ATMOSPHERIC AND OCEANOGRAPHIC DATA MODELING WITH A RELATIONAL DATABASE MANAGEMENT SYSTEM

Jim Copeland
Fleet Numerical Oceanography Center
Monterey, CA 93943-5005, USA

1. INTRODUCTION

The Fleet Numerical Oceanography Center (FNOC) has decided to acquire a commercial relational database management system (DBMS) to serve its environmental computer models. This database system is part of a larger development program, the Primary Oceanographic Processing System (POPS) of FNOC and a sister command, the Naval Oceanography Office (NAVOCEANO), Stennis Space Center, Mississippi. Both FNOC and NAVOCEANO are part of the U.S. Navy's Naval Oceanography Command.

POPS is the next generation computer system for processing fleet weather and ocean models at FNOC. A similar system will support research and development at NAVOCEANO, and facilitate data and computer model migration between the sites. The Naval Environmental and Operational Nowcasting System (NEONS), presented elsewhere in this issue of the Proceedings (Tsui 1991), was selected as prototype for the POPS database system.

FNOC started drafting DBMS requirements in October, 1990. Both functional and benchmark requirements were delivered to the prime contractor in October, 1991. We are planning for delivery of the DBMS in 1992. The procurement process is unfolding as this paper is being written; many technical and procedural details are necessarily sensitive and will not be addressed here. Database security is also beyond the scope of this paper; we have special requirements not of general interest to this conference.

2. MOTIVATION

The POPS computer hardware and system software represent a large expenditure. At a time when defense budgets are declining, efficient use of funding is essential, and off-the-shelf solutions which facilitate automation become highly desirable.

2.1 Data Management

Structural changes are driving government procurements toward proven commercial database management systems. The US Government, and the U.S. Navy, support the introduction of formal data administration and automated data management. This means establishing central resources for database technology, data standards, and data sharing, to allow data currently managed by individual programs to become an organizational resource. There are additional requirements to share or exchange data among many military commands, with common logical interfaces and data naming standards.

2.2 Standards

Standards are now essential for cost-effective software. FNOC requires a fully distributed, client-server-model DBMS, which is fully compliant with the portability, open systems architecture, and many standards now required for US Government acquisitions. Particularly important are the American National Standards Institute (ANSI) standards for SQL, and U.S. Military Standards (MILSTD) for TCP/IP communication protocols. The DBMS must also be fully compatible with the UNIX operating system. That makes a very long list of basic requirements, without addressing many other features which are necessary for our applications. The programming cost of providing equivalent capability by designing custom software is prohibitive.

2.3 NEONS Experience

Selecting the NEONS for the POPS database prototype is a successful application of NOARL's research to FNOC's operational requirements. The NEONS, implemented with a commercial DBMS, has the majority of data defined for the largest FNOC production models. This immediately provides a tested software development platform. Furthermore, the NEONS design was tailored to achieve satisfactory performance, which has been validated.

By providing standard FORTRAN and C language interfaces for atmospheric and oceanographic programming, NEONS is a stable user environment, independent of underlying data management software. The system administrator benefits by being able to make DBMS changes without impacting users. In fact, NEONS has been ported to more than one commercial DBMS.

3. PLATFORMS

From a software perspective, the POPS hardware configuration is essentially the same at FNOC and NAVOCEANO. Both systems are designed around two tightly-coupled Cray supercomputers, the larger acting as client and the smaller as server. Each supercomputer is to have its own local disk farm, and the server will also have a channel-attached Automated Archival System (AAS) of near-line mass storage. A high-speed network will connect the supercomputers with several workstations which will act as (or literally be) communication gateways to users. The Cray computers are in place at NAVOCEANO, but not yet at FNOC.

At FNOC the hardware is expected to be a YMP16/8128 and a YMP2E/116 networked via an Ultranet high-speed parallel interface

(HIPPI), and several Sun workstations. An Ethernet local area network (LAN), consisting of many other workstations and microcomputers, will connect to the high-speed network via an Ultranet Hub. The DBMS will be distributed across many of these platforms. Sun workstations on the HIPPI network will be communications gateways for a host of smaller computers which receive data and distribute products.

At NAVOCEANO the POPS hardware is principally a Cray YMP8/8128, a YMP2E/116, two Storage Technology cartridge tape silos and several Sun workstations. Many more workstations and microcomputers may have Ethernet access to the high-speed network via an Ultranet Hub.

Both installations will use UNIX to operate all hardware platforms directly networked with the supercomputers, as well as all platforms to be distributed under the DBMS. The DBMS will support TCP/IP and Sun Microsystems' Network File System (NFS). While the two sites are largely independent, they do exchange significant amounts of data. A T1 circuit connects Sun workstations on each HIPPI network, more than two thousand miles apart. This circuit supports TCP/IP, and will support the distributed DBMS.

4. SUPERCOMPUTING WITH A COMMERCIAL DBMS

The desire to implement a logical data schema independent of the physical base is strong. So is the desire to achieve the programming productivity cost benefit, and the market-driven technical development, of commercial database management systems. Their adaptability to a supercomputing environment is worth examining closely.

