THE SEQUOIA 2000 ARCHITECTURE AND IMPLEMENTATION STRATEGY

Michael Stonebraker and James Frew
Department of Electrical Engineering and Computer Science
University of California, Berkeley

Jeff Dozier
Center for Remote Sensing and Environmental Optics
and
Department of Geography
University of California, Santa Barbara

Sequoia 2000 Technical Report 93/23
University of California
Berkeley, CA 94720

Abstract

This paper describes the Sequoia 2000 software architecture and its current implementations, including layers for Footprint, the file system, the DBMS, applications, and the network. Early prototype applications of this software include a Global Change data schema, GCM integration, remote sensing, a data system for climate studies, and operational uses by the DWR. Longer-range efforts include transfer protocols for moving elements of the database, controllers for secondary and tertiary storage, distributed file system, and a distributed DBMS. The implementation plan ensures that the current architecture is stabilized and robust by the end of 1993.

Contents

1 Introduction

The purpose of the Sequoia 2000 project is to build a better computing environment for global change researchers, hereinafter referred to as Sequoia 2000 "clients." Global change researchers investigate issues of global warming, the Earth's radiation balance, the oceans' role in climate, ozone depletion and its effect on ocean productivity, snow hydrology and hydrochemistry, environmental toxification, species extinction, vegetation distribution, etc., and are members of Earth science departments at universities and national laboratories. A cooperative project among five campuses of the University of California, government agencies, and industry, Sequoia 2000 is Digital Equipment Corporation's (DEC) flagship research project for the 1990s, succeeding Project Athena at MIT. It is an example of the close relationship that must exist between technology and applications to foster the computing environment of the future [NRC92].

There are four categories of investigators participating in Sequoia 2000:

Computer science researchers are affiliated with the Computer Science Division at UC Berkeley, the Computer Science Department at UC San Diego, the School of Library and Information Studies at UC Berkeley, and the San Diego Supercomputer Center. Their charge is to build a prototype environment that better serves the needs of the clients.

Earth science researchers are affiliated with the Department of Geography at UC Santa Barbara, the Atmospheric Science Department at UC Los Angeles, the Climate Research Division at the Scripps Institution of Oceanography, and the Department of Land, Air and Water Resources at UC Davis. Their charge is to explain their needs to the computer science researchers and to use the resulting prototype environment to do better Earth science.

Government agencies include the State of California Department of Water Resources (DWR), the Construction Engineering Research Laboratory (CERL) of the U.S. Army Corps of Engineers, the National Aeronautics and Space Administration (NASA), and the United States Geological Survey (USGS). Their charge is to steer Sequoia 2000 research in a direction that is applicable to their problems.

Industrial participants (other than DEC) include Epoch Systems Inc., Hewlett-Packard, Hughes, MCI, Metrum Corp., PictureTel Corp., Research Systems Inc. (RSI), Science Applications International Corp. (SAIC), Siemens, and TRW. Their charge is to use the Sequoia 2000 technology and offer guidance and research directions. They are also a source of computing equipment grants and allowances.

The purpose of this document is to explain the computing architecture that Sequoia 2000 has adopted, the implementations of this architecture that will be delivered during 1993, enhancements planned for 1994 or beyond, and the schedule and responsibilities for the near-term deliveries. Section 2 describes the architecture that we are pursuing and explores specific implementations of this architecture in detail. Section 3 explores three different themes that cross most elements of the architecture. Section 4 discusses proposed use of the prototype system by Sequoia 2000 clients, and their expected benefits. Section 5 discusses the longer-term agenda for research and prototyping. Section 6 lays out the schedule, responsibilities, and deliverables.

2 The Sequoia 2000 Architecture

2.1 Objectives

The Sequoia 2000 architecture is motivated by four fundamental computer science objectives:

- big fast storage;
- an all-embracing DBMS;
- integrated visualization tools;
- high-speed networking.

2.1.1 High Performance I/O on Terabyte Data Sets

Our clients are frustrated by current computing environments because they cannot effectively manage, store, and access the massive amounts of data that their research requires. They would like high-performance system software that would effectively support assorted tertiary storage devices. Collectively, our Earth science clients plus DWR would like to store about 100 terabytes of data now. Many of these are common data sets, used by multiple investigators.

Unlike some other scientific computing users, much of our clients' I/O activity is random access. For example, several investigators use image data from the Landsat Thematic Mapper. Sometimes they want the most current image for a specific area, sometimes they want to examine a time sequence of mosaicked images for a larger area. Similarly, DWR is digitizing the agency's library of 500,000 photographic slides, and will put it on-line using the Sequoia 2000 environment. This data set will have some locality of reference but will have considerable random activity.

2.1.2 All Data in a DBMS

Our clients agree on the merits of moving all their data to a database management system (DBMS). In this way, the metadata that describe their data sets can be maintained, assisting them with the ability to retrieve needed information. A more important benefit is the sharing of information it will allow, thus enabling intercampus, interdisciplinary research. Because a DBMS will insist on a common schema for shared information, it will allow the researchers to define this schema; then all must use a common notation for shared data. This will improve the current confused state, whereby every data set exists in a different format and must be converted by any researcher who wishes to use it.

2.1.3 Better Visualization Tools

Our clients use visualization tools such as AVS, IDL, Khoros, and Explorer. They are frustrated by aspects of these tools and are anxious for a next-generation visualization toolkit that:

- allows better management, use, and manipulation of large data sets and model output;
- provides better interactive data analysis tools, including comparison of data sets and integration and composition of dissimilar data;
- fully exploits the capabilities of a distributed, heterogeneous computing environment, including workstations, large vector machines, and massively parallel processors;
- produces presentation materials that effectively convey information about the data sets presented;
- uses "computational steering" techniques to guide models during execution.

2.1.4 High-Speed Networking

Our clients realize that 100-terabyte storage servers will not be located on their desktops; instead, they are likely to be at the far end of a wide-area network (WAN). Their visualization scenarios often make heavy use of animation, (e.g., “playing” the last 10 years of ozone hole imagery as frames of a movie), which requires ultra-high-speed networking with real-time communication services.

2.2 Details About the Sequoia 2000 Architecture

As described in Figure 1, the Sequoia 2000 architecture is divided into four layers. Figure 2 shows the prototype implementations that we have running or planned. The rest of this section explores the various boxes in Figure 2. Schedules for planned development and deployment are in Section 6.

