BigSur:

A System For the Management of Earth Science Data

Paul Brown
EECS Department
University of California, Berkeley

Michael Stonebraker
EECS Department
University of California, Berkeley

ABSTRACT

In this paper we present a prototype system for the management of earth science data which is novel in that it takes a DBMS centric view of the the task. Our prototype -- called "BigSur" -- is shown in the context of its use by two geographically distributed scientific groups with demanding data storage and processing requirements. BigSur currently stores 1 Terabyte of data, about one thousandth of the volume EOSDIS must store. We claim that the design principles embodied in BigSur provide sufficient flexibility to achieve the difficult scientific and technical objectives of Mission to Planet Earth.

1. INTRODUCTION

The goal of NASA's Mission to Planet Earth (MTPE) is to better understand the Earth as an integrated system and to study the effects of human activity on it. Because this research requires collaboration among geographically dispersed scientific groups, and history shows that such cooperation is very difficult, NASA will need to construct a data management infrastructure to assist them. Currently, such systems are cumbersome, and the study of global change has suffered because many of its brightest, most innovative practitioners have avoided the analysis of data, and the use of data to verify their models. Further, these systems provide little or no support for the sharing of information among researchers; they have tended to focus on the gathering and storage of data, rather than its dissemination.[DOZ92] As the objective of Earth Observing System's Data and Information System (EOSDIS) is to foster synergistic interactions among users from a variety of disciplines, it's technical objective must be to grant each participant desk-top access to the collective knowledge of their peers.

Data management for MTPE will involve technical problems of striking scale and complexity. The Earth Observing System (EOS) component of MTPE will consist of 20 instruments on 6 satellites to be launched between 1998 and 2002, remaining in orbit for 15 years, continually gathering data about the planet. The system will eventually make available to the global scientific community on a full and open basis an unprecedented 10 Petabytes of raw data and the scientific data products derived from it. It is difficult to predict the size of the EOSDIS user community, because in addition to the 10,000 scientific groups, MTPE's findings will come to the attention of an enormously curious public -- the Internet 20 million. EOSDIS will have a very large, very diverse user community scattered all over the world.

Also, EOSDIS must be more than simply a distributed clearing house for satellite data. It must process each sensor's raw bit stream into some scientifically digestible format. Raw data must be calibrated into appropriate radiometric units based on ancillary readings from surface based instruments. Calibrated data is further processed through several discrete steps, eventually yielding a multi-band raster file. Data may only be rendered for viewing by scientific investigators at this final stage of processing, or they may use it to produce animations of sea surface temperature, or highly refined information like polygonal thematic maps where some biological or geological property is associated with each spatial entity. Over and above massive data storage, EOSDIS will be a system with considerable data processing responsibilities.

The ultimate objective of MTPE is the construction of comprehensive models of the earth system to assist in the management of global climate change. Scientific data for different geographic regions and times will be collated and used to refine current models of the Earth system. As more data is gathered these models will evolve. Once mathematically specified, they will be written into computer simulations generating decades of forecasts based on some set of initial conditions. EOSDIS is to be the system in which these various process models are compared with reality.

In this paper we present the BigSur prototype, a 1:1000 scale model of what we think such a system might look like. BigSur was developed with the cooperation of two scientific groups we introduce in Section 2. These groups are typical of those who will make use of EOSDIS's facilities to further their science. BigSur integrates a satellite feed and the output of a running computer model of the atmosphere into a DBMS centric system, and provides an environment in which these users may specify and deploy a range of scientific data processing functions. BigSur has almost all the features EOSDIS will require to meet it's scientific objectives, and is designed to scale into a system the size of EOSDIS. We conclude Section 2 with a list of several key technical considerations facing implementors of such a system.

Section 3 introduces the BigSur prototype, beginning with a description of its major components, and moving on to demonstrate how they work together to resolve the scientific questions and data processing requirements identified in Section 2.

Finally, we conclude this paper and include a summary of the questions which remain unanswered by this prototype.

2. USER SCENARIOS

In this section we introduce the two scientific groups whose data management needs have guided our development. These groups pursue independent research agendas, one specializes in remote sensing and the other climate modeling, but both would benefit from collaboration.

