving the accessibility and ease of use of large scientific databases, in the context of a hierarchical mass storage system.

Introduction

Present day supercomputer simulations, and automated collection of observations by monitoring devices and satellites, produce very large data sets at increasingly high rates. These data sets are overwhelming conventional methods of storage and access, leading to unreasonably long delays in data analysis. For such applications, processor speed is no longer an issue. Instead, the management of thousands of gigabytes of data is the major bottleneck.

Interactive analysis and visualization applications frequently require rapid access to relatively small subsets of modeling data. Moreover, it is most natural to characterize such accesses in terms of the basic data abstractions involved, such as multi-dimensional data arrays, rather than in terms of files or families of files.

In the context of hierarchical mass storage systems, there is very little support for such applications. Current commercial database management systems provide inadequate support [8]. The primary reasons are:

- Current DBMSs do not support tertiary storage. They only support secondary storage (i.e., disks), not sufficient for the massive data generated and analyzed in scientific applications.
- Current DBMSs do not adequately support data abstractions such as spatio-temporal data arrays. The indexing methods they provide are inappropriate for this type of data, making proximity queries, and operations based on spatial and temporal ranges, very expensive.

To support the kinds of indexing required, many scientific applications use specialized data formats, such as DRS [3] and netCDF [6]. However, these libraries are strictly file-oriented, and do not provide special support for tertiary storage.

A recent NSF Workshop on Scientific Database Management [4] concluded that requirements for scientific applications are not likely to be adequately addressed by commercial vendors, and that federal initiatives, such as the High Performance Computing and Communications Program, need to respond to the problems facing scientists with respect to scientific data management.

To address these problems, we are developing data-partitioning techniques based on analysis of data-access patterns and storage device characteristics, enhancements to current storage server protocols to permit control over physical placement of data on storage devices, use of customized data-compression methods, and data re-assembly techniques for assembling the desired subset. We are also developing models for storing the metadata associated with such data sets and their partitions in a commercial database management system.

The goal of this project is to develop data-management software tools which provide a natural way for modelers to access their data, and to achieve up to an order of magnitude improvement in the response time of the data

1051-9173/93 $3.00 ©1993 IEEE

access, in the context of commercially-available mass storage systems.

Our focus is on developing efficient storage and retrieval of climate modeling data generated by the Program for Climate Model Diagnosis and Intercomparison (PCMDI). PCMDI was established at Lawrence Livermore National Laboratory to mount a sustained program of analysis and experimentation with climate models, in cooperation with the international climate modeling community [5]. To date, PCMDI has generated over one terabyte of data, mainly consisting of very large, spatio-temporal, multi-dimensional data arrays.

The developmental and operational site for our work is the National Storage Laboratory, an industry-led collaborative project [2] housed in the National Energy Research Supercomputer Center (NERSC) at LLNL. Many aspects of our work complement the goals of the National Storage Laboratory. We will use the NSL UniTree storage system for the work described in this paper.

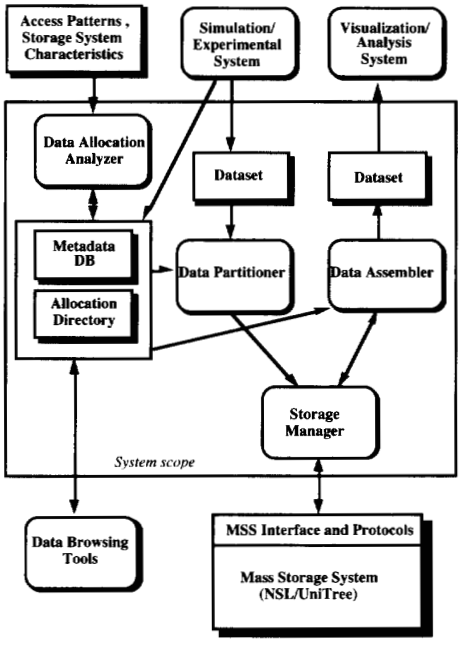

Figure 1. System Overview

Architecture

Figure 1 illustrates the system under development. The inputs to the system are:

- A data set, in the form of one or a family of files. The files are in a self-describing format, which supports multidimensional arrays, such as DRS or netCDF; and
- The expected access patterns for the data set.