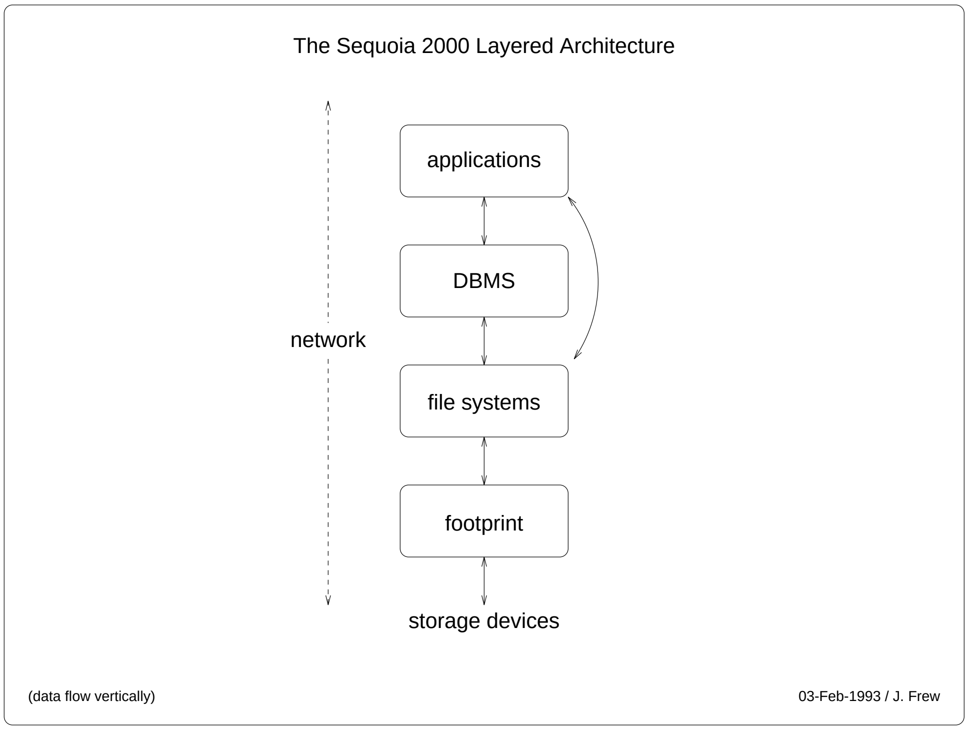

Figure 1: Sequoia 2000 layered architecture

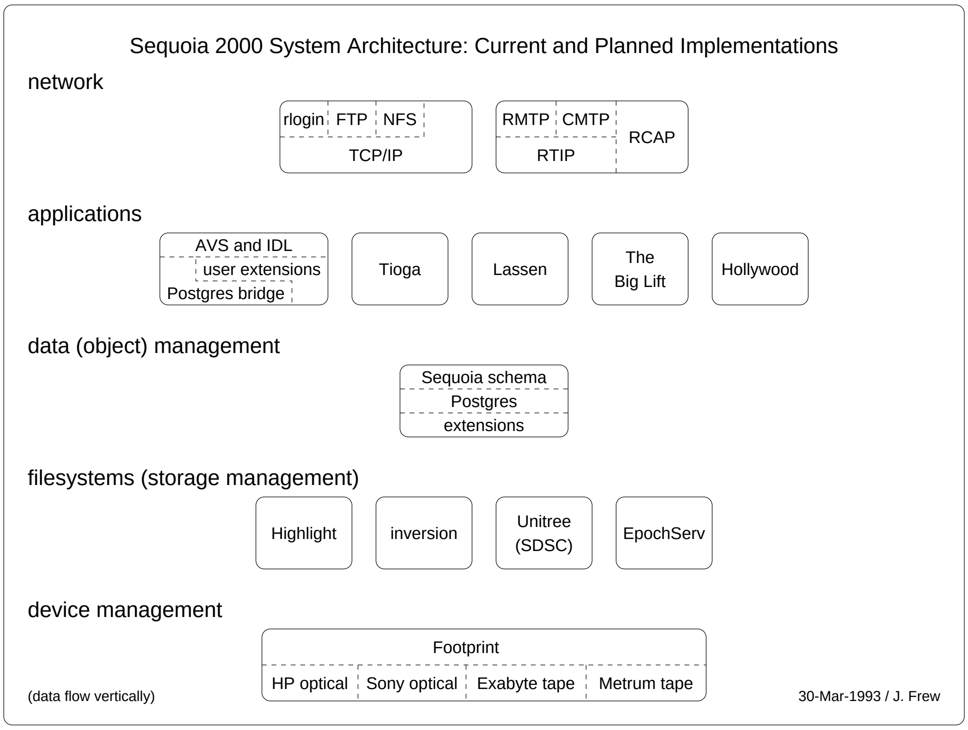

Figure 2: Sequoia 2000 architecture implementations

2.2.1 The Footprint Layer

Footprint is a generic programming interface for robotic storage devices ("jukeboxes"). The Footprint software shields higher level software, such as file systems, from device-specific characteristics of robotic devices, such as specific robot commands, block sizes, and media-specific issues. We currently have a Footprint implementation for each of the four robotic storage devices used by the project: Sony WORM optical disk, HP rewritable optical disk, Metrum VHS tape, and Exabyte 8mm tape.

The robotic storage devices and their associated CPUs and secondary (magnetic disk) storage are collectively called **Bigfoot** after the legendary gigantic ape-man of the Pacific Northwest. Bigfoot is currently deployed on DECstation hardware running the ULTRIX operating system. Later in 1993 or perhaps in 1994, we will move Bigfoot to DEC Alpha platforms running either the OSF/1 or Windows NT operating system.

2.2.2 The File System Layer

On top of Footprint, we plan to support four different file systems that will manage data in the Bigfoot multi-level storage hierarchy. Two of these file systems are academic prototypes, written by Sequoia 2000 researchers, and two are commercial products. All file systems will support a standard UNIX file system interface.

The first file system is **Highlight** [Kohl93]. It is an extension of the Log-structured File System (LFS) pioneered for disk devices by Rosenblum and Ousterhout [Ros92]. LFS treats a disk device as a single continuous **log** onto which newly-written disk blocks are appended. Blocks are never overwritten, so a disk device can always be written sequentially. In particular problem areas, this may lead to much higher performance [Selt90, Selt93]. LFS also has the advantage of rapid recovery from a system crash: potentially damaged blocks in an LFS are easily found, because the last few blocks that were written prior to a crash are always at the end of the log. Conventional file systems require much more laborious checking to ascertain their integrity.

Highlight extends LFS to support tertiary storage by adding a second log-structured file system on top of Footprint, plus migration and bookkeeping code that treats the disk LFS as a cache for the tertiary storage one. Highlight should give excellent performance on a workload that is "write-mostly." This should be an excellent match to the Sequoia 2000 environment, whose clients want to archive vast amounts of data.

The second file system is **Inversion** [Ston93a], which is built on top of the POSTGRES DBMS. Like most DBMSs, POSTGRES supports binary large objects (blobs), which can contain an arbitrary number of variable-length byte strings. These large objects are stored in a customized storage system directly on a **raw** (i.e. non-file-structure) storage device. It is a straightforward exercise to have the DBMS make these large objects appear to be conventional files. Every read or write is turned by the DBMS front end into a query or update, which is processed directly by the DBMS.

Simulating files on top of DBMS large objects has several advantages. First, DBMS services such as transaction management and security are automatically supported for files. In addition, novel characteristics of POSTGRES, including **time travel** and an extensible type system for all DBMS objects [Ston91b], are automatically available for files. Of course, the possible disadvantage of files on top of a DBMS is poor performance, but our experiments show that Inversion performance is exceedingly good when large amounts of data are read and written [Ols93], a characteristic of the Sequoia 2000 workload.

Our third file system is **UniTree** [Hos90, GA91], originally written by Lawrence

Livermore Laboratory and currently licensed to General Atomics (GA), who operate the San Diego Supercomputer Center in partnership with the University of California. There are UniTree implementation for many popular platforms, and GA is porting UniTree to the ULTRIX/Footprint platform adopted by Sequoia 2000. Inclusion of UniTree will allow Sequoia 2000 clients to use a commercial, presumably robust, file system for tertiary storage.

Our fourth file system is **EpochServ** [Epoc92], another commercial file system. EpochServ was chosen to provide a second highly robust tertiary storage file system for Sequoia 2000 data.

We plan to conduct a "bakeoff" of the four file systems on all four of our robotic storage devices, using two large benchmarks. The first is the national version of the Sequoia 2000 benchmark, a 25-Gbyte dataset and associated queries, specified as a project standard [Ston93b]. The second benchmark is a scientific and engineering workload derived from a tracing study of the Cray supercomputer at the National Center for Atmospheric Research (NCAR) [Mill92]. The purpose of the bakeoff is to ensure that all Sequoia 2000 file systems are robust, and to help Sequoia 2000 clients identify the file system that would best serve their particular applications.

2.2.3 The DBMS Layer

Some users will simply run application programs against the file system, and will have no use for DBMS technology. Others will store their data in a DBMS. To have any chance of meeting Sequoia 2000 client needs, a DBMS must support spatial data structures such as points, lines, polygons, and large multidimensional arrays (e.g. satellite images). Currently these data are not supported by popular general-purpose relational and object-oriented DBMSs [Ston91, Doz92]. The best fit to Sequoia 2000 client needs would be either a special-purpose Geographic Information System (GIS), or a next-generation prototype DBMS. Since we have one such next-generation system within the project, we have elected to focus our DBMS work on this system, POSTGRES[Ston90, Ston91b].

To make POSTGRES suitable for Sequoia 2000 use, we require a **schema** for all Sequoia 2000 data. This database design process is evolving as a cooperative exercise between various database experts at Berkeley, SDSC, CERL, and SAIC. As we develop the schema, we are loading it with several terabytes of client data; we expect this load process to continue for the duration of the project. As the schema evolves, some of the already-loaded data will need to be reformatted. How to reformat a multi-terabyte database in finite time is an open question that is troubling us.

In addition to schema development, we are tuning POSTGRES to meet the needs of our clients. The interface to POSTGRES arrays is being improved, and a novel **chunking** strategy [Sara93] is being prototyped. The R-tree access method in POSTGRES is being extended to support the full range of Sequoia 2000 spatial objects.

2.2.4 The Application Layer

There are five elements of our application layer:

- AVS and IDL—commercial visualization software;
- Tioga—next-generation visualization and recipe-management tools;
- Lassen—browsing for textual information;
- The Big Lift—link between a global circulation model and POSTGRES;
- Hollywood—video teleconferencing.

AVS and IDL: Sequoia 2000 has standardized on IDL and AVS as our "official" off-the-shelf visualization software packages. AVS is liked for its easy-to-use "boxes and arrows" user interface, while IDL has a more conventional procedural programming notation. On the other hand, IDL is liked for its more flexible 2D graphics features. Both IDL and AVS allow a user to read and write file data.

To connect to the DBMS, we have written an AVS-POSTGRES bridge [Koch93]. This program allows one to construct an ad-hoc POSTGRES query and pipe the result into an AVS boxes-and-arrows network. Our clients can thus use AVS for further processing of any data retrieved from the DBMS. IDL is being interfaced to AVS by the vendor, so data retrieved from the database will be moved into IDL using AVS as an intermediary.

Tioga: AVS has a collection of severe disadvantages as a visualization tool for our clients:

- A type system that is different from the POSTGRES type system, without direct knowledge of the common Sequoia 2000 schema.
- A severe appetite for main memory. AVS depends on virtual memory to pass results between various boxes. It maintains the output of each box in virtual memory for the duration of an execution session, so if a user changes a run-time parameter somewhere in the network, AVS will recompute only the

"downstream" boxes, by taking advantage of the previous output. As a result, Sequoia 2000 clients, who produce large intermediate results, consume large amounts of both virtual and real memory: they report that 64 megabytes of real memory on a workstation is often not enough to enable serious AVS use.

- No support for "zooming" into data of interest to obtain higher resolution.
- No history of how any given data element was constructed, i.e. the so-called **data lineage** of an item.
- A "video player" model for animation, which is too primitive for many Sequoia 2000 clients.

To correct these deficiencies, we have designed **Tioga,** a new boxes-and-arrows programming environment that is "DBMS-centric," i.e. the environment's type system is the same as the DBMS type system. The user interface presents a "flight simulator" paradigm for browsing the output of a boxes-and-arrows network, allowing users to "navigate" around their data and then zoom in to obtain additional data on items of particular interest. Tioga [Ston92b] is a joint project between Berkeley and SDSC. A prototype "early Tioga" [Chen91] is currently running.