2.1. Institute for Computational Earth Science Systems (ICESS)

The University of California, Santa Barbara campus is home to the Institute for Computational Earth Science Systems (ICESS). ICESS is an institute providing an environment in which Earth Science and Computer science are strongly coupled, although their research focus is earth science with an emphasis on the processes governing environmental optics of the Earth. ICESS maintains a TeraScan ground station capable of receiving telemetry from the Advanced Very High Resolution Radiometer (AVHRR) aboard the National Oceanic and Atmospheric Administration (NOAA) satellite. Throughout this paper we refer to instances of such data as 'large objects'. A large object is a logically atomic array of bytes, which is typically greater than 8K in size.

Data from overpasses of the NOAA platform is received three times per day by a dish on the roof of Ellison Hall, and is dumped onto a single, large disk partition. This telemetry is an example of a real time data feed, pushed down from space, which must be stored with complete reliability since it can never be re-constructed. Once stored, this data is fed into one end of the processing pipeline shown in Figure 2.1, in order to produce the multi-band rasters which constitute AVHRR data.

A year of processing at ICESS generates about 1000, 12 Mbytes AVHRR rasters and requires the storage of 3000 intermediate images, a total of about 50 Gigabytes of data. The necessity of storing these intermediate steps will be explained shortly. This covers just the area visible in a single pass, roughly the three westernmost state of the United States, approximately 1/250th of the globe's surface area.

It is possible that, due to a bug in the code performing one of the steps or incorrect callibration table data, some sub-set of this processing must be redone. This explains why the storage of the intermediate files is desirable. Rather than re-submit raw data to the entire pipeline it would be better to repeat only elements of the process flow affected by the error. During the late 1980's, the Antarctic ozone hole went undetected for several years because code processing bit data interpreted values outside a certain range as erroneous readings and 'filled' the hole in. As EOSDIS is to be a supplier of quality data, it must include some means of repairing its holdings from this kind of problem.

From the large collection of material available, an investigator wishes to work with only what is relevant to their efforts. However, the scope of these efforts, both spatio-temporal and thematic, varies from investigator to investigator, and even from session to session. Sometimes investigators have questions relating to a small geographic region; the size of the body of water in an alpine lake, for example. Other times they may concern themselves with questions about warm currents which are as big as the oceans themselves!

Clearly, the interaction between a scientific user and the data EOSDIS manages needs to be very ad hoc. Users will work with data sets they identify by criteria like time, geo-spatial location and data lineage and they will vary these criteria as they attempt to discern significant patterns in the data. EOSDIS should be designed with this kind of flexible interaction between scientific users and the data in mind.

2.2. UCLA Department of Atmospheric Sciences (UCLA)

The UCLA Department of Atmospheric Sciences is constructing a computer simulation of the earth's atmosphere in order to predict climatic changes given certain observed trends. Such computer systems are based on conceptual models of the Earth's atmosphere, chemistry