4.1 Data and Metadata

An important topic governing the use of a relational DBMS in scientific or engineering computing is the distinction between data and metadata (data about other data). Metadata is essential to data management, while ordinary or primary data is used for computation. Primary data must flow efficiently through the DBMS or at least be efficiently managed by it. This performance is possible if the DBMS is properly designed.

Several designs have emerged. Some DBMSs now incorporate the ability to manage binary objects having no intrinsic structure. An alternative design is to use a relational DBMS as a data directory, with the primary data in files outside the DBMS. A third alternative is an object-oriented DBMS, currently most favored in the academic community, but the technology is not yet mature enough for our large-scale operational requirements.

A combination of the first and second alternatives seems the best current approach, and one which provides a good migration path to object oriented systems. Most data in the POPS database will be stored in the form of large binary objects, or (for images) in UNIX files managed by the DBMS. Lat-long-time (llt) data will be supported in BUFR format, and gridded (or spectral, or other compressed) data in GRIB format. Metadata, whether descriptive or associative, will be stored as atomic data in ordinary database tables. Associative tables may be regarded as catalogs of GRIB and BUFR data. Descriptive tables describe grid or report formats, as well as descriptions of images. Geographical data will be in ordinary database tables, as well as other kinds of slowly-changing data. (NEONS also uses this data schema.)

4.2 Functionality

The relational model is mathematically sound and provides support for multiple user views of data without significant programmer overhead. The relational model is also easy to normalize (a process whereby redundant data is excised), and distributes data easily. It adapts easily to vectorization or parallel processing. It supports more natural query languages than previous data models.

Client-server functionality is needed to support supercomputing hardware architecture. This implies that data locking across distributed platforms with a two-phase commit is necessary, as well as semi-transparent network interfaces. The performance of fully transparent systems, in which the application does not need to know where data is stored, is not yet satisfactory.

Essential functionality also includes the ability to handle large binary objects, sophisticated indexing, data access optimization, and data integrity features. Data integrity must be insured during system changes and upgrades as well as during transaction processing. Integrity constraints include uniqueness, referential (other data must or must not exist), not null, and domain constraints. The more integrity features the better, such as user-defined datatypes, and procedures invoked by transactions.

Many other features now available in commercial DBMSs can increase utility and cut programming costs. Some of the most useful are "fourth generation" languages, graphical interfaces, access through UNIX shell scripts, and support for both interactive and batch use.

4.3 Performance Benchmarking

Industry-developed benchmarks for commercial DBMSs have not addressed the principal concerns of the scientific user. Some recent literature, however, is helpful if carefully interpreted. Cattell and Skeen (1990) have found that object-oriented database systems compare favorably with indexed file systems, while conventional relational systems do not. However, the relevant architecture here is that of the object, specifically the binary GRIB and BUFR objects that we expect the DBMS to deliver. The success of object-oriented benchmarks bodes well for our design.

Yao et. al. (1987) have shown that a database machine can significantly outperform a relational database system running on the same processor as the application, and that the database architecture is the principal factor. Again one needs to look closely at the architecture examined, distinguishing between software and hardware architectures. A symmetric multiprocessing database server resembles a database machine, more than it resembles a DBMS sharing a unary processor with an application. The Cray servers can have more than one processor, and we expect some of the satellite workstations to have multiple processors.

FNOC will extensively benchmark the new POPS DBMS before acceptance. The benchmark tests are based on the NEONS, and are designed to test all of the essential distributed features of the DBMS as well as performance. The tests do not use any atmospheric or oceanographic models, but do compress and decompress megabytes of data from GRIB and BUFR format, as well as inserting, deleting and updating the database. They effectively simulate the I/O found in our models. There are fifty-two tests in all, many near duplicates of others, but with various software or hardware configuration changes.

Assuring us an acceptable working DBMS upon delivery (in an industry plagued with "vaporware", i.e. promised software which fails to materialize on schedule, if at all) is not the only purpose of the benchmark. By adhering to standards, the benchmark software is designed to last the life of the system, which could be many years. We expect to quantitatively test the evolution of our data management platform as well as new releases of the DBMS.

4.4 File Systems

Not all questions associated with introducing a DBMS into a supercomputing environment have been answered. One in particular stands out. There is presently no relational interface between a distributed DBMS and any of the various distributed file systems supported in a supercomputing environment. For the Cray environment, the current distributed file system, Cray's Data Migration Facility (DMF), may be superseded by the Andrew File System (AFS), which is supported by the Open Systems Foundation (OSF) Distributed Computing Environment (DCE) (Lanzatella 1991).

One way to skirt most of the mass storage issues is to require the DBMS to use UNIX file I/O throughout the database. We have chosen this path to avoid future problems. Some immediate performance may be lost, but very little compared to the interface problems which might arise otherwise. Moreover, the industry is not likely to sacrifice relational database interfaces to file systems design, even if the relational interface is ad hoc, so no interface problems are anticipated. However, procedures to automate archiving, backup and recovery of the DBMS across distributed mass storage have to be developed in house until more tools come into the marketplace.