Lassen: The third element of our application layer is **Lassen,** a browsing capability for textual information. Lassen has two components. The first is **Cheshire** [Lars91], a facility for constructing weighted keyword indices for the words in a document, stored as an instance of some particular POSTGRES type. Cheshire builds on the pioneering work of the Cornell Smart system [Salt71] and operates as the action part of a POSTGRES rule [Ston92a] that is triggered on each document insertion, update, or removal. The second piece of Lassen is a front-end query tool with natural language understanding, allowing a user to ask for all documents that satisfy a collection of keywords, by inquiring in a subset of Natural English.

Lassen is now operational, and retrievals can be requested against the currently loaded collection of Sequoia 2000 documents. This document collection includes some (soon to be all) Berkeley Computer Science technical reports, a collection of DWR publications, the Berkeley Cognitive Science technical reports, and the technical reports from the UC Santa Barbara Center for Remote Sensing and Environmental Optics (CRSEO).

Over the next year, we expect to:

- Install phrase recognition software in Cheshire that will extend its indexing capabilities from single words to noun phases. Other research has shown this to be a good way to increase the precision of the answer to a query [Evan91].

- Move Lassen to a Z39.50 protocol [Lyn91, Lyn92, Zee92]. The client portion of Lassen would emit Z39.50 and we would write a Z39.50 to POSTGRES translator on the server side. This would allow the Lassen client code to access non-Sequoia 2000 information, and the Sequoia 2000 server to be accessed by text retrieval front ends other than Cheshire.
- Extend Lassen coverage to include non-document materials such as business cards, marketing reports, etc.

The Big Lift: Our fourth thrust in the application layer is a facility to interface the UCLA General Circulation Model (GCM) to POSTGRES. This interface is a "data pump" because it pumps data out the simulation model and into POSTGRES. As such, it has been named the **Big Lift** after the DWR pumping station that raises Northern California water over the Tehachapi Mountains into Southern California.

The UCLA GCM produces a vector of simulation output variables for each time step of a lengthy run, for each cell in a three-dimensional grid of atmosphere and ocean. Depending on the scale of the model, its resolution, and the capability of the serial or parallel machine on which the model is running, the UCLA GCM can produce anywhere from 0.1 to 10 Mbytes/sec of output. The purpose of Big Lift is to install these data into a POSTGRES database in real time. UCLA scientists will then use AVS or (eventually) Tioga to visualize their simulation output. It is likely that Big Lift will have to exploit parallelism in the data manager if it has to keep up with the execution of the model on a massively parallel architecture.

Hollywood: Since Sequoia 2000 is a distributed project, we learned early that airplane tickets and electronic mail did not keep project members working coherently as a distributed team. As a result, we purchased conference room videoteleconferencing equipment for each project site. This technology costs around $50,000 per site and allows multiway teleconferences over ISDN lines. In the longer run, we expect to move this equipment onto the Sequoia 2000 network (described below) to obtain higher-bandwidth video interactions.

Although the conference room equipment has helped project communication immensely, its use has to be scheduled because it occupies rooms at each site that are used for classes and faculty meetings. Thus, it mainly is used for carefully scheduled conferences, not for "spur of the moment" interactions. To alleviate this shortcoming, Sequoia 2000 has also invested in desktop videoteleconferencing. This project, **Hollywood,** uses a video compression board, microphone, speakers, network connection, video camera, and appropriate software to turn a conventional workstation into a desktop teleconferencing facility. Video can be easily transmitted over the network interface present in virtually all Sequoia 2000 client machines, and will benefit from the Sequoia 2000 guaranteed-delivery network services. Several commercial and public domain desktop video conferencing systems have been and are being developed (e.g., Communique! from InSoft, DECSpin, PictureWindows from BBN, and the INRIA Videoconferencing System). We plan to install and experiment with several of these systems.

It should be clearly noted that the Sequoia 2000 researchers are not clamoring for "groupware," i.e. the ability to have common windows on multiple client machines separated by a WAN, in which common code can be run, updated and inspected. Instead, our researchers need a way to hold impromptu discussions on project business. As such they want a low-cost multicast "picturephone" capability, and Hollywood's efforts are focused in this direction.

However, researchers have expressed interest in better tools for on-line presentations. Tools are needed that can: 1) project an on-line slide show at multiple sites concurrently during a video conference presentation and 2) play video and animations stored in a database. The video playback capability is essentially a video-on-demand server that can be accessed locally or remotely using the transfer protocol discussed below.

2.2.5 The Network Layer

In Figure 1, it is possible for the implementation of each layer to exist on a different machine. Specifically, the application can be remote from the DBMS, which can be remote from the file system, which can be remote from the storage device. Each layer assumes a local UNIX socket connection or a LAN or WAN connection using TCP/IP. Actual connections among Sequoia 2000 sites use either the Internet or a dedicated T1 (1.54 Mbit/sec) network, contributed to Sequoia 2000 by the University of California.

The Sequoia 2000 T1 network uses DECstation 5000's (soon to become Alphas) as routers, instead of "custom iron." The project will soon upgrade to T3 (45 Mbit/sec) lines, and the computer science researchers in charge of the network are confident that workstation-based routers will continue to be fast enough. Furthermore, the Sequoia 2000 network is installing a **guaranteed delivery** service, through which a client program can **contract** with the network to guarantee a specific bandwidth and latency if the client agrees not to try to send faster than the contract. This service uses the RTIP protocols instead of TCP/IP, and requires a "set-up" phase for a connection that will allocate bandwidth on all the lines and in all the switches. [Ferr90].

The network researchers are concerned that ULTRIX copies every byte four times between retrieving it from storage and sending it out over a network connection. Even Alphas may not be fast enough to overcome this bottleneck. We are modifying ULTRIX to "fast-path" network connections through the operating system, bypassing the redundant copyings.

3 Common Concerns

Four concerns of Sequoia 2000 researchers cannot be isolated to a single layer in the architecture:

- guaranteed delivery;
- abstracts;
- compression;
- integration with other software.

3.1 Guaranteed Delivery

Guaranteed delivery must be an **end-to-end contract.** Suppose a Sequoia 2000 client wishes to visualize a specific computation, for example, observing Hurricane Andrew as it moves from the Bahamas to Florida to Louisiana. Specifically, the client wishes to visualize appropriate satellite imagery at 500×500 resolution, in 8-bit color, at 10 frames per second. This requires 2.5 Mbytes/sec of bandwidth to the client's screen. The following scenario might be the resulting computation steps:

- The DBMS runs a query to fetch the satellite imagery. It might require returning a 16-bit data value for each pixel that will ultimately go to the screen, so the DBMS agrees to execute the query in such a way that it returns 5.0 Mbytes/sec.
- The storage system at the server fetches some number of I/O blocks from secondary and/or tertiary storage. DBMS query optimizers can accurately guess how many blocks they need to read to satisfy a query. It is an easy extension for the DBMS to generate a guaranteed delivery contract that the storage system must satisfy that will in turn allow the DBMS to satisfy its contract.

- The network agrees to deliver 5.0 Mbytes/sec over the link connecting the client to the server. The Sequoia 2000 network software is designed to accommodate exactly this sort of contract request.

- The visualization package agrees to translate the 16-bit pixels into 8-bit colors, and to render the result onto the screen at 2.5 Mbytes/sec.

In short, guaranteed delivery is a collection of contracts that must be adhered to by the DBMS, the visualization package, the storage system, and the network [Ston92b].

3.2 Abstracts

The Sequoia 2000 visualization process needs abstracts. Consider again the Hurricane Andrew example. Clients might initially want to browse the hurricane at 100$\times$100 resolution. Then, if they found something of interest, they would like to **zoom** in and increase the resolution, usually to the maximum available in the original data. This ability to change the amount of resolution in an image dynamically has been termed **abstracts** [Fine92].

Abstracts are a much more powerful construct than merely providing resolution adjustment. Obtaining more detail may entail moving from one representation to another. For example, one could have an icon for a document, zoom in to see the (textual) abstract, and then zoom in further to see the entire document. This use of abstracts was popularized in the DBMS community by SDMS [Her80].

Sequoia 2000 clients wish to have abstracts. However, they could be managed by any combination of the visualization tool, the network, the DBMS, or the file system. In the visualization tool case, abstracts are defined for boxes-and-arrows networks [Ston92b]. In the DBMS case, abstracts would be defined for individual data elements or for data classes. If the network manages abstracts, then it will use them to automatically lower resolution to eliminate congestion. Much research on the optimization of network abstracts (called hierarchical encoding of data in that community) has been presented [Dix91]. Lastly, in the file system case, abstracts would be defined for files. There are Sequoia 2000 researchers pursuing all four possibilities.

3.3 Compression

The Sequoia 2000 clients are open to any compression scheme as long as it is lossless. For many satellites, the characteristics of the sensor and the quantization and transmission of the data were designed around processing algorithms for interpretation of geophysical phenomena. Hence every bit is significant, and a lossy compression algorithm would probably introduce large errors into the interpretation of the data.

"Old" data also must be preserved. Twenty years ago, the equatorial Pacific Ocean was less interesting than in the last decade, when the El Niño has been discovered to affect weather patterns in the Western United States. Old data about El Niño are now central to many scientific research agendas. Such unpredictability of the future importance of data can be expected to continue indefinitely and leads to the decision to keep everything at its finest available resolution.

