EFFICIENT ORGANIZATION AND ACCESS OF MULTI-DIMENSIONAL DATASETS ON TERTIARY STORAGE SYSTEMS

L.T. Chen[1], R. Drach[2], M. Keating[2], S. Louis[2], D. Rotem[1], A. Shoshani[1]

[1]Lawrence Berkeley Laboratory, Berkeley CA 94720

[2]Lawrence Livermore Laboratory, Livermore CA 94550

(Published in Information Systems Journal, April, 1995)

Abstract — This paper addresses the problem of urgently needed data management techniques for efficiently retrieving requested subsets of large datasets from mass storage devices. This problem is especially critical for scientific investigators who need ready access to the large volume of data generated by large-scale supercomputer simulations and physical experiments as well as the automated collection of observations by monitoring devices and satellites. This problem also negates the benefits of fast networks, because the time to access a subset from a large dataset stored on a mass storage system is much greater than the time to transmit that subset over a fast network. This paper focuses on very large spatial and temporal datasets generated by simulation of climate models, but the techniques described here are applicable to any large multidimensional grid data. The main requirement is to efficiently access relevant information contained within much larger datasets for analysis and interactive visualization. Although these problems are now becoming more widely recognized, the problem persists because the access speed of robotic storage devices continues to be the bottleneck. To address this problem, we have developed algorithms for partitioning the original datasets into "clusters" based on analysis of data access patterns and storage device characteristics. Further, we have designed enhancements to current storage server protocols to permit control over physical placement of data on storage devices. We describe in this paper the approach we have taken, the partitioning algorithms, and simulation and experimental results that show 1 to 2 orders of magnitude in access improvements for predicted query types. We further describe the design and implementation of improvements to a specific storage management system, UniTree, which are necessary to support the enhanced protocols. In addition, we describe the development of a partitioning workbench to help scientists select the preferred solutions.

1. INTRODUCTION

Scientists working with spatio-temporal data do not naturally think of their data in terms of files or collections of files, but rather in terms of basic abstractions such as spatial and temporal variables, multidimensional arrays, and images. This work is directed toward providing support for such abstractions within the context of current hierarchical mass storage systems. One of the most critical issues for scientific investigators is the increased volume of data generated by large-scale supercomputer simulations and physical experiments. In addition, the automated collection of observations by monitoring devices and satellites produce vast data at increasingly faster rates. These large datasets have, in some cases, led to unreasonably long delays in data analysis. In these situations, the speed of supercomputers is no longer an issue; instead, it is the ability to quickly select subsets of interest from the large datasets that has become the major bottleneck.

To address this need, we have developed algorithms for partitioning datasets into "clusters" based on anticipated data access patterns and storage device characteristics, as well as enhancements to current storage server protocols to permit control over physical placement of data on storage devices. The access patterns considered in this paper are range specifications in the multidimensional space. The techniques developed can be applied to any multidimensional datasets, although our emphasis and example applications is on spatio-temporal datasets.

In order to have a practical and realistic environment, we choose to focus 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 Na-

PCMDI has generated over one terabyte of data, mainly consisting of very large, spatio-temporal, multidimensional data arrays.

The main requirement is to efficiently access relevant information contained within much larger datasets for analysis and interactive visualization. Although the initial focus of this work is on spatial and temporal data, our results can be applied to other kinds of multidimensional grid datasets.

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. The system integrator for the National Storage Laboratory, IBM Federal Sector Division in Houston, has projects already in place that are investigating improved access interface and data reorganization techniques for atmospheric modelers at NCAR [3]. Many aspects of our work complement the goals of the National Storage Laboratory.

2. BACKGROUND

Large-scale scientific simulations, experiments, and observational projects, generate large multidimensional datasets and then store them temporarily or permanently in an archival mass storage system until it is required to retrieve them for analysis or visualization as shown in Figure 1. For example, a single dataset (usually a collection of time-history output) from a climate model simulation may produce from one to twenty gigabytes of data. Typically, this dataset is stored on up to one hundred magnetic tapes, cartridges, or optical disks (current IBM 3480 tape cartridge technology, used in the storage systems at LLNL, allows 200-250 megabytes per cartridge). These kinds of tertiary devices (i.e., one level below magnetic disk), even if robotically controlled, are relatively slow. Taking into account the time it takes to load, search, read, rewind, and unload a large number of cartridges, it can take many hours to retrieve a subset of interest from a large dataset.

