The SEQUOIA 2000 Project *

Michael Stonebraker
EECS Dept., University of California at Berkeley
Berkeley, California

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 flagship research project for the 1990s, succeeding Project Athena. It is an example of the close relationship that must exist between technology and applications to foster the computing environment of the future [6].

There are four categories of investigators participating in SEQUOIA 2000: (a) 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 (SDSC). Their charge is to build a prototype environment that better serves the needs of the clients. (b) Earth science researchers are affiliated with the Department of Geography at UC Santa Barbara, the Atmospheric Science Department at UCLA, 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. (c) 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, and the United States Geological Survey. Their charge is to steer SEQUOIA 2000 research in a direction that is applicable to their problems. (d) Industrial participants include DEC, Epoch, Hewlett-Packard, Hughes, MCI, Metrum Corp., PictureTel Corp., Research Systems Inc., 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 free or discounted computing equipment.

The purpose of this document is to give an overview of SEQUOIA 2000 project directions. For more detailed information, the reader should consult our strategic plan [16]. Section 2 first motivates the computer science objectives of SEQUOIA 2000. Then, Section 3 continues with a discussion of certain specific projects. Section 4 then explores four themes that cross most elements of the SEQUOIA 2000 architecture. Lastly, Section 5 discusses the longer-term agenda for research and prototyping.

*This research was sponsored by Digital Equipment Corporation under Research Grant 1243, DARPA Contract DABT63-92-C-007, NSF Grant RI-91-07455, and ARO Grant DAAL03-91-6-0183.

2 SEQUOIA 2000 Motivation

The SEQUOIA 2000 architecture is motivated by four fundamental computer science objectives, namely big fast storage, an all-embracing DBMS, integrated visualization tools, and high-speed networking. We now discuss these points in turn.

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 would like to store about 100 terabytes of data now. Many of these are common data sets, used by multiple investigators. Unlike some other applications, much of our clients' I/O activity is random access.

Our clients agree on the merits of moving all their data to a database management system. 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.

Our clients use visualization tools such as AVS, IDL, Khoros, and Explorer. They are frustrated by aspects of these products and are anxious for a next-generation visualization toolkit that allows better manipulation of large data sets, provides better interactive data analysis tools, and fully exploits the capabilities of a distributed, heterogeneous computing environment.

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.

3 SEQUOIA 2000 Technical Projects

To address these needs, SEQUOIA 2000 is pursuing six interrelated projects in the areas of massive storage, file systems for a deep store, DBMS, networking, visualization tools and electronic repositories. In this section we briefly discuss these projects.

Our environment is DECstation 5000's for both servers and client machines, moving to Alphas later this year. All clients are connected to FDDI local area networking, and the SEQUOIA 2000 sites are joined by a private T1 (soon to be T3) network. Deep storage is a collection of 6 robotic devices at Berkeley with a current aggregate capacity of 10 Tbytes.

The Storage Project: The focus of the hardware group is on extending RAID ideas [8] to tertiary memory. We are considering striping and redundancy over media in a jukebox, robot arms in a jukebox, whole jukeboxes and even whole systems. Also, we are concerned with the issue of backup and recovery in deep storage. For example, taking a dump of a 10 Tbyte storage system requires several months, and cannot be reasonably contemplated.

The File System Projects: We are building two file systems for deep storage, and plan to run three additional commercial systems. The first file system is **Highlight** [5]. It is an extension of the Log-structured File System (LFS) pioneered for disk devices by Rosenblum and Ousterhout [9]. 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 [11]. 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, 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** [7, 17], 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 [12], 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 [7], a characteristic of the SEQUOIA 2000 workload.

The DBMS Project: 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 [13]. 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 [12, 14].

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 [10] is being prototyped. The R-tree access method in POSTGRES is also being extended to support the full range of SEQUOIA 2000 spatial objects. Moreover, our clients typically use pattern classification functions in POSTQUEL queries that are very expensive to compute. We have been working on the POSTGRES optimizer to deal intelligently with such queries [4].

To focus the attention of the DBMS research community on the needs of our clients, we have designed the SEQUOIA 2000 Storage benchmark and run it on several software platforms [18]. We are also working on an "end-to-end" benchmark, that would include explicit visualization and networking operations.

The Network Project: The networking project uses the SEQUOIA network as a prototype for our ideas. Specifically, we have avoided running "custom iron" as routers, instead believing that Alphas are fast enough to route T3 packets. In addition, we are trying to lower the number of copies of each byte made by the operating system on the way from storage to the network. Furthermore, we are

exploring optimizing multicast protocols, required for successful video teleconferencing by SEQUOIA 2000 participants. Lastly, we are exploring guaranteed delivery protocols that will allow a client to specify an animation sequence which will be delivered to his workstation with a service guarantee. This will allow him to display it smoothly without local buffering. For a description of these algorithms, consult [2].