Some Sequoia 2000 data are not economically compressible, and should be stored in clear (uncompressed) form. For such data, the use of abstracts offers a mechanism to lower the bandwidth required between the storage device and the visualization program. However, little saving of tertiary storage space via compression is available for such data.

On the other hand, some Sequoia 2000 data are compressible and should be stored in compressed form. When should compression and decompression occur? The only concept that makes any sense is the principle of **just in time** decompression. For example, if the storage system compresses data as they are written and then decompresses them on a read, then the network system may then recompress the data for transmission over a WAN to a remote site where they will be decompressed again. Obviously, data should be moved in compressed form and only decompressed when necessary. In general, this will mean in the visualization system on the client machine. If the data are searched by some criteria, then the DBMS may have to decompress the data to search through them. Lastly, it is possible that an application resides on the same machine as the storage system. If so, the file system must be in charge of decompressing the data. All software modules in the Sequoia 2000 architecture must co-operate to decompress just-in-time and compress as-early-as-possible. Like guaranteed delivery, compression is a task where every element must cooperate.

3.4 Integration with Other Software

Sequoia 2000 researchers will always need access to other commercial and public-domain software packages. It would be a serious mistake for the project to develop every tool the researcher needs, or to add a needed function to our architecture when it can be provided by integration with another package. Sequoia 2000 thus needs "grease and glue," so that interface modules to other packages, e.g. S [Beck84], are easily written.

4 Use of Sequoia 2000 Environment

In this section we report on the planned utilization of the Sequoia 2000 environment by our clients. First, we report on the common data collection and schema construction effort. Then we consider the client prototyping efforts in four separate areas.

4.1 Schema Construction and Data Loading

The Sequoia 2000 schema is the collection of **metadata** describing the data stored in the POSTGRES DBMS on Bigfoot. Specifically, these metadata comprise:

- a standard **vocabulary** of terms with agreed-on definitions that are used to describe the data;
- a set of **types,** instances of which may store data values;
- a hierarchical collection of **classes** that describe aggregations of the basic types; and
- **functions** defined on the types and classes.

The Sequoia 2000 schema will define types for all fundamental phenomena (e.g., temperature). An instance of a type will comprise a value (e.g., 273), a precision (e.g., ± 1) and a coordinate system or unit of measure (e.g., degrees Kelvin). For each type, a canonical unit will be defined, to which differing units (e.g., Celsius) will be converted before they are compared or combined. Values with differing precisions will normally be compared or combined at the coarser precision. This strategy will also be used to support geographic coordinates in multiple projections.

The Sequoia 2000 schema will accommodate four broad categories of data: scalar, raster, vector, and text. Scalar quantities will be stored as POSTGRES types and assembled into classes in the usual way. Vector quantities will be stored in special line and polygon types, which will in turn comprise POSTGRES coordinate and attribute types. Vectors will be fully enumerated (as opposed to an arc-node representation) to take advantage of POSTGRES indexed searches.

Raster data will comprise the bulk of Sequoia 2000 data. These data will be stored in a Sequoia 2000 file system. The Sequoia 2000 schema will embed large-object pointers to the raster files in a generic multidimensional array data type, allowing the contents of the arrays to be accessed directly by POSTGRES functions. Text data will be treated similarly: the actual text (in PostScript, or scanned page bitmaps) will be stored in large objects, and the bibliographic information will be stored in the Sequoia 2000 schema.

Priorities for data loading have evolved according to three criteria:

- The dataset must have a **champion,** i.e., a Sequoia 2000 investigator who will take personal responsibility for making sure the data are loaded.
- The dataset must be immediately useful to several Sequoia 2000 investigators.
- Everything else being equal, the dataset should "stretch" the schema in directions that would otherwise remain undeveloped (e.g., new types).

Based on the above criteria, plus the desire to limit the initial loading to a manageable number of datasets, the following datasets were selected as having top priority for Phase 1:

- AVHRR;
- global coastal outline
- some point meteorological dataset;
- UCLA GCM output.

4.2 GCM Integration in Sequoia 2000

A collaborative project between scientists in Sequoia 2000 and other researchers aims to develop a first-generation Earth System Model (ESM) encompassing the coupled global atmosphere and ocean systems, including chemical tracers that are found in, and may be exchanged between, the atmosphere and the oceans. This will be the first attempt we know of to extend a General Circulation Model (GCM) with a chemical tracer model, and it will be used to study major current problems in climate, climate change, and climate/chemistry interactions. These problems include the general circulation of the coupled atmosphere/ocean system, and the global biogeochemical cycle of carbon.

One goal of this ESM is to have a modular structure suitable for deployment on massively parallel computer environments and workstation farms. Specifically, we are planning deployment on a Thinking Machines CM-5 system at Berkeley as well as a collection of loosely coupled DEC Alpha systems at SDSC.

A second goal is to couple this model to the Sequoia 2000 DBMS, and this is the objective of the Big Lift, mentioned earlier. Since the UCLA ESM will be running on parallel hardware, the Big Lift will have to exploit parallelism to keep up with the expected data rate. Specifically, lightweight protocols have to be developed to support data entry at low CPU cost, and solutions have to be found for synchronization and consistency problems that arise with parallel data entry into a DBMS.

A third goal of this project is to couple the ESM to a visualization system [Spa93, Mech93]. Model output in AVS can be browsed through the AVS-POSTGRES bridge described earlier, and this capability will provide "after the fact" visualization facilities. In addition, the Tioga system will be used by UCLA researchers when it becomes available. More aggressively, the UCLA group wishes to obtain real time visualization of ESM model output so they can apply "computational steering" to a running model. To achieve this goal, the ESM must be directly interfaced to a visualization system. How to accomplish this task is currently in the investigation stage.

4.3 Remote Sensing Applications

By "remote sensing application," we mean the interpretation of remotely sensed data from aircraft or satellite to provide some geophysical or biological information. Because the input, output, and intermediate steps of the application produce images, visualization and recipe management are intimately tied to remote sensing applications. Typically, these applications involve image processing–of a single image or a combination of images–combined with surface or atmospheric measurements and models of the processes. They differ from traditional image processing analyses [Shap92] in that the image data represent geophysical units, for example the radiance received by the satellite above the atmosphere in a specific spectral band. Moreover, typical requirements of precision in the data for their use in interpretation of geophysical information, i.e. the number of bits per pixel, are much greater than needed for visual examination. Remote sensing applications drive the Sequoia 2000 architecture in several ways. The images are large (a Landsat Thematic Mapper frame is about 300 Mbytes), accessed from remote archives over networks, and produced in a plethora of different formats. Analysis requires quick perusal of multiple images (browsing through abstracts) followed by intensive calculations over large data sets.

Four archetypical applications will be developed during 1993 using the Sequoia 2000 architecture, covering applications in snow hydrology and hydrochemistry, ocean productivity, terrestrial ecology, and the Earth's radiation balance.

1. Analysis of snow properties using data from two satellite sensors—the

Landsat Thematic Mapper (TM) and NOAA's Advanced Very High Resolution Radiometer (AVHRR)—and two aircraft sensors—Airborne Visible and Infrared Imaging Spectrometer (AVIRIS) and the NASA/JPL Airborne Synthetic Aperture Radar (AIRSAR). Together these data sets enable estimation of different snow properties (coverage, albedo, grain size, liquid water content, depth and density) at different spatial and temporal scales [Doz81, Doz89, Nol92, Shi91].

2. Analysis of oceanic productivity using data from the Coastal Zone Color Scanner (CZCS) [Hov78], which operated through 1986, and the Sea-viewing Wide-Field-of-View Sensor (SeaWiFS), scheduled for launch in mid-1993. These sensors measure "ocean color," from which ocean chlorophyll concentration and thereby phytoplankton concentration are estimated [NASA92]. At-sea measurements and models use these estimates to estimate the biological productivity and carbon dioxide uptake of the ocean [Itur89].

3. Analysis of terrestrial vegetation using data from the Landsat TM and AIRSAR. These investigations require correction for the topographic influence, hence the data must be co-registered to digital elevation models, and the processing algorithms for the TM data are different in the shadowed areas. Moreover the interpretation of the data and the sampling scheme to correlate the vegetation with surface energy exchange requires information about the topography [Dav92].

4. Analysis of the Earth's radiation budget and estimation of the surface radiation from geostationary satellite data. Estimation of the surface radiation budget is needed for studies of the land- and ocean-surface climatology and their relationship with productivity. Data from the Geostationary Operational Environmental Satellites (GOES) are used to examine cloud cover, and radiative transfer models use these data to calculate the radiation balance at the Earth's surface [Gaut80, Frou88]. The data processing requires examination of long time series of co-registered images.

4.4 Department of Water Resources Use of Sequoia 2000

DWR's photo laboratory has a collection of about 500,000 slides, which only exist in photographic hardcopy, and the indexing consists of text information in a PC database. DWR's goal is to make many of these slides publicly available electronically. DWR will digitize 250,000 of these slides and store them in Bigfoot, using Sequoia 2000 browsing and indexing tools to allow internal and external users to locate easily images of interest.

DWR also has multiple publications, many indexed in the UC electronic library system MELVYL. Full text of a small subset of these documents exists electronically on Bigfoot. DWR will investigate the possibility of scanning in additional documents, and Sequoia 2000 researchers will explore strategies to enable full-text retrieval of DWR documents through on-line library cataloging systems.