This inefficiency generally requires that the entire set of original data be retrieved and downloaded to a disk cache for the researcher to analyze or interactively visualize a subset of the data. Future hardware technology developments will certainly help the situation. Data transfer rates are likely to increase by as much as an order of magnitude as will tape and cartridge capacities. However, new supercomputers and massively parallel processor technologies will outstrip this capacity by allowing scientists to calculate ever finer resolutions and more time steps, and thus generating much more data. Because most of the data generated by models and experiments will still be required to reside on tertiary devices, and because it will usually be the case that only a subset of that data is of immediate interest, effective management of very large scientific datasets will be an ongoing concern.

A similar situation exists with many scientific application areas. For example, the Earth Observing System (EOS) currently being developed by NASA [3,4], is expected to produce very large datasets (100s of gigabytes each). The total amount of data that will be generated is expected to reach several petabytes, and thus will reside mostly on tertiary storage devices. Such datasets are usually abstracted into so called "browse sets" that are small enough to be stored on disk (using coarser granularity and/or summarization, such as monthly averages). Users typically explore the browse sets at first, and eventually focus on a subset of the dataset they are interested in. We address here this last step of extracting the desired subsets from datasets that are large enough to be typically stored on tape.

It is not realistic to expect commercial database systems to add efficient support for various types of tertiary storage soon. But even if such capabilities existed, we advocate an approach that the mass storage service should be outside the data management system, and that various software systems (including future data management systems) will interface to this service through a standardized protocol. The IEEE is actively pursuing such standard protocols [6] and many commercially available storage system vendors have stated that they will help develop and support this standards effort for a variety of tertiary devices. Another advantage to our approach is that

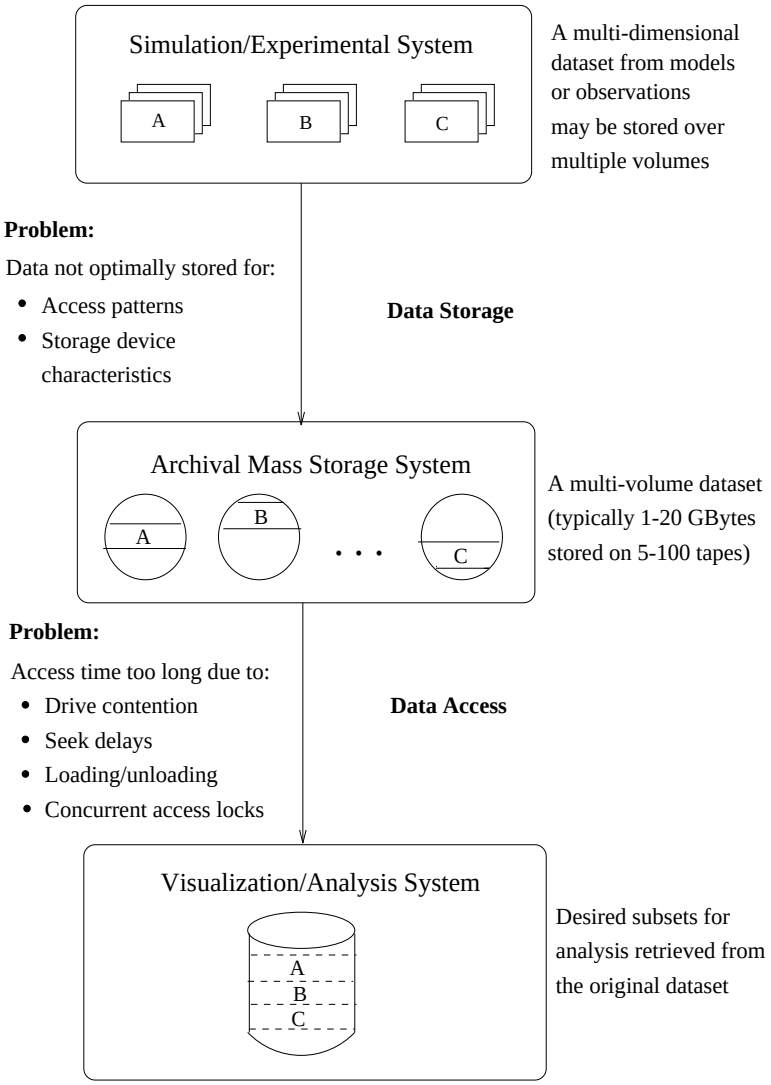

Fig. 1: Current Situation

formats and often prefer to interface to files directly rather than use a data management system.