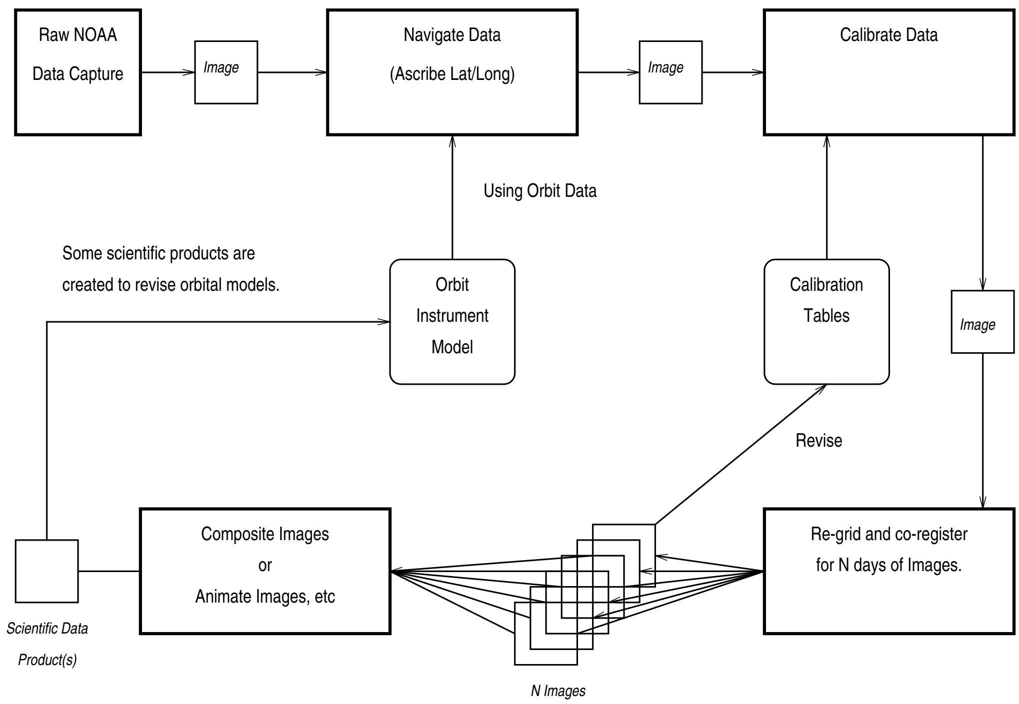

Figure 2.1: Data Processing Schema; NOAA Data to Scientific Data Products

and oceans. These models are derived from a set of differential equations representing the relationship between real world phenomenon; ground wetness, temperature, wind speed and direction. Given some set of initial conditions -- real world measurements of atmospheric conditions, for example -- these programs are designed to predict the behaviour of these variables decades into the future.

The UCLA General Circulation Model (GCM) tiles the atmosphere into a 72 x 44 x 15 array. For each element of this array, the program predicts the value of 23 variables. At set intervals, the state of the model is captured and stored along with enough metadata to uniquely identify the sample. Every simulated day generates ~20 Mbytes of data. A century of simulated time would produce about 700 Gigabytes of data, and require metadata entries for about 400 large objects.

The long term storage of model results is made necessary by the fact that over time, the aspect of the model's results which is of interest changes. Ten years ago the central Pacific Ocean was largely ignored. Then it was realised that the mid-Pacific oceanic current El Nino exerts tremendous influence over US weather patterns, so many historically uninteresting models were reinvestigated.

Within this array data, climate modelers look for atmospheric features like cyclones, or changes in ocean currents[MESR94]. Typically they wish to compare model run results, rendering of slices output identified by model run, initial conditions and time into the run ie "What did models X and Y predict for the mid-Pacific in the year 2000?" They also need to know what model run data they have stored. Once the model's data has been ingested by the system, investigators will be interact with it in an almost exclusively ad hoc mode using sophisticated rendering or animation software.

2.3. Integration

While the ultimate objective is to compare the predictions of computational models with actual observation, there are several immediate benefits in merging these data sets.

For example, there is the mutual benefit derived from sharing reference information. We have ingested a signifigant amount of point and polygonal survey data into BigSur which can be used to verify geo-rectification by comparing the course of a stream in the image with precise ground surveys. The atmospheric science group wishes to systematize the way they verify their models by ingesting other modeler's results and data systematically gathered from weather balloons. We also anticipate that BigSur will foster the co-development of client applications flexible enough to be used in both environments.

Ultimately however, BigSur's objective is the same as NASA's EOSDIS: providing an environment to evaluate competing conceptual models of the earth system by comparing their predictions with actual data. This comparison requires that we perform a spatial join over appropriately processed data from these two sources.

2.4. Summary of Technical Requirements

BigSur must find solutions for several technical problems in order to fulfill its user requirements. These problems are the same ones faced by EOSDIS: only BigSur's scale is smaller.

1. Scallability. BigSur handles about 1 Terabyte of data, about one thousandth of what EOSDIS will be obliged to manage and process. Every feature of BigSur's design and implementation must address this.
2. Distribution. BigSur must manage the movemement of data and the running of processes over widely distributed sites.
3. Parallelism. The only way to perform the processing and input/output BigSur is required to deliver, given the resources at our disposal, is to design the system to take advantage of parallelism at every opportunity.

3. BigSur's Architecture

In the BigSur prototype of EOSDIS, we have adopted a DBMS centric view of this task. Each data set has been added to an Illustra database containing the BigSur metadata schema, and BigSur provides location transparent access to the entire holdings. All interactions with data in this schema are in the form of Illustra SQL, a dialect of SQL which is extended to include new data types and functions. This results in a clean, simple, consistent interface which provides the necessary ad hoc flexibility and permits BigSur all the advantages of a DBMS; minimized redundancy, consistency, a ready means to share data among users, enforced standards and data integrity, and sophisticated security features.