4.5 Interdisciplinary Climate Change Studies in Sequoia 2000

As a contribution to Sequoia 2000, the Climate Research Division at Scripps Institution of Oceanography is developing a Data Information System for Interdisciplinary Climate Change Studies (ICCS-DIS). This is a demonstration pilot project establishing a paradigm for conducting interdisciplinary climate change studies. The three scientific objectives for the ICCS-DIS are:

1. describe how the hydrosphere, atmosphere, land, and cryosphere of planet Earth interact on interannual and decadal time scales;
2. describe the global character of the El Niño/Southern Oscillation (ENSO) phenomenon;
3. seek an understanding of how the physics, chemistry, and biology of the ocean, atmosphere, land and cryosphere interact to bring about climate change.

Our goal is to allow scientists to do the following using an intuitive Graphical User Interface:

1. browse and select interdisciplinary data sets on POSTGRES using GIS technology;
2. Access and subset global and regional interdisciplinary data resident on POSTGRES;
3. register and visualize these interdisciplinary data together;
4. conduct multivariate analyses on these interdisciplinary data (e.g., compute Fourier spectra and complex empirical orthogonal functions).

The following interdisciplinary data sets will be loaded into POSTGRES:

1. GRIDDED FIELDS
   (a) Subsurface ocean temperature (1979-present)
   (b) Altimetric sea level (1985-1990)
   (c) Global meteorological data (1980-present)
   (d) Sea surface temperature (1980-present)
   (e) Air-sea fluxes (1980-present)
   (f) Long and short wave radiation (1984-present)
   (g) Sea ice (Arctic and Antarctic (1966-present)
   (h) Snow depth (1966-present)
   (i) Global vegetation index (1982-present)
2. IMAGES
   (a) AVHRR GAC SST for 1984-present
   (b) ISCCP radiation data (1984-1990)
   (c) GEOSAT altimetry (1985-1990)
   (d) ERS-1 and TOPEX altimetry (1992-present)
3. TIME SERIES
   (a) Global meteorological station data (1900-present)
   (b) Global hydrological station data (1900-present)
   (c) NDBC buoy data (1980-present)
   (d) Solar Irradiance (1978-present)
   (e) Global ecosystems and vegetation time series (1980-present)
4. OBSERVATIONS
   (a) Surface and subsurface ocean temperature (1979-present)
   (b) Hydrographic physical and chemical data (1890-present)
   (c) COADS data set (1890-present)
   (d) Radiosonde observations (1950-present)

These selected data sets will be made available to Sequoia 2000 scientists, accompanied by visualization and analysis tools with which to conduct multivariate analyses. We expect that these analyses will tell us something new about the way in which the ocean, atmosphere, land, and ice interact to bring about interannual climate and global change.

5 Longer-Term Efforts

Phase 1 of the Sequoia 2000 project started in July 1991 and will end in June 1994. We hope to continue with a second phase of Sequoia 2000 that will start in July 1994. The following sections show some of our efforts that will come to fruition only in Phase 2. These include an schema transfer protocol, a hardware storage manager, a distributed file system and a distributed DBMS.

5.1 Transfer Protocol

The Sequoia 2000 schema contains a mechanism to store any Sequoia 2000 data. Specifically, the metadata that describe how any data element is to be interpreted is stored as additional data in other classes inside the DBMS. As long as users submit queries to obtain relevant data, they can inspect the metadata to decide how to operate on or search for desired information. However, suppose a client wants to move data from one machine to another, say to run them through a program that resides on a supercomputer. There must be a way to transfer the metadata along with the data, so that complete information is available at the remote site. This function requires an **schema transfer** protocol, and we are working on the definition of this protocol [Ahme93].

5.2 Storage Controller

The Berkeley hardware group has pioneered the development of Redundant Arrays of Inexpensive Disks (RAID) [Katz89, Katz91a]. RAID requires a sophisticated I/O controller be placed between the CPU and the collection of disk devices. This I/O controller must keep the redundant parity information up to date and map logical blocks to physical locations on the media.

The same group is now focused on the possible construction of a better I/O controller that might control data migration between secondary and tertiary storage as well as play a part in any end-to-end compression scheme [Katz91b]. A last area of possible research is the design of a backup scheme for tertiary storage. It is impossible to take a dump of a 10-terabyte storage system. At 1 Mbyte/sec, 10^7 seconds, about 4 months, would be needed. Obviously a new idea for data reliability is required.

5.3 Shasta

The Sequoia 2000 clients are adamant on the issue of distribution. They expect that their data will be remotely stored on multiple Sequoia 2000 systems. However, they expect frequently used data to be cached locally on the disk of their client machine or on a local server in their immediate vicinity. In addition, the clients do not want to know the name or location of the Sequoia 2000 server where their data are stored. Similarly, any data redundancy through multiple copies of objects should be likewise transparent. Lastly, since their files are so gigantic, they wish a file system be able to store part of a file at one location and the remainder at another location. In short, they want a **distributed** file system, that supports location transparency. Several file systems have been designed that begin to serve this need. The most robust is arguably the Andrew File System [Harr91], developed at CMU. The improvements that we expect to make to the Andrew design are [And92]:

- optimizing for network bandwidth instead of server CPU load;
- caching of file blocks, instead of caching whole files;
- the ability to disable caching, when data being fetched are too large to fit in local cache;
- "write back" cache coherence, so that when temporary files are created they are not immediately sent over the wide area network;
- data structures designed to scale to terabytes of local cache and millions of cached files;
- application control (when needed) over the file system's caching and migration policies.

We are embarking on a prototype effort in this direction, known internally as The Sequoia 2000 File System (TSFS). In keeping with the project goal of naming all software systems after California places, it is called **Shasta.**

5.4 Mariposa

A second approach to distribution in Sequoia 2000 is a distributed database effort called **Mariposa,** because Sequoia 2000 data must be distributed at multiple sites connected by a WAN. Unlike a distributed file system that moves data on demand from one or more remote sites to the user's program as needed, a distributed database system has the option of moving the user's query to the data or moving the data to the query, whichever is thought to be more efficient.

Unlike previous distributed DBMSs, which have assumed that data are statically partitioned among the sites in a computer network, Mariposa will assume that data will freely migrate among sites, and that data placement is a dynamic optimization issue. Lastly, Mariposa will attempt to make placement decisions by constructing a rule engine that will interpret a rule base. In this way, it is easy for a user to freely change the behavior of the system by changing a few rules. Mariposa is at its initial design stage [Ston93c].

6 Implementation Plan

This section contains a collection of milestones that we expect to achieve during the remainder of Phase 1 of the project. They are ordered by layer in the architecture. Some tasks are in a "critical path" and will seriously delay other tasks if not completed on schedule. Associated with each subsection are the names of each Sequoia 2000 investigator or staff person responsible for the deadlines in that section.

6.1 Architecture Layers

6.1.1 Footprint (Tom Anderson)

- 01 May: Beta fpserv (client-server version of Footprint) available on Metrum, Exabyte, and HP jukebox devices.

6.1.2 File Systems

Highlight (Carl Staelin):

- 01 Jul: Beta NFS access to Highlight on Metrum and/or Exabyte.

Inversion (Mike Olson):

- 01 Mar: Beta NFS access to Inversion file system for Sequoia 2000 clients. Supported hardware: magnetic disk, Sony WORM jukebox.
- 01 Apr: Initial implementation of Metrum VHS tape jukebox storage manager in POSTGRES.
- 01 May: Beta NFS access to Inversion on Metrum.

UniTree (Reagan Moore):

- 15 Jan: Modify POSTGRES storage manager to access UniTree via LibTree read/write routines.
- 01 Mar: Install pre-beta release of NSL UniTree on RS6000 platform.
- 25 Mar: Install POSTGRES on DECstation 5000 under ULTRIX version 4.3.
- 15 Apr: Install beta release of NSL UniTree on RS6000 platform.
- 01 May: Install NSL UniTree LibNSL library for supporting read/write routines on DECstation 5000.
- 15 May: Test compatibility of POSTGRES with NSL UniTree multitasking. If POSTGRES is unable to interact correctly with NSL UniTree multitasking, a NFS interface will be used.
- 01 Jun: POSTGRES-NSL UniTree ready for user data archived on disk.
- 15 Sep: POSTGRES-NSL UniTree ready for user data archived on STK robot silo.

EpochServ (Jon Forrest):

- 01 Mar: Epoch EpochServ up and ready for testing.
- 08 Mar: Data from Ninja copied to EpochServ.
- 15 Mar: EpochServ ready for general use.

This schedule assumes that:

- Our existing Sparcstation 1 can be upgraded by 15 Mar.
- The new 100 Gbyte HP jukebox can be successfully attached to the Sun.

6.1.3 DBMS (Mike Stonebraker)

- 03 Mar: POSTGRES version 4.1 released; supports:
  - security
  - authentication by Kerberos

  - untrusted functions
  - NFS interface for Inversion large objects

- 15 Sep: POSTGRES version 5.0 released; supports:
  - multikey indices
  - sets
  - clustering
  - tertiary memory (better/faster than current version)
  - hashed access method(s)
  - chunked arrays
  - automatic type-coercion of constants

6.1.4 Applications

AVS and IDL (Terry Figel and Peter Kochevar):

- 01 Mar: Send out notice to Sequoia 2000 IDL and AVS users about this development plan.
- 01 Apr: Organize a meeting to see which tools need to be developed.
- 01 May: Develop tools within the systems.
- 01 Jun: Organize a meeting to discuss shortcomings of tools being written Discuss possible solutions.
- 01 Jul: Begin gathering all routines written by Sequoia 2000 investigators.
- 01 Aug: Assemble all AVS and AVS routines for package at retreat.