3. APPROACH

As mentioned above, the main problem we address is the slow access of small subsets from a large dataset in archival storage needed for visualization and analysis. As can be seen in Figure 1, this problem has a storage organization component and an access component. Naturally, the data access depends on the method used for the initial storage of this dataset. Because a dataset is typically stored on tertiary storage systems in the order it is produced and not by the order in which it will be retrieved, a large portion of the dataset needs to be read in order to extract the desired subset. This leads to long delays (30 minutes to several hours is common) depending on the size of the dataset, the speed of the device used, and the usage load on the mass storage system. We show schematically in Figure 1 that the desired subset (which consists of pieces A, B, C which belong to a single dataset) is scattered over multiple volumes of the Mass Storage System.

We address the above problem by developing algorithms and software that facilitate the parti-

partitioning the data such that the corresponding variables are stored as "yearly clusters" in contiguous storage locations will facilitate efficiently reading the desired data. In general, the portions of a dataset that satisfy a query may be scattered over different parts of the dataset, or even on multiple volumes. For example, typical climate simulation programs generate multiple files, each for a period of 5 days for all variables of the dataset. Thus, for a query that requests a single variable (say "precipitation") for a specific month at ground level, the relevant parts of the dataset reside on 6 files (each for a 5 day period). These files may be stored on multiple volumes. Further, only a subset of each file is needed since we are only interested in a single variable and only at ground level. If we collected all the parts relevant to a query and put them into a single file, then we would have the ideal cluster for that query. Of course, the problem is one of striking a balance between the requirements of all queries, and designing clusters that will be as close as possible to the ideal cluster of each query.

The term "partitioning algorithm" is used to indicate that as a result of the algorithm a dataset will be partitioned (or restructured) into many such clusters. The term "cluster" is used to convey the idea that all the data that satisfy a query should ideally reside in a single cluster. The goal is to minimize the amount of storage that has to be read when a subset of the data is needed.

The way that the above techniques interact with the existing software is shown schematically in Figure 2. The same basic system components shown in Figure 1 also exist in Figure 2, along with additional components. The component labeled "Data Allocation and Storage Management" is responsible for determining how to reorganize a dataset into multiple "clusters", and for writing the clusters into the mass storage system in the desired order. The parts of the dataset that go into a single cluster may be originally stored in a single file or in multiple files (as mentioned above, a typical climate modeling dataset is stored in multiple files, each containing 5 days worth of data). The component labeled "Data Assembly and Access Management" is responsible for accessing the clusters, and for assembling the desired subset from clusters (rather than reading the dataset). One consequence of this component is that analysis and visualization programs are handed the desired subset, and no longer need to perform the extraction of the subset from the file. Note that the schematic illustration in the Archival Mass Storage System is intended to show that the desired cluster "ABC" may be stored in contiguous storage space for efficiency as a result of the allocation analysis. The details of the two components are shown in Figures 3 and 4.

On the left of Figure 3, the Data Allocation Analyzer is shown. It accepts specifications of access patterns for analysis and visualization programs, and parameters describing the archival storage device characteristics. This module selects an optimal solution and produces an Allocation Directory that describes how the multidimensional datasets should be partitioned and stored.

The Allocation Directory is used by the File Partitioning Module. This module accepts a multidimensional dataset, and reorganizes it into "clusters" that may be stored in consecutive archival storage allocation spaces by the mass storage system. The resulting clusters are passed on to the Storage Manager Write Process. In order for the Storage Manager Write Process to have control over the physical placement of clusters on the mass storage system, enhancements to the protocol that defines the interface to the archival mass storage system were developed. Unlike most current implementations that do not permit control over the direct physical placement of data on archival storage, the enhanced protocol permits forcing of "clusters" to be placed adjacent to each other so that reading adjacent "clusters" can be handled more efficiently. Accordingly, the software implementing the mass storage system's bitfile server and storage servers, needs to be enhanced as well. More details on the modified protocols are given Section 7.