Users work with the data sets BigSur manages through one of a variety of client applications which may formulate the Illustra SQL. The needs of each user group are sufficiently specific that each requires the ability to write its own applications, and the political reality of EOSDIS is that a single, ubiquitious application would be counter-productive. The UCLA team requires sophisticated 3d rendering tools which permit them to navigate their model output, while UCSB uses a mix of 2d renderers, public domain and commercial image processing toolkits, all in a heterogeneous computer environment.

In order to present a consistent API to user programs, and provide an environment in which processing of data may be done in a parallel, location transparent way, we have wrapped the DBMS in a layer of software glue. Ideally, BigSur would use a fully distributed DBMS like Mariposa[STON94], which handles location transparency as part of its core functionality. Unfortunately there is no commercial system of this type available so we have been obliged to implement a rudimentery subset of these features as extensions to Illustra SQL. Also, BigSur must move data objects among participating sites, a requirement which precludes the use of traditional distributed DBMSs which simply partition tables and perform query decomposition. BigSur uses the popular Tcl/Tk package for this because it is a flexible GUI application development tool, a complete scripting language, and RPC environment.

Each site participating in the BigSur system runs a local installation of the Illustra ORDBMS and a set of the middleware services. User applications write queries against their local DBMS. Metadata in each database is replicated between participating sites, which are linked via a high bandwidth network, necessary to acommodate the large volumes of data which must be moved around. Large objects are allocated a unique handle -- similar to a Universal Resource Locator (URL) -- which allows any BigSur site location transparent access to the object. As the amount of metadata which must be copied around is typically a minute fraction of a single large object, this approach should scale well. An additional advantage of this approach is its backup and recovery utility. If a failure causes the loss of metadata at one site, BigSur simply synchronizes it with one of its replication peers.

This architecture is shown in Fig 3.1. We shall discuss the design and implementation of each component in subsequent sections.

3.1. Illustra DBMS and the BigSur Metadata Schema

As we have described, EOSDIS will be the repository for order 10^7 large data objects from which an investigator wishes to examine a small set in a timely, ad hoc fashion. This is clearly a data management problem for which a DBMS is the appropriate solution. Such a system is well within the tolerances of existing DBMSs which use a query language like SQL. Further, the consolidation of information about the total holdings in a DBMS will greatly assist in the management of the BigSur's space resources. Because of the location transparency afforded by the middleware layer, data may be moved from site to site in accordance with some optimization strategy.

However earth science data is too complex to be readily managed using commercial systems, many of which are predicated on a small fixed library of data types and designed for industrial applications where the tendency is toward small, simple data structures. This complexity has up until now been the major inhibiting factor in the development of DBMS centric systems for the management of earth science data. For example, we are confronted with an abundance of file format standards for raster and image data (Net-CDF, HDF, Grib) each associated with a library of code. Remote sensing scientists deal with more than just raster images because their objectives, thematic mapping, elevation models, topological mapping, and field reportage, requires that they be able to handle, vector, point and textual information. As time passes the number of earth science data types will increase to the point where this aspect of the system becomes an exercise in data management on its own, so whatever benefits acrue from writing a one-off specialist DBMS would erode quickly.

In creating a metadata schema, BigSur was required to resolve a paradox. On the one hand, each participating site wanted a degree of schema autonomy. Each had a small but signifigant set of attributes which were specific to their science. Historically this has led to the development of distinct schemas for every earth science group. Yet these groups wanted to cooperate, and to share their data, which required a degree of consistency in the schemas they used.

Features of the Illustra Object Relational DBMS helped us to resolve each of these problems. The Illustra ORDBMS includes a powerful, extensible type mechanism. New types and functions are easily added to the library available and become elements in the Illustra SQL