Tioga (Mike Stonebraker and Peter Kochevar):

- 01 Apr: Front-end <-> back-end protocol design completed.
- 15 Apr: Visualization architecture document completed.
- 15 Aug: Version 0 front-end completed
- 15 Dec: Tioga version 1.0 released; includes
  - recipe editor

  - recipe storage system
  - (prototype) recipe executor
  - version 1 front-end

The Tioga front-end consists of three parts: the Intelligent Visualization Subsystem, the Display Subsystem, and the Visualization Executive. Both version 0 and version 1 of the front-end will be fully functional in that they will do visualization planning and data browsing albeit at different levels of capability. Version 1 will have a larger ingredient set, a more elaborate rule set, and a richer set of task operators than version 0.

The major tasks that must be completed for the front-end are:

- Intelligent Visualization Subsystem
  - Visualization Planner
    - Develop/acquire a knowledge management system containing a rule-based reasoning component
    - Develop a rule set for combining ingredients into recipe snippets that expand "eye" boxes
  - Knowledge Base
    - Ingredient knowledge
      - Settle on a core set of ingredients that do conversion from database objects into renderable forms (short term AVS, long-term ?)
      - Settle on a mechanism that describes ingredients (inputs, outputs, and functionality)
      - Settle on a recipe scripting language
    - Task knowledge
      - Settle on a core set of visualization task operators
      - Develop a simple tasking language capable of combining task operators into task specifications
      - Develop a task editor to create and alter task specifications
    - Data knowledge
      - Develop a uniform data representation based on the abstract notion of fiber bundles
    - Domain knowledge
      - Settle on a core list of Earth science vocabulary terms

- Display Subsystem
  - Develop a schema-independent visual database browser
    - Settle on a representation for interactive renderable forms
    - Develop a small set of 3-D widgets to augment a 2-D graphical user-interface
- Visualization Executive
  - Build a central controller for the Tioga visualization management system (The controller handles communication with the Tioga back-end and fires up the Intelligent Visualization and Display Subsystems when they are needed)

Lassen (Ray Larson):

- 15 Feb: Demo version available to DARPA project members for testing. (Uses preliminary version of schema and database, with approximately 40 full-text CS technical reports in PostScript form. Uses version 4.0.1 POSTGRES.
- 01 Mar: Conversion of interface to version 4.1 POSTGRES completed. Support for page image browsing added to interface (i.e., general image display and browsing support).
- 15 Mar: Conversion of demo data to conform to "official" Sequoia 2000 text document schema completed. Begin conversion for all CS technical reports. Conversion of basic keyword indexing for text and large objects to version 4.1 of POSTGRES completed.
- 01 Apr: Rules system support for keyword indexing. Preliminary Z39.50 support in Lassen interface. Start development of keyword indexing access method for POSTGRES.
- 01 May: Full CS Technical Report Database loaded. Preliminary DWR report database loaded. First "Distribution Version" of Lassen interface and indexing software. Sequoia 2000 tech reports on Lassen interface and keyword indexing.
- 01 Jun: Z39.50 support in Lassen interface and POSTGRES backend. Noun Phrase support. Sequoia 2000 tech reports on both.

- 01 Aug: Keyword Access Method Support working. Geographic Name extraction and georeferencing support. Tech reports on both.

The Big Lift (Keith Sklower): The Big Lift will be implemented as a network daemon that will accept GCM connections using the NCSA DTM protocol. Incoming GCM data will written into POSTGRES using a modified POSTGRES "copy" command that will allow specification of array indices.

Implementation schedule:

- 15 May: prototype running

This schedule assumes the availability of:

- large ("chunked") arrays in POSTGRES
- source code for the SDSC AVSGCM bridge (uses the NCSA DTM protocol)

Hollywood (Larry Rowe):

- 15 Aug: Demonstrate desktop videoconferencing system on Sequoia 2000 network.
- 15 Sep: Complete video playback experiments using RTIP protocol on Sequoia 2000 network.
- 15 Oct: Demonstrate video playback on DECstations using Jvideo boards.
- 15 Nov: Demonstrate distributed presentation system coupled with PictureTel video conferencing system.
- 15 Dec: Demonstrate video database browser accessing video-on-demand server.
- 15 Mar 1994: Demonstrate PictureTel interface to desktop video conferencing system.

6.1.5 Network (Joe Pasquale)

- 01 Apr: begin testing of S2Knet routers with T3 boards by Dave Boggs
- 01 Jul: complete upgrade of S2Knet backbone to T3

6.2 Multi-Layer Components

6.2.1 Guaranteed Delivery (Domenico Ferrari and Fred Templin)

We are porting the protocols in the Tenet real-time protocols suite to S2Knet. Once the port is completed, we plan to experiment with the protocols using a variety of types of traffic; in particular, interactive traffic involving transmission of large images, including multiple sequences of images to the same remote workstation. When the correctness and performance of the protocols will have been verified, we plan to release them for use alongside IP, TCP, and UDP by Sequoia 2000 scientists.

The prototype Tenet suite, which is the one to be ported to S2Knet, consists of four protocols, three for data delivery:

- RTIP (the Real-Time Internetworking Protocol)
- RMTP (the Real-Time Message Transport Protocol)
- CMTP (the Continuous Media Transport Protocol)

and one for control (establishment, teardown) of real time channels:

- RCAP (the Real-Time Channel Administration Protocol)

The client of the real-time service implemented by the Tenet suite will specify when calling RCAP the following traffic characteristics:

- the minimum interpacket interval
- the average interpacket interval
- the averaging interval
- the maximum-packet size

and the following performance requirements:

- the maximum end-to-end delay,
- the probability that a packet satisfies the delay bound,
- the delay jitter bound (optional),
- the probability that a packet is lost due to buffer overflow.

Implementation schedule:

- 15 Mar: port of RMTP and RTIP to T1 S2Knet completed.
- 31 Mar: port of RCAP to T1 S2Knet completed.
- 30 Apr: port of CMTP to T1 S2Knet completed; testing of, and initial experiments with, RMTP, RTIP, and RCAP completed.
- 31 Jul: testing of, and initial experiments with, CMTP completed.
- 31 Oct: port of protocols to T3 S2Knet completed.
- 31 Dec: testing of, and experimentation with, the protocols on T3 S2Knet completed; Tenet suite released for general use.

6.2.2 Abstracts (Joel Fine)

- 15 May: The first step in supporting abstracts is a modified kernel which records all accesses to data so that we can discover opportunities for using abstracts. We expect the kernel to be in use by this date.
- 01 Aug: Beta "transparent make" facility available.
- 15 Aug: After recording usages, we intend to release a prototype system to replace accesses to large objects with abstracts. This relies on a file system being available for large objects.

6.2.3 Compression (George Polyzos)

We are working on using hierarchical coding for providing better performance or guaranteeing timely delivery (when combined with the RTIP protocols) to a subset of the signal, and therefore enabling the network to (potentially) support more "users."

In addition, mainly for still images, this scheme can mask some of the latency of transmission by progressively painting the whole image from coarser to finer resolution (also known as pyramiding).

- 15 Aug: Prototype for progressive retrieval of hierarchically coded images.
- 15 Oct: Support for efficient transmission of hierarchically coded video (or animations).

6.2.4 Integrating Existing Software (Bill Weibel)

For Phase I, the work of this committee is based on the assumption that scientists will retrieve data from Bigfoot in the form of files. "External" files produced by Sequoia 2000 software should be readable by applications currently in use by Sequoia 2000 scientists. The formats will contain sufficient metadata, and be supported by application-specific functions, such that the more mundane tasks associated with importing data into an application, such as determinating array sizes, can be taken out of the user's hands and left entirely to the software.

- 22 Feb: List of commonly used applications is compiled from survey of Sequoia 2000 community. Highest priority items are selected. Common file formats are identified.
- 01 Mar: External file formats to be supported for Phase I are selected. Assignments of responsibility for formats are distributed among Sequoia 2000 community.
- 15 Mar: Baseline application-specific function support for file formats is outlined.
- 15 Apr: Prototype implementations of supported formats are designed, based on the data dictionary described in 4.1.
- 01 May: Functions written for data conversion between Sequoia 2000 schema and the supported formats.
- 01 Jun: Baseline application-specific function support is completed.

6.3 Using Sequoia 2000

6.3.1 Schema Construction and Data Loading (Jim Frew)

- 15 Feb: scenarios identified that will use the priority datasets.
- 01 Mar: glossary completed.
- 01 Apr: data dictionary completed.
- 15 Apr: priority datasets loaded.
- 01 May: function support for data dictionary completed.
- 15 May: priority datasets converted as needed to support schema.

This schedule is designed to have the priority datasets available under the schema for 3 months prior to the August retreat.

6.3.2 GCM Integration (Roberto Mechoso)

- 15 Feb: Eight year-long simulation with UCLA atmospheric GCM completed.
- 01 Mar: Twenty-five year-long simulation with UCLA coupled atmosphere-ocean GCM completed.
- 15 Apr: Data from both GCM simulations installed on Bigfoot.
- 01 May: Report completed on GCM capture by AVS.
- 01 Sep: Preliminary tests of atmospheric GCM running in CRAY Y-MP (SDSC, JPL) coupled to oceanic GCM running on Intel Paragon (Caltech, SDSC) displaying output and loading data onto Bigfoot in real time.
- 01 Nov: Preliminary tests of coupled atmosphere-ocean GCM running on a "farm" of DEC scientific workstations.