In Figure 4, we show the details of reading subsets from the mass storage system. Upon request for a data subset, the Storage Manager Read Process uses the Allocation Directory to determine the "clusters" that need to be retrieved. Thus, reading of large files for each subset can be avoided. Here again, the bitfile server and storage server of the mass storage system needs to be extended to support enhanced read protocols. Once the clusters are read from the mass storage system, they are passed on to the Subset Assembly Module. Ideally, the requested data subset resides in a single cluster (especially for queries that have been favored by the partitioning algorithm).

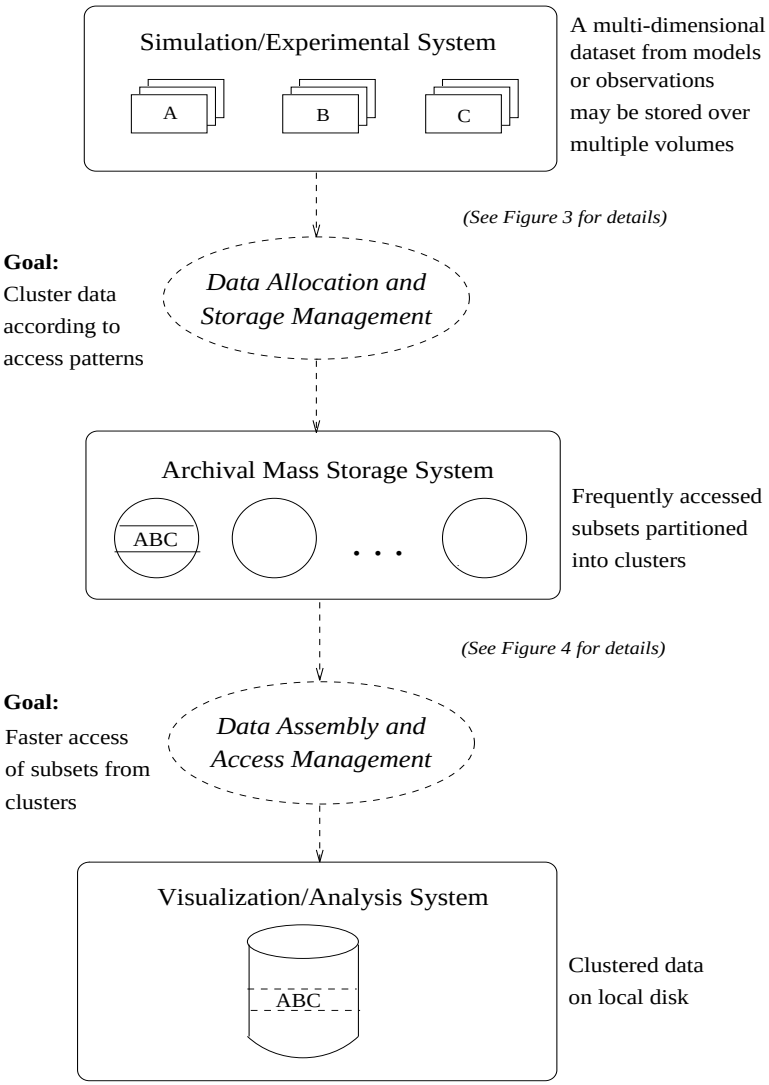

Fig. 2: Areas of Improvements

smaller than the entire dataset. The Subset Assembly Module is responsible for accepting multiple clusters, selecting the appropriate parts from each, assembling the parts selected into a single multidimensional subset, and passing the result to the analysis and visualization programs.

Next, we discuss some details of the partitioning and subset assembly processes, as well as the management of the allocation directory and associated metadata.

3.1. The Partitioning Process

The characterization of access patterns of the analysis and visualization programs is essential for the organization of data to achieve high access efficiency. Of course, there may be conflicting access patterns. Thus, an analysis of the access patterns is needed to determine the optimal partitioning and allocation of clusters on archival storage. The partitioning algorithms use a model of the access patterns as well as a model of the physical device characteristics. The specific techniques used for determining the optimal allocation are given in Section 4.

In an environment of typical mass storage systems we find a multi-level hierarchy consisting

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.