The data allocation analyzer uses this information, together with knowledge of the mass storage system, to generate a partitioning specification of the input data. The partitioning module then breaks the data sets into *clusters*. The storage manager controls placement of the clusters on the mass storage system. On output, the data assembler consults the allocation directory to determine which clusters are needed to satisfy the user's request. Then the storage manager determines the location of the data clusters and feeds data to an assembly module, which outputs the data in its requested final form.

The following sections describe our work in more detail.

Physical Database Design

Efficient physical database design on the mass storage system is highly critical to achieve fast response times. A key element of this project is the design of algorithms for partitioning incoming data sets based on expected access patterns, and for allocating the partitioned data sets to tertiary storage such that expected response times are minimized. These algorithms will be used to implement the data-allocation analyzer, data-partitioner, and data-assembly modules shown in Figure 1. In particular, we are:

1. Designing a scheme for allocation of files to tape such that the expected response time for accessing a file is minimized;
2. Developing a file caching strategy so that files which are accessed more often have a high probability of being resident on the disk system; and
3. Designing efficient algorithms for initial loading of the data to the mass storage system based on the generated allocation scheme.

Task 1 is highly sensitive to the performance parameters of the hardware, such as the robot mechanism, as well as to the access pattern statistics. We are building a mathematical model which takes these parameters into account. To better predict performance in a realistic environment, we are also conducting experiments to characterize performance on the Livermore Computer Center and NERSC mass storage systems.

We have modeled robotic libraries as queuing systems and have obtained explicit performance results [7]. The physical model corresponds to a mass storage system where the data is stored on cassette and is retrieved by robots using one or two tape pickers. We have obtained theoretical results about the effect of file splitting on cassettes (file splitting refers to the case where the data needed by a read request is stored on two or more

cassettes) and optimal configuration and control of the robots.

We examined three models. In the first model, we assumed the existence of a single robot and single tape picker. The main results are explicit formulae for the delay obtained if the query arrivals are modeled as a Poisson process. Tight bounds are given when the variance in the arrival process is small. Under the assumptions of this model, file splitting increases the delay substantially. The effect appears to be quadratic in the cases studied.

In the second model, we compared the performance of two robots, each with a single tape picker, to the performance of a single robot with twice the speed. This quantified the benefit of a two robot configuration. The main result is that if the number of cassettes required per query is small, the load on the system is light, and the fast robot is very reliable, then the single faster robot is preferred. Otherwise, the two slower robots are preferred.

The third model treats the case of two robots, each with two tape pickers. We determined the optimal retrieval sequence for deterministic fetch times. For general distributions the question is open, but we conjecture that the prescribed policy is good.

For Task 2, we are investigating the possibility of allowing the system to follow caching "hints" given by the user rather than following a standard caching policy such as least-recently-used.

Once an allocation scheme is adapted, a solution to the third problem will determine in which order files are loaded to the tapes to allow maximum tape parallelism to speed up the loading time.

Metadata Storage

Scientific applications, such as climate modeling, need access to multiple data sets that may be generated or collected over various space and time dimensions, by different projects, using different models, parameters, and granularity levels. Because of the large volume of "raw" scientific data, many data sets are derived from other data sets by statistical summaries (e.g. monthly means), reduction in precision, sampling, and other calculations. Thus, the history of each data set is also important. We refer collectively to all information about data sets as the *metadata*, to distinguish it from the actual data values in the scientific files. To organize metadata, support abstract data types, and store partitioning information, we are implementing a metadata database.

The metadata information is essential in order to find out what data sets are available, to specify subsets of interest, and to benefit from other people's work. Often, such information is not collected in a regular manner or even available in computer-readable form. We have adopted the approach of treating the metadata information as a database in its own right, so that various user interfaces can access that information directly from a database management system.

