SANDIA REPORT
SAND96–2094 • UC–903
Unlimited Release
Printed November 1996

Evaluation of Database Technologies for the CTBT Knowledge Base Prototype

Ralph Keyser, Ellen Shepard-Dombroski, David Baur,
Jim Hipp, Susan Moore, Chris Young, Eric Chael

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND 96-2094
Unlimited Release
Printed November 1996

Distribution
Category UC-903

Evaluation of Database Technologies for the CTBT Knowledge Base Prototype

Ralph Keyser
Ellen Shepard-Dombroski
David Baur
Jim Hipp
Susan Moore
Decision Support Systems Department

Chris Young
Geophysics Department

Eric Chael
Cooperative Monitoring Department

Sandia National Laboratories
Albuquerque, NM 87185

Abstract

This document examines a number of different software technologies in the rapidly changing field of database management systems, evaluates these systems in light of the expected needs of the Comprehensive Test Ban Treaty (CTBT) Knowledge Base, and makes some recommendations for the initial prototypes of the Knowledge Base. The Knowledge Base requirements are examined and then used as criteria for evaluation of the database management options. A mock-up of the data expected in the Knowledge Base is used as a basis for examining how four different database technologies deal with the problems of storing and retrieving the data. Based on these requirements and the results of the evaluation, the recommendation is that the Illustra database be considered for the initial prototype of the Knowledge Base. Illustra offers a unique blend of performance, flexibility, and features that will aid in the implementation of the prototype. At the same time, Illustra provides a high level of compatibility with the hardware and software environments present at the US NDC (National Data Center) and the PIDC (Prototype International Data Center).

Acknowledgments

This work was funded by the Automatic Data Processing portion (ST485D) of the Comprehensive Test Ban Treaty Research &Development Program at Sandia National Laboratories under Department of Energy contract DE-AC04-94AL85000.

List of Tables

List of Figures

1.0 Introduction

With the approach of a Comprehensive Test Ban Treaty (CTBT), the requirements placed on treaty monitoring systems continue to become more complex and demanding. Systems are now being required to monitor treaties with the synergistic application of multiple monitoring technologies, and the task of tuning and maintaining just the information needed to extract the relevant signals from a stream of sensor data is becoming more and more important. Proper application of information management technologies will benefit both the US NDC (United States National Data Center) and the DOE researchers by providing a well-organized, quality and version controlled store of the information needed to accurately process incoming data. This information management portion of the DOE regionalization efforts is known as the CTBT Knowledge Base.

The Knowledge Base ensures that the user of the information is able to quickly and easily apply the information when it is needed. They are also assured of having a store of information that can be maintained, tuned, and updated as knowledge is added or changed over time. This, in turn, ensures that the results of DOE regionalization research are provided in a form that enables them to be used by the customer without the additional steps of extracting information from paper reports and putting it in a form accessible by the codes used in automated processing.

Information management is a rapidly changing technology field that offers more options for storing and managing data than ever before. This paper is an effort to evaluate the options that can be applied to the CTBT Knowledge Base application. It examines a number of data management technology options and presents the pros and cons of each technology. The conclusions summarize the results and make a recommendation for this particular application.

1.1 Knowledge Base in Relationship to Overall Regionalization Efforts

Understanding the relationship of the Knowledge Base to the overall DOE regionalization efforts may help in understanding the requirements placed on the Knowledge Base. This section is intended to briefly summarize that relationship.

A significant portion of the overall DOE CTBT R&D program is the characterization of signal propagation in a few major regions of the world. This overall regionalization effort includes calibration of stations, identification of reference or master events, identification of physical effects, development of propagation models, and development of correction grids. The research that makes up this DOE regionalization effort is being conducted by the National Labs, universities, and private sector contractors throughout the country, and these efforts will yield results and information in a variety of forms ranging from paper reports to on-line databases. This relatively "raw" information will be critical to researchers in the future as they attempt to understand the "interesting" events that develop as the U.S. monitors a CTBT. In addition, a distilled subset of this information will be useful to the computer processes that routinely scan the complete set of data from the International Monitoring System (IMS) and the computer tools that are used by evaluators as they begin to look at events that the computer identifies as "interesting". This distilled subset of information, organized in a form that is readily accessible, is the CTBT Knowledge Base, and the information contained in the Knowledge Base will aid in event detection, location, and identification.