6.3.3 Remote Sensing Applications

UCSB will develop remote sensing scenarios, in the following order and schedule:

Snow Properties (Jeff Dozier):

- 15 Feb: scenario for snow mapping and classification from the Landsat Thematic Mapper completed.
- 01 Mar: snow mapping scenario installed in Bigfoot; Sequoia 2000 technical report published, based on [Doz93].
- 15 Apr: scenario for estimation of snow grain size from AVIRIS data completed and installed in Bigfoot.
- 15 Apr: scenario for snow cover mapping from AIRSAR data completed and installed in Bigfoot.
- 15 Jun: historical sequence of snow cover in Sierra Nevada for all our available Landsat data completed using Sequoia 2000 environment; Sequoia 2000 technical report published.

- 15 Aug: analysis of snow cover for all our available AVIRIS data completed; Sequoia 2000 technical report published.

- 15 Sep: analysis of snow cover for all our available AIRSAR data completed; Sequoia 2000 technical report published.

- 15 Oct: distributed energy-balance model of snow cover developed using Sequoia 2000 environment; Sequoia 2000 technical report published.

Ocean Radiant Heating (Dave Siegel and Catherine Gautier):

- 15 Feb: complete compositing of monthly level 3 CZCS chlorophyll imagery into a climatology on a 1 degree by 1 degree grid.

- 1 Mar: sample the ISCCP solar radiation data from Catherine Gautier's group onto a 1 degree by 1 degree grid. This conform to the Levitus seawater climatology which will be used to determine global mixed layer depth.

- 15 Mar: use our hybrid bio-optical model for mixed layer radiant heating rates and sub-surface solar fluxes.

- 15 Apr: Present poster of this work at the 3rd annual The Oceanography Society (TOS) meeting in Seattle.

- 15 Sep: Analysis of CZCS data completed; Sequoia 2000 technical report published.

Global Primary Production Estimation (Ray Smith and Catherine Gautier):

- 1 Mar: complete development of monthly and seasonal chlorophyll imagery from level 3 CZCS data on a 20km x 20km grid.

- 1 Apr: finish chlorophyll converting data into IDL format.

- 1 June: complete the development of solar radiation model to calculate daily rates of photosynthetically available radiation from net shortwave data. These estimates will include an accounting of the effects of aerosols.

- 15 Sep: Estimate net primary production using the model of Morel (1991); Sequoia 2000 technical report published.

Terrestrial Vegetation (Frank Davis):

- 15 Feb: scenario for fire detection and mapping from AVHRR LAC composites completed.
- 15 Apr: scenario for vegetation mapping and monitoring from TM data, digital topographic data and existing vegetation maps completed.
- 15 May: statewide composite of 1990 TM data for California completed and installed in Bigfoot.
- 15 May: Daily AVHRR LAC data covering western U.S. for 5/90 through 10/90 compiled and installed in Bigfoot.
- 15 Sep: Analysis of AVHRR LAC data for land cover mapping, fire detection and fire mapping completed; Sequoia 2000 technical report published.
- 15 Dec: TM mapping of California vegetation completed. Sequoia 2000 technical report published.

Earth Radiation Budget (Catherine Gautier):

- 01 Mar: Install input data for ARM processing (GOES vis 1km over Oklahoma) in POSTGRES. Set up scripts to place hourly values into POSTGRES database in real time.
- 04 Mar: Present initial results at ARM conference.
- 01 Apr: Generate SW, PAR, UVA and UVB hourly, saving them in POSTGRES.
- 01 May: Sequoia 2000 technical report on processing (user algorithms, IDL, Tcl, POSTGRES, and AVS.)
- 15 May: Sequoia 2000 technical report on algorithms.
- 15 May: Presentation at AGU.
- 01 Jun: Begin changes required to produce the above values on a global levels.
- 01 Jul: Install global model, begin running
- 15 Jul: Sequoia 2000 technical report describing global processing
- 15 Aug: Presentation of results at retreat

6.3.4 DWR Applications (Gary Darling)

- 01 Apr: first delivery to UCB of photo CD-ROM containing digitized slides from DWR library. Will deliver 1000 slides per week thereafter.
- 15 Jun: submit plan for loading additional DWR text documents into Bigfoot.

6.3.5 Interdisciplinary Climate Change Studies at SIO (Warren White, Norm Hall, Dan Cayan, John Roads, Tim Barnett, Richard Somerville)

- Mar: Adapt the AVS- POSTGRES bridge to browse, clip, and extract data from ICCS data sets to be installed on POSTGRES.
- Apr: Develop a GUI in IDL that will allow the mean, variance, and residuals to be computed for each ICCS data set.
- May: Develop a GUI in IDL that will allow ICCS data to be co-registered onto a common grid in space-time.
- Jun: Develop a GUI in IDL that will allow up to 12 different time series to be cross-correlated, with multi-variate CEOF between the 12 series computed and displayed.
- Jul: Begin loading ICCS data sets into POSTGRES.
- Aug: Give an example of the ICCS-DIS at the Sequoia 2000 retreat using a subset of the selected data presented in Section 4.5.

6.4 Long-Term Efforts

6.4.1 Data Transfer Protocol (Zahid Ahmed)

The data transfer protocol will be based on the Data Interpretation Language. The DIL will be used to specify the export schemas of data objects based on an extendible common scientific and geometric data model.

- 01 Mar: [Ahme93] completed.
- 15 Mar: Devise a scientific and geometric data model for Earth science application; design an extendible language specification for the DIL.
- 30 Mar: Discuss DIL implementation approach with Sequoia 2000 DBMS group, NCSA's HDF group, and possibly with NASA GSFC's CDF group.

- 15 Apr: DIL funding proposal (along with Mike Folk, NCSA) to NASA, and possibly to two NSF divisions submitted; distribute proposal to some Sequoia 2000 members.
- 15 Jun: Complete feasibility study on the addition of an inference-based scientific and visualization terminological reasoner to the DIL design; publish study as Sequoia 2000 technical report.
- 01 Jul: Negotiate agreement with Stonebraker on schema mappings between on-the-wire schema and POSTGRESbased local schema; publish agreement as Sequoia 2000 technical report.
- 15 Oct: Complete prototype schema translator between DILbased on-the-wire schema and POSTGRESbased local schema. Coordinate completion of incorporation efforts of the DILbased export schema into EOSDIS Level 0 HDF datasets at certain NCSAEOSDIS DAAC sites.
- 01 Nov: Complete investigation of extending the DILś data model for including data lineage information; publish findings as Sequoia 2000 technical report.
- 15 Nov: Complete testing of DILbased Data Transfer Protocol with Sequoia 2000 POSTGRESbased local schema. Obtain performance evaluation of DILbased HDF datasets at EOSDIS DAAC sites from Mike Folk, NCSA.
- 01 Dec: Publish a Sequoia 2000 technical report: "Performance evaluation of the DIL used for Data Transfer Protocol in a Heterogeneous Information Sources Environment".

6.4.2 Backup for Tertiary Storage (Dave Patterson)

- 15 Jun: We will characterize alternative Robot Archival Tape Libraries (RATLs) to determine cost, performance, reliability of the various options. This will include recommendations of technologies that appear key to the future.
- 15 Aug: We will describe a backup/disaster/reliability scheme for RATLs. We will devise a scheme that economically solves the following problems, which are related but treated independently:
  - Site Disaster Recovery
  - Data availability despite hardware failure

- Daily backups so that can restore in case of failures
- Migration of old tapes into new technologies to avoid ending up with unreadable tapes due to disuse or advances
- Higher error rate of tapes vs. disks

6.4.3 Shasta (Tom Anderson)

- 01 Aug: prototype running

6.4.4 Mariposa (Mike Stonebraker)

- 01 Jun: architectural design document
- 01 Jul: implementation plan
- 31 Dec: prototype running

7 Conclusion

The Sequoia 2000 project plans an initial software distribution consisting of Footprint, Highlight, Inversion, POSTGRES, the AVS-POSTGRES bridge, the Big Lift, Lassen, and perhaps an early version of Tioga during 1993. While this software distribution is in preparation, Sequoia Global Change investigators will use the prototype tools for analysis of Earth science data and models, in innovative ways that would have been difficult without the Sequoia 2000 environment.

Acknowledgements

This research was sponsored by the Digital Equipment Corporation's External Research Program, and by the University of California. Government sponsors include the California Department of Water Resources, Coordinated Environment Research Laboratory, National Aeronautics and Space Administration, and the U.S. Geological Survey. Industrial participants include Epoch Systems Inc., Hewlett-Packard, Hughes, MCI, Metrum Corp., PictureTel Corp., Research Systems Inc., Science Applications International Corp., Siemens, and TRW.

References

[Ahme93] Z. Ahmed. Data Interpretation Language: Definition, Purpose, and Implementation Strategy. Sequoia 2000 Report 93/ ??? , University of California, Berkeley, ??? 1993.

[And92] T. Anderson and R. Wang. Design notes of a distributed tertiary file system. [UNPUBLISHED: NEED A REAL CITATION], December 1992.