The Visualization Project: To improve on the limitations of visualization tools such as AVS, and IDL, 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 the user to "navigate" around his data and then zoom in to obtain additional data on items of particular interest. Tioga [15] is a joint project between Berkeley and SDSC, and a prototype "early Tioga" [1] is currently running.

The Repository Project: The final project entails viewing the entire 10 Tbyte storage system as a large electronic library, containing some text but mostly raw satellite data, "cooked" images, simulation output, computer programs, computational sequences ("recipes"), and polygonal data. This project is focused on providing indexing for such objects, and an ability for clients to browse the repository, without knowing exactly what they are looking for. In addition, a natural language understanding query tool is currently under development.

We are also loading a sizeable collection of text, including 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.

4 Common Concerns

Four concerns of SEQUOIA researchers cannot be isolated to a single layer in the architecture; namely guaranteed delivery, abstracts, compression, and integration with other software.

Guaranteed delivery must be an **end-to-end contract**, agreed to by the visualization system (which puts information on the screen), the network (which transports data between machines), the DBMS (which satisfies an underlying query) and the storage system (which retrieves blocks of storage). One approach to this issue is discussed in [15].

Our clients want to **browse** information at low resolution. Then, if something of interest is found, 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** [3], and we are exploring providing them in the visualization package and in the file system.

The SEQUOIA 2000 clients are open to any compression scheme to save storage capacity and network bandwidth, as long as it is lossless. In addition, they are not willing to throw any data away, since its future relevance is unknown. We are exploring the concept of **just in time** decompression. For example, if the storage manager compresses data as they are written and then decompresses them on a read, then the network manager 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. 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.

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, are

easily written.

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, and are embarked on several projects that will come to fruition only in Phase 2. These include an on-the-wire transfer protocol, a hardware storage manager, a distributed file system and a distributed DBMS.

References

[1] J. Chen, et. al., "The SEQUOIA 2000 Object Browser," University of California, Berkeley, SEQUOIA 2000 Technical Report 91/4, December 1991.

[2] D. Ferrari, "Client Requirements for Real-time Communication Services," IEEE Communications Magazine, November 1990.

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

[4] J. Hellerstein and M. Stonebraker, "Predicate Migration: Optimizing Queries with Expensive Predicates," Proc. 1993 ACM-SIGMOD International Conference on Management of Data, Philadelphia, Pa., May 1993.

[5] J. Kohl, et. al., "Highlight: Using a Log-structured File System for Tertiary Storage Management," USENIX Association Winter 1993 Conference Proceedings, San Diego, January 1993.

[6] 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.

[7] M. Olson, "The Design and Implementation of the Inversion File System," USENIX Association Winter 1993 Conference Proceedings, San Diego, CA., January 1993.

[8] D. Patterson, et. al., "RAID: Redundant Arrays of Inexpensive Disks," Proc. 1988 ACM-SIGMOD International Conference on Management of Data, Chicago, Ill, June 1988.

[9] M. Rosenblum and J. Ousterhout, "The Design and Implementation of a Log-structured File System," ACM Transactions on Computer Systems, February 1992.

[10] S. Sarawagi, "Improving Array Access Through Chunking, Reordering, and Replication," (in preparation).

[11] M. Seltzer, et. al., "An Implementation of a Log-structured File System for UNIX," USENIX Association Winter 1993 Conference Proceedings, San Diego, January 1993.

[12] M. Stonebraker, et. al., "The Implementation of POSTGRES," IEEE Transactions on Knowledge and Data Engineering, March 1990.

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

[14] M. Stonebraker and G. Kemnitz, "The POSTGRES Next Generation Database Management System," CACM, October 1991.

[15] M. Stonebraker, "Tioga: Providing Data Management Support for Scientific Visualization Applications," University of California, Berkeley, SEQUOIA 2000 Technical Report 92/20, December 1992.

[16] M. Stonebraker, et. al., "The SEQUOIA 2000 Architecture and Implementation Plan," University of California, Berkeley, SEQUOIA 2000 Technical Report 93/5, March 1993.

[17] M. Stonebraker and M. Olson, "Large Object Support in POSTGRES," Proc. of the 1993 International Conference on Data Engineering, Vienna, Austria, April 1993.

[18] M. Stonebraker, et. al., "The Sequoia 2000 Storage Benchmark," Proc. 1993 ACM-SIGMOD International Conference on Management of Data, Philadelphia, Pa., May 1993.