We plan to use existing database design tools that provide an object-level view of the metadata, and support the metadata on a commercial relational database system. The object-level design uses an Entity-Relationship model. Once the database is designed and populated, the objects can be queried and their content browsed. Here again, we plan to use existing tools for query and browsing.

Typical objects (entities) of interest are *data arrays*, *domains*, *climate models*, etc. We have found that the concept of a *variable* (such as used in netCDF or DRS), which represents "a multidimensional array of values of the same type", is not sufficient for our needs. Since in this project we plan to reorganize and partition data sets into clusters for efficient access from mass storage, a single cluster can have several value types, such as temperature and pressure. Consequently, our data model consists of the following object types:

- A *data object* is a measured or computed value type, such as temperature or wind velocity.
- A *dimension* represents an ordered set of data values, such as longitude or time. Each instance of a dimension has properties such as starting value and granularity.
- A *domain* is a cross product of dimensions. A domain may have properties such as spatial granularity when applicable.
- A *data array* is a multidimensional array which has a domain associated with it, and one or more *data objects*. Only when data objects are associated with data arrays do they assume specific identity with corresponding units of measurement and precision. Note that a data array with a single data object is equivalent to the concept of a variable mentioned above.
- A *data set* is a collection of one or more data arrays.
- A *cluster* is a physical data array, whose content corresponds to a subset of a data array. The subset is expressed in terms of range partitions over the dimensions of the array. Each cluster corresponds to a physical file.
- A *cluster set* is an ordered set of clusters. A property of a cluster set is a specification of how it should be stored in the mass storage system. For example we

may indicate that the clusters in a cluster set should be stored physically adjacent, or on separate devices.

In addition, there are objects that describe *models*, *projects*, *people*, *citations*, etc. These objects are associated in the model using relationships. The model is then translated into a relational database schema, and the metadata loaded directly into the relational data-management system. Using the query and browsing tools mentioned above users can search the metadata using the object concepts described above.

Mass Storage System Interface

To support the kinds of data allocation and access methods discussed above, we are defining a new functional interface to the mass storage system that allows the partitioning and re-assembly modules to influence the behavior of the storage system. The methods employed will remain consistent with current directions of the IEEE Storage System Standards Working Group (IEEE SSSWG).

As discussed above, a cluster corresponds to a physical file or, in IEEE terminology, an individual *bitfile*. The major requirement of the storage system is to provide the partitioning and re-assembly modules with the ability to influence where a cluster set (i.e., a set of bitfiles) is placed on storage devices and physical volume locations.

We propose to investigate and further develop the concept of *bitfile sets* within the mass storage system to allow clients (in our case, the Storage Manager of Figure 1) to influence more of the operational and behavioral characteristics of the storage system servers. The grouping of multiple bitfiles into a particular bitfile set will cause those bitfiles to be stored and treated in ways that meet the expectations of the partitioning and re-assembly modules. In some cases, it may be beneficial for an individual bitfile to be a member of multiple bitfile sets.

The key concept is that the properties of a bitfile set made visible to the Storage Manager correspond well to those characteristics used by the partition and allocation algorithms. If so, the Storage Manager has a reasonable guarantee that the clusters will be placed correctly to provide the required level of performance. We will endeavor to define an appropriate functional interface for operations on bitfile sets to provide a close match to the requirements of the partitioning and re-assembly modules. Depending on the level of success, it may or may not be necessary to encapsulate the bitfile set operations in a logically separate Storage Manager. The partitioning and re-assembly modules might be able to operate as direct clients of the storage system if the developed interface is sufficiently rich.

There is reason to believe that it is possible to merge the metadata storage and the physical database directly into the mass storage system. The Los Alamos National Laboratory's High Performance Data System uses a SYBASE relational database management system to store and access system table information [1]. The Sequoia 2000 Project [9] is another example of an effort to extend database management systems into large archival storage systems. Sequoia 2000 is also investigating the decomposition of large multidimensional arrays into clusters that can be stored together and reconstituted by a database management system. Provided questions of scale and performance are adequately addressed, the joining of databases and mass storage systems may allow a simpler interface between the partitioning and re-assembly modules and the physical storage system.