1.2 Summary Knowledge Base Requirements

Modern monitoring systems involve a large number of different computer processes with different requirements for the information needed to support the processing of incoming data. There are also different processing steps in each system that require different levels of information such as real-time processing vs. routine event processing vs. special event processing. There are also a large number of different organizations involved in the process of producing and using the information expected to be found in the Knowledge Base. All of these factors combine to make the precise definition of requirements for the Knowledge Base a difficult task.

As a result, the development of the Knowledge Base will follow an evolutionary development cycle that involves a series of prototypes. Past experience has shown this type of development to be very helpful for arriving at a useful product when requirements can not be clearly stated initially. An understanding of the requirements driving the development is critical, however, to understanding the trade-offs that arise when doing a technology evaluation such as this one. This section attempts to provide a summary of the requirements as they are seen today, and it is based on earlier efforts to define requirements for the Knowledge Base [Keyser 1995]. The reader should understand that these requirements may evolve along with the software as part of the evolutionary development cycle.

1.2.1 Support complex data structures

The information needed to support processing of sensor data results in complex, multidimensional data structures to describe the information. The Knowledge Base needs to be able to manage these complex structures.

1.2.2 Cover multiple technologies

The Knowledge Base is expected to provide information to support the processing of Seismic, Hydroacoustic, Infrasound, and Radionuclide data. Seismic, Hydroacoustic, and Infrasound are fairly similar in their requirements, although supporting the different technologies introduces complexity such as the need to support time variant data such as hydroacoustic travel times or infrasonic azimuth corrections. Radionuclide processing requires a different type of information structure that is very dynamic in nature.

1.2.3 Handle information on a global scale

Effective global monitoring will require that the Knowledge Base be able to handle information about any portion of the globe.

1.2.4 Handle information at different resolutions in different locations

Although the Knowledge Base will be global in scope, some regions are more complex than others and require that information be represented at a finer resolution. Other regions may simply have more information available for them, which once again requires a finer resolution. Therefore, the Knowledge Base must be able to support a varying density of information representation in different regions.

1.2.5 Interpolate between known points

Regardless of how dense or sparse data points may be in a certain region, the Knowledge Base must be able to provide information for an arbitrary point on the globe. It must be able to recognize when it is appropriate to interpolate given the existing data, and when it does interpolate, this mechanism must be fast and efficient. The actual interpolation process is a significant issue for the Knowledge Base since it must be able to handle edge effects, non-uniform density of data points, and meet requirements for accuracy and repeatability. The software, such as LOCSAT and EVLOC, that will be using this interpolated data impose additional requirements on the interpolated output for smoothness and continuity.

1.2.6 Support metadata

The Knowledge Base must be able to store metadata, which is data about the data. Examples include who originated the data, when, the uncertainty associated with the data value, etc. Some care must be taken with this process or the metadata can quickly be more voluminous than the actual data.

1.2.7 Provide quick access to information

The Knowledge Base is intended to support fully automated processes along with processes that run interactively. The details of interfaces to these processes are still being developed although a related requirement is 1.2.13 “Do not disrupt the existing processing environment” A goal, though not a requirement, of the Knowledge Base development is to provide a fast enough interface that other software processes could make library calls to the Knowledge Base to retrieve the information needed by the calling software processes. This would free the calling process from needing to read and store large quantities of data in memory. A better evaluation of this goal can be made when the first prototype is complete. Regardless of the results, programs that demand faster performance than the Knowledge Base can provide can always “pre-load” just the information needed for that program into a form that they can handle such as disk files. The GAcons program [LeBras 1994] is an excellent example of this approach. This adds a layer of complexity that makes tuning and maintaining the knowledge more complex, but ensures the performance required by the using software process.

1.2.8 Be easy to maintain

One of the difficulties with existing systems is that the information used by different processes is stored in different formats in different locations about the system, which makes tuning and maintenance of this information difficult. The Knowledge Base should provide a single source for information and should provide tools to allow reviewing, adding, deleting, and changing this information. The maintenance process needs to be controllable, however, and this is addressed under 1.2.11 "Support history / version control"

1.2.9 Be easily extensible in volume

Although the Knowledge Base may eventually contain large quantities of data for most of the world, it will start with smaller quantities of data in focused regions and grow with time. To support this trend, the system must scale well as additional volumes of information are added.