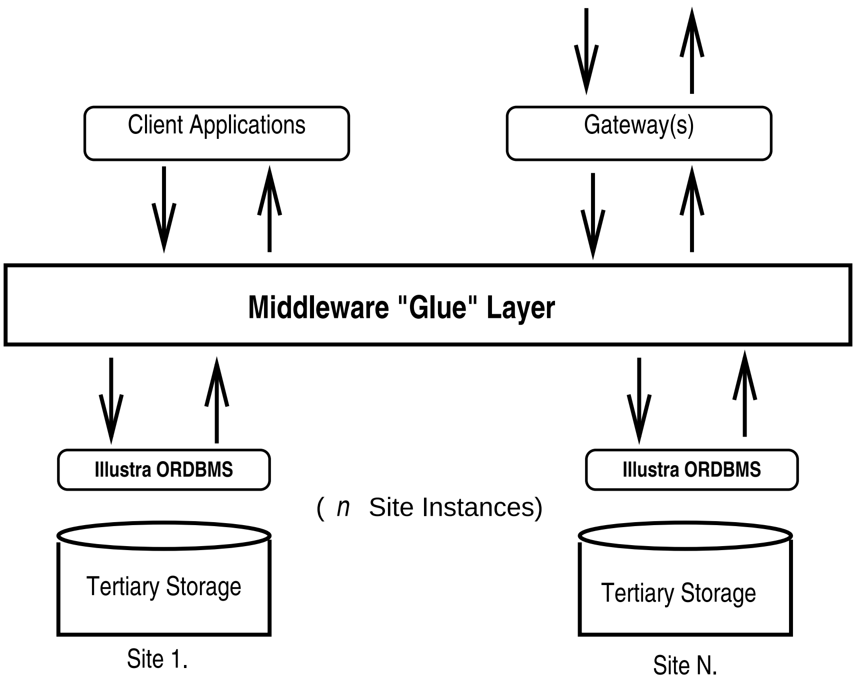

Figure 3.1: BigSur Architecture

relational query language. We have written type libraries for the plethora of file formats endemic to scientific data processing. Our users are abstracted from this engineering detail -- their SQL queries return consistent results regardless of physical organization -- but if information about underlying file formatting is desired it is stored as part of the metadata. We use the Illustra Image and 2d Spatial type libraries to pose sophisticated GIS style queries over raster and spatial types[ILL95]. Much of BigSur's functionality is implemented as extensions to the SQL query language in Illustra.

The Illustra ORDBMS allows schema designers to use many object oriented data modelling techniques like inheritance. In order to overcome the difficult problem of different schemas in each user's domain, we created a library of metadata super-classes which are inherited into BigSur's domain specific schemas. The design of this class library is discussed below. Where extra attributes were required, the super-classes were extended by their sub-class. However, inheritance on its own wasn't sufficient. We also used standard relational views combined with query rewrite rules to provide a simplified view to to BigSur's users which did not violate the integrity of the underlying schema.

The design of the metadata super-class library introduced above was far from trivial. Much energy had been invested in designing standards for metadata schemas, and our prototype leveraged these efforts as much as possible, extending them only where they fell short of our requirements. [JTA94] This library of super-classes, what we refer to as the BigSur schema, is divided into three modules. The first is based on the US Federal Government Data Committee *Content Standards for Digital Spatial Metadata* (FGDC) [FGDC94], and it stores the identification information and general audit information for each large object. This module includes the data's origin and general spatio- temporal information, locates the large object in the real world, authorial and citation information, as well as security and a set of audit trail information about the metadata itself. This is the general catalog of all information available in BigSur, and it was inherited without much ornamentation by each user group.

The second section of the schema manages a complex set of information which describe the contents of each large object in considerable detail. This section is called the BigSurObject module. Each BigSur large object is of a type we call ScientificArray, which inherits the location transparent type introduced above. The ScientificArray is a multidimensional data type where the individual elements of the n-array may be complex objects. ScientificArrays support operations like HyperSlab(), which takes a ScientificArray and arguements describing some portion of it which is to be 'sliced' out. The BigSurObject portion of the schema ties each ScientificArray to the real world.

For example, an AVHRR image is a ScientificArray with three dimensions. Two of these correspond to X and Y coordinates describing the geographical extent of the Image's coverage. Each element of the third dimension corresponds to one AVHRR band. Raster files are arrays of double precision floating point numbers, a fact which is also stored in the ScientificArray type. In the BigSurObject we fix the ScientificArray's origin to some geospatial coordinates. The BigSurObject portion of the schema was based on the *Spatial Archive and Interchange Format* (SAIF)[SAIF94].

This section of the schema is not mandatory. Some users are interested in highly unstructured text data OCRed from field journals, or more conventional relational tables of data, so BigSur does not require that these super-classes are used.

One area where none of the standards provided direction was lineage, the reproducible audit trail for every item, which is the third module in the BigSur schema. When a user adds a new scientific function, it is cataloged along with the ordered set of the parameters it requires. When a new data object is added to the BigSur schema -- which is synonymous with saying that one of these functions has been run -- we store the function creating it, and the arguements passed as each of its parameters, with the new object's other metadata. We give more detail about this aspect of BigSur's data management in Section 3.2 when we discuss the processing environment.