In the future, advanced data management systems may incorporate relational data management with very fast file server technology, such as data caching, prefetching, and concurrency. Considerable obstacles need to be overcome (Bell 1987). We do not require the DBMS to manage intermediate calculations for large forecasting models, but will use the DBMS whenever it becomes advantageous.

5. DEVELOPMENT SCHEDULE

The NEONS prototype is already available on Sun workstations at FNOC, with samples of production data. It is being enhanced on an as-needed basis to assist programmers converting environmental models. We are developing, again building on NEONS, an extensive set of FORTRAN and 'C' language interfaces to the DBMS. These interfaces make transparent to users changes in the underlying data model, particularly changes which may be necessitated by future performance considerations. We have already achieved real-time data migration from our current production system to the databases on Sun workstations, and have tested some preliminary modeling software with the prototype. No functional or performance problems have been encountered.

One of the singular virtues of contemporary relational database management systems is the ability to quickly redesign data structures, without causing the redesign of application programs. This means most of our prototype effort will be transportable to the production POPS database. No database in a dynamic operating environment is long static, and no relational database remains thoroughly well-normalized. We expect the POPS database to develop gradually and evolve continually.

During the coming year we plan to develop the archive, backup and recovery system. Three kinds of database storage will be available to users; transient storage enduring the length of a job, temporary storage which is not backed up, and permanent storage. Selecting among these types will mean selecting the UNIX file directory. All data in permanent storage will be automatically backed up and archived.

Applications will not suddenly change to the DBMS. We are allowing several years of dual hardware and software platforms for the transition. We have a parallel software engineering program to upgrade and convert production applications at FNOC to the POPS environment. At FNOC alone there are several million lines of code to migrate, covering a host of major models and highly specific applications.

Here is a POPS DBMS timetable, which depends mostly on hardware and software acquisition schedules:

- Workstation prototype online at FNOC, fall 1991
- Cray YMP2E delivery to FNOC, winter 1991
- DBMS delivery, spring 1992

- Cray prototype database online at FNOC, spring 1992
- Cray prototype database online at NAVOCEANO, summer 1992
- Archive, backup and recovery capability, fall 1992
- Cray YMP16 delivery to FNOC, winter 1992
- Initial operating capability at FNOC, spring 1993

6. SUMMARY

We are marrying two cultures, scientific supercomputing and business data management. Both are well established, with proven paradigms and tested techniques. Many specialized hardware and software data management systems continue to be designed. But the simplest solution, and perhaps the most elegant, is to use systems originally designed for business software in the scientific environment. The advent of commercial portable systems, enhanced for scientific applications, makes this possible.

Cultural change will accompany this change in technology. Introducing a fully distributed DBMS to an environment accustomed to linear or indexed sequential files means a lot of individual programming skills, acquired over many years, will no longer be as highly valued as they once were. New skills, emphasizing teamwork and data sharing, become more valued. Conversely, scientists should have more time to do science.

We have embraced commercial database technology as the only practical way to keep up with the rapid increases in data volume and data exchange expected in environmental computer systems. While we are pushing this technology to the limit now, we believe we will be well positioned for the future.

7. REFERENCES

American National Standards Institute, 1986: American National Standard Database Language SQL, ANSI X3.135-1986. New York, American National Standards Institute, 108 pp.

American National Standards Institute, 1989: American National Standard Database Language SQL with Integrity Enhancement, ANSI X3.135-1989. New York, American National Standards Institute.

American National Standards Institute, 1989: American National Standard Database Language Embedded SQL, ANSI X3.168-1989. New York, American National Standards Institute.

Bell, J. L. and G. S. Patterson, Jr., 1987, "Data Organization in Large Numerical Computations". The Journal of Supercomputing, 1, 105-136.

Cattell, R. G. G. and J. Skeen, 1990, "Engineering Database Benchmark". Sun Microsystems Technical Report, 27 pp.

European Center for Medium-Range Forecasts, 1988, "FM 94 BUFR: Binary Universal Form for Data Representation; Collected Papers and Specification". ECMWF, 92 pp.

Lanzatella, Thomas W., 1991, "An Evaluation of the Andrew File System". Cray Research Inc. report, 13 pp.

Silberschatz, Avi, Michael Stonebraker, Jeff Ullman, et. al., 1991, "Database Systems: Achievements and Opportunities". Comm. ACM, Vol. 34, No. 10, 110-120. (The entire October 1991 issue, devoted to "Next Generation Database Systems", is relevant.)

Superperformance Computing Service, 1989, "Data Base Management for Technical Computing". Dataquest Incorporated report, 158 pp.

Tsui, Ted L., 1991, "Naval Environmental Operational Nowcasting System (NEONS)". ECMWF Workshop Proceedings on Meteorological Operational Systems (this issue).

US Department of Commerce, 1988, "FM 92 GRIB". US Department of Commerce, National Oceanographic and Atmospheric Administration, National Meteorological Center, 39 pp.

Yao, S. Bing, Alan R. Hevner, and Helene Young-Meyers, 1987, "Analysis of Database System Architectures Using Benchmarks". IEEE Transactions on Software Engineering, Vol. SE-13, No. 6, 709-725.