[Beck84] R. A. Becker and J. M. Chambers. *S: An Interactive Environment for Data Analysis and Graphics*. Wadsworth, Monterey, CA, 1984.

[Chen91] J. Chen, R. Larsen, and M. Stonebraker. The Sequoia 2000 Object Browser. Sequoia 2000 Report 91/4, University of California, Berkeley, December 1991.

[Dav92] F. W. Davis, D. S. Schimel, M. A. Friedl, J. C. Michaelsen, T. Kittel, R. Dubayah, and J. Dozier. Covariance of biophysical data with digital topographic and land-use maps over the FIFE site. *Journal of Geophysical Research*, 97(D17):19,009–19,021, 1992.

[Dix91] S. S. Dixit and Y. Feng. Hierarchical address vector quantization for image coding. *CVGIP–Graphical Models and Image Processing*, 53(1):63–70, January 1991.

[Doz81] J. Dozier, S. R. Schneider, and D. F. McGinnis, Jr. Effect of grain size and snowpack water equivalence on visible and near-infrared satellite observations of snow. *Water Resources Research*, 17(4):1213–1221, 1981.

[Doz89] J. Dozier. Spectral signature of alpine snow cover from the Landsat Thematic Mapper. *Remote Sensing of Environment*, 28:9–22, 1989.

[Doz92] J. Dozier. Opportunities to improve hydrologic data. *Reviews of Geophysics*, 30(4):315–331, November 1992.

[Doz93] J. Dozier and C. W. Rosenthal. Sequoia 2000 end-to-end processing scenario: Automated snow mapping from Landsat Thematic Mapper data. [UNPUBLISHED: NEED A REAL CITATION], January 1993.

[Epoc92] Epoch Systems, Inc. *EpochServ Technical Summary*. Westborough, MA, 1992.

[Evan91] D. A. Evans, S. K. Handerson, R. G. Lefferts, and I. A. Monarch. A Summary of the CLARIT Project. Laboratory for Computational Linguistics Report CMU-LCL-91-2, Carnegie Mellon University, Pittsburgh, PA, November 1991.

[Ferr90] D. Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine*, 28(11), 1990.

[Fine92] J. A. Fine. Abstracts: A Latency-Hiding Technique for High-Capacity Mass-Storage Systems. Sequoia 2000 Report 92/11, University of California, Berkeley, June 1992.

[Frou88] R. Frouin, C. Gautier, and J. Morcrette. Downward longwave irradiance at the ocean surface from satellite data: methodology and in situ validation. *Journal of Geophysical Research*, 93(C1):597–619, 1988.

[GA91] General Atomics/DISCOS Division. *The UniTree Virtual Disk System*. San Diego, 1991.

[Gaut80] C. Gautier, G. Diak, and S. Masse. A simple physical model to estimate incident solar radiation at the surface from GOES satellite data. *Journal of Applied Meteorology*, 19:1005–1012, 1980.

[Harr91] B. T. Harrison. AFS and wide-area filesystems. *DEC Professional*, 10(12):96, November 1991.

[Her80] C. Herot. SDMS: A spatial data base system. *ACM Transactions on Database Systems*, 1980.

[Hos90] J. M. Hosinski. Lab markets its storage system for supercomputers. *Government Computer News*, 9(9):40, April 1990.

[Hov78] W. Hovis. The coastal-zone color scanner (CZCS) experiment. In *Nimbus 7 User's Guide*. NASA Goddard Space Flight Center, Greenbelt, MD, 1978.

[Itur89] R. Iturriaga and D. A. Siegel. Microphotometric characterization of phytoplankton and detrital absorption properties in the Sargasso Sea. *Limnology and Oceanography*, 34:1706–1726, 1989.

[Katz89] R. H. Katz, G. A. Gibson, and D. A. Patterson. Disk system architectures for high performance computing. *Proceedings of the IEEE*, 77(12):1842–1858, December 1989.

[Katz91a] R. H. Katz. High Performance Network and Channel-Based Storage. Sequoia 2000 Report 91/2, University of California, Berkeley, October 1991.

[Katz91b] R. H. Katz, T. Anderson, J. Ousterhout, and D. Patterson. Roboline Storage: Low Latency, High Capacity Storage Systems Over Geographically Distributed Networks. Sequoia 2000 Report 91/3, University of California, Berkeley, October 1991.

[Koch93] P. Kochevar, Z. Ahmed, M. Bailey, J. Shade, and C. Sharp. A Visualization Agenda for the Sequoia 2000 Project. [UNPUBLISHED: NEED A REAL CITATION], January 1993.

[Kohl93] J. T. Kohl, C. Staelin, and M. Stonebraker. Highlight: using a log-structured file system for tertiary storage management. In *USENIX Association Winter 1993 Conference Proceedings*, San Diego, January 1993.

[Lars91] R. R. Larson. Classification, clustering, probabilistic information retrieval and the Online Catalog. *Library Quarterly*, 61(2):133–173, April 1991.

[Lyn91] C. Lynch. SIG LAN and ASIS Standards Committee – The NISO Z39.50 information retrieval protocol: applications and implementation. *Proceedings of the ASIS Annual Meeting*, 28:353, 1991.

[Lyn92] C. A. Lynch. The next generation of public access information retrieval systems for research libraries – lessons from 10 years of the MELVYL system. *Information Technology and Libraries*, 11(4):405–415, December 1992.

[Mech93] C. R. Mechoso, C. C. Ma, J. D. Farrara, J. A. Spahr, and R. W. Moore. Parallelization and distribution of a coupled atmosphere-ocean general circulation model. *Monthly Weather Review*, 1993. (to appear).

[Mill92] E. L. Miller and R. H. Katz. Input/Output Behavior of Supercomputing Applications. In *Proceedings of Supercomputing '91*, pp. 567–576, November 1991.

[NASA92] NASA. Sea-viewing Wide-field-of-view Sensor (SeaWiFS) Global Ocean Primary Production. Research Announcement NRA-92-OSSA-7, NASA, Washington, D.C., 1992.

[Nol92] A. W. Nolin and J. Dozier. Retrieval of snow properties from AVIRIS data. In R. O. Green, editor, *Third Airborne Science Workshop*, JPL Report 91-28, pp. 281–288, Pasadena, CA, 1992.

[NRC92] National Research Council, Computer Science and Telecommunications Board. *Computing the Future: A Broader Agenda for Computer Science and Engineering*. National Academy Press, Washington, D.C., 1992.

[Ols93] M. A. Olson. The design and implementation of the Inversion File System. In *USENIX Association Winter 1993 Conference Proceedings*, San Diego, January 1993.

[Ros92] M. Rosenblum and J. K. Ousterhout. The design and implementation of a log-structured file system. *ACM Transactions on Computer Systems*, 10(1), February 1992.

[Salt71] G. Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[Sara93] S. Sarawagi. Efficient Organization of Large Multidimensional Arrays. **???** 1993.

[Selt90] M. Seltzer and M. Stonebraker. Transaction support in read optimized and write optimized file systems. In *Proceedings 1990 VLDB Conference*, Brisbane, Australia, 1990.

[Selt93] M. Seltzer, K. Bostic, M. K. McKusick, and C. Staelin. An implementation of a Log-structured File System for UNIX. In *USENIX Association Winter 1993 Conference Proceedings*, San Diego, January 1993.

[Shap92] L. Shapiro and A. Rosenfeld, editors. *Computer Vision and Image Processing*. Academic Press, Boston, 1992.

[Shi91] J. Shi, J. Dozier, H. Rott, and R. E. Davis. Snow and glacier mapping in alpine regions with polarimetric SAR. In *Proceedings IGARSS '91*, IEEE No. 91CH2971-0, pp. 2311–3214, 1991.

[Spa93] J. A. Spahr, C. C. Ma, C. R. Mechoso, and M. Stonebraker. GCM captured by AVS. **???** 1993.

[Ston90] M. Stonebraker, L. Rowe, and M. Hirohama. The implementation of POSTGRES. *IEEE Transactions on Knowledge and Data Engineering*, March 1990.

[Ston91] M. Stonebraker and J. Dozier. Sequoia 2000: Large Capacity Object Servers to Support Global Change Research. Sequoia 2000 Report 91/1, University of California, Berkeley, July 1991.

[Ston91b] M. Stonebraker. An Overview of the Sequoia 2000 Project. Sequoia 2000 Report 91/5, University of California, Berkeley, December 1991.

[Ston92a] M. Stonebraker. The integration of rule systems and database systems. *IEEE Transactions on Knowledge and Data Engineering*, 4(5):415–423, October 1992.

[Ston92b] M. Stonebraker, J. Chen, N. Nathan, and C. Paxson. Tioga: Providing Data Management Support for Scientific Visualization Applications. Sequoia 2000 Report 92/20, University of California, Berkeley, December 1992.

[Ston93a] M. Stonebraker and M. A. Olson. Large object support in POSTGRES. In *Proceedings of the 1993 International Conference on Data Engineering*, Vienna, Austria, April 1993. (to appear).

[Ston93b] M. Stonebraker, J. Frew, K. Gardels, and J. Meredith. The Sequoia 2000 Benchmark. In *Proc. 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., May 1993.

[Ston93c] M. Stonebraker, P. Aoki, R. Devine, W. Litwin, and M. A. Olson. Mariposa: a new distributed DBMS. [UNPUBLISHED: NEED A REAL CITATION].

[Zee92] J. Zeeman. The Z39.50 standard – almost a reality. *Canadian Library Journal*, 49(4):273–276, August 1992.