Illustra is a mature DBMS, complete with concurrency control, transaction processing and query optimizing features. It's novel 'no overwrite' storage system means that writers rarely block readers and allows us to implement a system with Degree 3 consistency.

3.2. Middleware "Glue" Layer

The BigSur prototype must do more than act as a simple repository. It is required to manage the processing of data, as well as its storage. Figure 3.2 shows how such a processing scheme looks.

Within the pipeline, some process P1 (say 'Calibrate' satellite imagery, for example) is run over data added by P0 ('Ingest' from a sensor's bit stream) This new data object is then subjected to another processing step P2 ('Navigate') the results of which may be used by other processes, and so on. Each process adds metadata -- including lineage information which catalogs whatever data objects the process took as input -- to BigSur's schema. There may be many of these pipelines running concurrently over data from distinct P0 processes, with each image at a different stage of processing. Output from any of these processes may be input to more than one

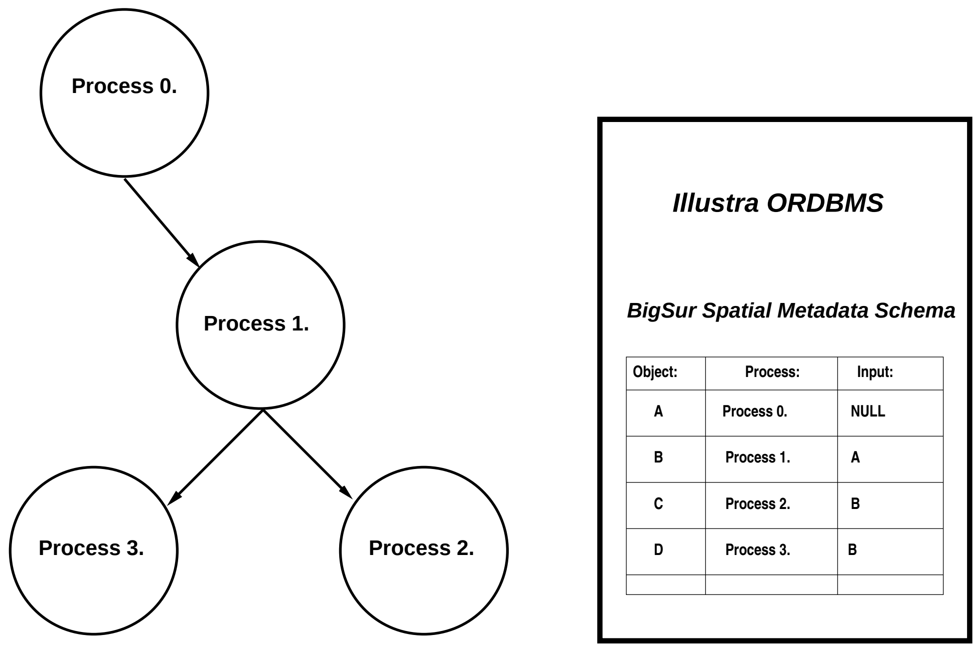

Figure 3.2: Process Pipeline and Resulting Lineage

other process, and the production of a single large object may require several others as arguements, so the relationships among processes within BigSur may characterised as a directed acyclic graph.

The first thing the middleware glue layer provides is a means by which this processing of data may occur in a heterogenus, distributed environment. In BigSur, these processes are initiated by Illustra SQL queries, and each requires considerable computer resources to resolve. The only way this can be made scalable is by designing BigSur's middleware layer as a completely parallel system, which performs load balancing optimization, moves intermediate data and does so in a fault tolerant way.

At each BigSur site, clusters of machines which are nominated as processing clients register themselves with a schema in the Illustra database. When an expensive user function is to be run, an Illustra function selects the best machine available based on a matrix of statistics about each machine; CPU architecture, available memory and disk, and so on. The expensive function is scheduled to the selected machine where it is run, while the Illustra server process waits for it to complete. Given hardware of sufficient power for the Illustra server this design simply requires the addition of processing clients to scale up.

One of the key decisions facing EOSDIS is how much of this work to do in advance of any demand for the data, and how much to leave undone until it is demanded. We call these two modes of processing 'Eager' and 'Lazy' respectively. It does not seem feasible to pre-compute every process pipeline. On the other hand, some popular data is obviously better off being done in anticipation of demand for it. In BigSur we provide mechanisms for both processing modes. Once we have been in production for some time we will work toward a scheme where as much data is pre-computed as we have hardware resources available, basing our priorities on what data products BigSur's users demand most frequently.