The concept of bitfile sets presented here is somewhat different than the IEEE SSSWG notion of *bitfile containers*. The bitfile container was developed to convey "containment" properties for multiple entities residing within the system. While the bitfile container would contain bitfiles as well as servers and volumes, the impetus for containers is that it provides the ability to easily "lift the lid" of a container to determine what is inside. If, however, the concept of bitfile containers, as defined by the IEEE SSSWG, can be expanded to provide a more direct mapping to physical storage allocation, then an attempt will be made to use containers. This avoids introducing a parallel, but slightly different, mechanism for group operations on bitfiles.

Reduced Data Sets

To explore ways of further increasing I/O bandwidth, we are developing methods for the production and efficient representation of reduced data sets, which retain the essential features of the original data but are small enough to be stored on a fast-access medium.

Although a typical climate modeling data set contains a large amount of data, the number of variables is generally only at most a few hundred. Each variable value is a function of up to four dimensions, usually longitude, latitude, height, and time. Our goal is for a reduced data set to be two or more orders of magnitude smaller than the original data set. At the same time, we wish to preserve the features of interest to the climate modelers. We are currently using the DRS library (a library of specialized access methods developed at LLNL), to extract data from the PCMDI data sets, and perform a statistical analysis of the variables. The strategy is to look for both spatial and temporal correlations of data values. An appropriate approximation method is chosen on the basis of correlations exhibited by each variable. We are devising hierarchical data-abstraction method(s) which make use of these approximation methods. The data-abstraction

method is a multidimensional indexing data structure which has the following properties:

- Lossy data reduction uses an appropriate approximation method to take advantage of the spatial and temporal correlations.
- The structure supports the efficient execution of operations such as subsetting, data-slicing, and multi-variable queries, and enables quick random and sequential access of data. These operations are required for visualization, browsing, and ad-hoc analyses.

Gridded data sets of global climate data often have an unnecessarily high concentration of points around the poles. We are developing a spatial indexing method which circumvents this problem.

Summary

The ultimate goal of this project is to improve the productivity of scientific researchers who deal with large amounts of spatial and temporal array data in their application. We are developing a system that will increase the speed of access to such data sets, and will support access to the associated metadata, on a hierarchical mass storage system. This project is supported by the High Performance Computing and Communications Program.

References

[1] W. Collins, et al., *Los Alamos HPDS: High Speed Data Transfer*, IEEE Twelfth Symposium on Mass Storage Systems, April 1993.

[2] Coyne, R. A., H. Hulen, and R. W. Watson, "Storage Systems for National Information Assets", *Supercomputing '92 Proceedings*, Minneapolis, MN, November 1992.

[3] R. Drach, R. Mobley, *DRS User's Guide*, September 1991, UCRL-MA-110369.

[4] J. French, A. Jones, J. Pfaltz, *Report of the Invitational NSF Workshop on Scientific Database Management*, Technical Report 90-21, University of Virginia Department of Computer Science, August 1990.

[5] W. Gates, G. Potter, T. Phillips, R. Cess, *An Overview of Ongoing Studies in Climate Model Diagnosis and Intercomparison*, Energy Sciences Supercomputing 1990, UCRL-53916, pp 14-18.

[6] R. Rew, G. Davis, *The Unidata netCDF: Software for Scientific Data Access*, Proceedings of the Sixth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, American Meteorological Society, 1990.

[7] A. Segev, S. Seshadri, D. Rotem, *Performance Evaluation of Mass Storage Systems for Scientific Databases*, LBL Technical Report, 1992.

[8] A. Shoshani, *Properties of Statistical and Scientific Databases*, in Statistical and Scientific Databases, Zbigniew Michalewicz, Ed., Ellis Horwood, February 1991.

[9] M. Stonebraker, *An Overview of the Sequoia 2000 Project*, Proceedings of the 1992 COMPCON Conference, San Francisco, 1992.