The DBMS centric architecture of BigSur provides a way to prevent redundant processing. Because each large object stores details about how the process creating it was invoked, if the process is invoked again with the same arguements, this can be detected from the state of the database. Thus if the same query is submitted to the Illustra database twice, the first time it will create the large object it returns, and the second time it simply returns the large object created initially. Further, because of BigSur's distributed design, if a process has been run anywhere then it has been run everywhere, ensuring optimal use of computer and storage resources. This example illustrates the first benefit of the integration of lineage and workflow within the DBMS.

A second benefit acrues when we consider 'Eager' evaluation. The Illustra ORDBMS includes the active database features common to commercial RDBMS systems. In BigSur, Illustra's rule mechanism is used to raise an alert whenever a process adds data. If the data just added is necessary to complete another process in the pipeline, then this Eager process may be invoked automatically . When an investigator adds something new to BigSur's catalog of scientific data processes, they indicate which other function's output could be used as input to it. In this way they connect their new function into the process pipeline. If they mark this relationship as an Eager link, their new process is run whenever sufficient data becomes available from prior process. Thus investigators may co-opt each other's results into their own, and 'plug in' to each other's work flows, features which actively promotes the kind of cooperation NASA must foster in EOSDIS.

This scheme can be extended to implement a partially materialized view. When a user issues an Illustra SQL query which requests the results of a particular process, but the perquisite data to run that process is not available, BigSur has all the information it needs to create what is needed by invoking another function. Thus, the only reason that even the most highly processed data is unavailable is if no raw bit stream with the desired properties exists. This feature gives scientific investigators enormous leverage over their data set. In addition, it means that BigSur can limit its processing to precisely that which is required by end users, which has tremendous potential for reducing the computational burden on the system.

Lineage and workflow management in a DBMS is a powerful innovation. It allows researchers to understand how the data they are examining was created. It provides a means by which errors within the process pipeline may be detected. When a process is found to have a problem, it is an easy task to find out which of the myriad of data objects is tainted and needs to be re-produced. And through the mechanism of the partially materialized view it provides leverage over both the scientific and technical problems of EOSDIS.

3.3. Gateways and Client Applications

EOSDIS will need to interoperate with a variety of external systems; repositories of older remote sensing imagery, other earth science data systems, and systems with only an orthogonal interest in earth science, like libraries. This is handled in BigSur by writing gateways to the middlware API. For example, we will be providing a Z39.50 gateway which will allow BigSur to interoperate with library systems, and an ODBC interface allows users to develop applications using the popular Microsoft Windows and NT client toolkits.

In order to accommodate the needs of the Internet 20 million, we have a HTTP gateway allowing our user's to write World Wide Web applications. We have prototyped several small client applications using his approach.

3.4. Storage Management

The daunting question of EOSDIS's size presents implementors with several difficulties. We expect that EOSDIS data will be accessible without human intervention which implies that the bulk of it will reside on a hierachical tertiary storage, the vast bulk of which will be near-line tape. Currently we are using a 10 Terabyte Metrum tape robot a and group HP optical disk juke boxes, which the Illustra ORDBMS views as a collection of file systems. It seems likely that users will not wish to access all data with equal indifference. Recent satellite imagery, for example, will be the subject of more interest than older data, so it is reasonable to expect some of it to reside on magnetic disk, avoiding the long mount and seek times inherent in tape robot storage. How such a large database can be secured against total system failure is an interesting research question, but in BigSur we simply write every large object twice.

This redundant write strategy affords us fault tolerance, as well as recovery. When an access request is made for an instance of a ScientificArray type, the function responsible for returning the data interrogates its environment to discover where the most available copy of the object is stored. If one of the devices is inoperative, the object is retrieved off the alternate. More typically however, the user requests a subset of data from within the file. Because the Illustra ORDBMS has no control over the caching strategies used by these systems, and can only get file level -- rather than block level -- access, this approach is far from optimal.

Ideally, the DBMS servicing BigSur should have more control over the placement of data on the various physical media in the hierachical storage. With more information about data location at its disposal the DBMS engine could make better oprimizing choices about query execution, and prefetch data from slow to faster storage to improve response time. In addition, because the DBMS is in the best position to store information about access patterns, it can run more sophisticated caching algorithms than those available in the absence of complete information. For example, the DBMS may be able to perceive that several requests have been made for data which reside on a single tape, or determine which of the blocks in a full cache is the least commonly used.

We are working with the developers of the High Performance Storage System at Lawrence Livermore Labs on these issues. [HPSS93]

4. Conclusions

We have introduced BigSur, an innovative prototype system for the management of earth science data. Specifically, it was designed to fulfill the requirements of two scientific communities, but its broader goal was to demonstrate how NASA's EOSDIS could be constructed. It's DBMS centric architecture facilitates the kind of information sharing necessary for effective scientific collaboration because it provides an environment in which to integrate multiple data sets.

The BigSur prototype demonstrates the feasibility of a quite novel approach to the management of earth science data. Its focus on the use of an ORDBMS for data management -- a tenet of our design -- provides solutions to the problems of complex type and process management common in earth science information systems. Type and function extensibility overcomes many of the reservations scientific users have expressed about the use of DBMS technology. Object oriented data modeling techniques like inheritance ensures that each research group remains independent of others while ensuring that collaboration is possible.

The integration of lineage information with workflow made possible by the use of an extensible DBMS system is a powerful innovation. It allows the abstraction of the BigSur's holdings into a view of the data processed to varying degrees, while ensuring that only as much work is done as is required by the user community. As users add processes and types to the system the variety of data available to them grows accordingly. This approach allows users to leverage each other's efforts, promoting the desired synergies between research disciplines. From being at best an occasional adjunct to earth science metadata, BigSur shows that lineage information is central to the efficient production of quality data sets.

We intend to continue with the incremental development of BigSur, adding a number of other user groups, and revisiting the vexatious topic of schema design. At this point in time we are limited by the availability of stable tertiary storage devices and network bandwidth, though we will address our attention to these concerns once the core features of BigSur are stable.

References

[ALT94] Davis, Frank, William Farrell, Jim Gray, C. Roberto Mechoso, Reagan Moore, Stephanie Sides, Michael Stonebraker, *EOSDIS Alternative Architecture*, Final Report Contract #ECS-00012, Submitted to HAIS, September 6th 1994.

[DOZ92] Dozier, Jeff. "How Sequoia 2000 Addresses Issues in Data and Information Systems for " Global Change," Sequoia Technical Report 92/14. UC Berkeley, Berkeley, CA, 1992.

[FGDC94] Federal Geographic Data Committee, *Content Standards for Digital Geospatial Metadata,* Washington, D.C., June 8, 1994.

[ILL95] Illustra Information Technologies, Inc. *Illustra User's Guide: Release 2.4.1*, Oakland, CA, March 1995.

[JTA94] Anderson, J.T. and M. Stonebraker, "Sequoia 2000 Metadata Schema for Satellite Images", *SIGMOD Special Issue on Metadata for Digital Medias, Vol. 23, No. 4,* December 1994.

[MESR94] Mesrobian, E., R.R. Muntz, J.R. Santos, E.C. Shek, C.R. Mechoso, J.D. Farrara, and P. Stolorz, "Extracting Spatio-Temporal Patterns from Geoscience Datasets". *IEEE Workshop on Visualization and Machine Vision*, Seattle, WA, IEEE Computer Society, June 1994.

[NASA91] NASA, *EOS Reference Manual*, NASA Goddard Space Flight Center, Greenbelt, MD, 1991.

[SAIF94] Surveys and Resource Mapping Branch, *Spatial Archive and Interchange Format: Formal Definition Release 3.1,* Province of British Columbia, Ministry of Environment, Lands and Parks, April 1994.

[STON93] Stonebraker, Michael, James Frew, and Jeff Dozier, "The Sequoia 2000 Architecture and Implementation Strategy", Sequoia Technical Report 93/23, University of California, Berkeley, CA, April 1993.

[STON94] Stonebraker, M., Aoki, P. M., Devine, R., Litwin, W. and Olson, M., "Mariposa: A New Architecture for Distributed Data," *Proc. 10th Int. Conf. on Data Engineering*, Houston, TX, Feb. 1994, pp. 54-65.

[HPSS93] R. Coyne, H. Hulen, and R. Watson, "The High Performance Storage System", *Proc. Supercomputing '93*, Portland, OR, November 15-19, 1993.