1.2.10 Be easily extensible in scope

Due to the evolutionary nature of this development, it is expected that there will be data types needed in the Knowledge Base that are not identified in the initial design documentation or prototype code. The Knowledge Base design must allow for both extensions to existing data types and the addition of new data types.

1.2.11 Support history / version control

The information stored in the Knowledge Base has a significant impact on the results of the automated processing system. As a result, it is important to provide control over which version of the Knowledge Base was in use at any given time. Version in this application has two meanings; the version of the overall "structure" of the data, and the version of a particular data entry. There will be a need for multiple versions of the Knowledge Base in existence simultaneously. For example, a research version with experimental changes and a stable operational version might both exist. The current preference seems to be that this be handled by separate physical copies of the Knowledge Base rather than virtual copies that originate from a single software service.

Along with the configuration management implied by the version control, it would also be useful to "go back in time" and recover the exact state of the Knowledge Base at some point in the past. This would help answer questions about how the system got a given answer at a given point in time.

1.2.12 Run in the US NDC/PIDC environment

The primary customer of the Knowledge Base is the United States National Data Center (US NDC), with the Prototype International Data Center (PIDC) as a possible additional customer. There are still issues open concerning how much of the Knowledge Base should be available to the PIDC, but the computing environments of the US NDC and PIDC are similar, so it is clear that the Knowledge Base should run in the software and hardware environment of the US NDC/PIDC.

1.2.13 Do not disrupt the existing processing environment

Many years worth of scientific and software development have gone into reaching the state of the current monitoring systems at the Data Centers. Significant efforts are still underway to improve the systems and make them ready to monitor a CTBT. The Knowledge Base should be able to exist in this environment without disrupting the existing process flow. Implementation should allow for an orderly phase-in of the Knowledge Base and the information it contains. Processes that want to use the Knowledge Base may do so, but other processes may continue to “do business as usual”. In particular, this means that the Knowledge Base must not interfere with the existing event-detection-feature database (the CSS 3.0 schema) or the process communication and management system.

2.0 Evaluation Description

Current operational monitoring systems rely heavily on text only files in a very specific format to remember the parameter and reference information needed to allow automated processing routines to do their jobs. These files are often called "flat files" and are managed just like any other file by the computer's operating system. This is an improvement over the previous practice of encoding this information directly into the software, since a change can now be made to a threshold or filter parameter without the need to recompile the software. However, as systems evolve toward monitoring a CTBT, the quantity and complexity of knowledge needed by the automated processing software continues to grow. The flat file scheme for storing knowledge becomes more difficult and cumbersome as the flat files become more numerous and more complex, so prototyping new mechanisms to manage this information needs to be part of the Knowledge Base development.

2.1 A brief overview of database technology

DataBase Management Systems (DBMS) are the most flexible and general of the information management systems, and today's market consists of a huge number of products to chose from. Despite the large number of different products, however, they can all be categorized into a number of broad classes of database technologies. These classes include Hierarchical DBMS, Relational DBMS, Object Oriented DBMS, and hybrid Object-Relational DBMS.

Hierarchical databases were some of the earliest formal data management systems. They organize data along a structure very similar to the way most operating systems manage file systems today. Data can be stored in a bundle or a folder, and this folder, in turn, may be stored in another folder along with other folders. This process continues until you reach the top of the tree, and all of your data is in a single folder. The hierarchical mechanism is relatively easy to implement, but only allows fairly simple queries against the data and does not support any notion of how data relates to other data beyond the fact that they may be in the same folder. These drawbacks make hierarchical databases useful only for relatively simple applications.

Relational databases were developed in response to the limitations of hierarchical databases. They store information in a tabular form, where each table consists of a number of columns that define the data types and their format, and the actual data makes up the rows of the table. Tables can use identically defined columns as a method of showing how one row of data is related to another row of data. This allows very complicated searches to be made against the data, but the rigid structure requires that the individual data types be fairly simple. Relational databases have been used extensively in information management systems, and they have been the state-of-practice for over a decade.

Object Oriented databases are relative newcomers to the information management scene. Developed as a response to the growing move toward object oriented analysis, design, and development, these systems are ideally placed to take advantage of the benefits of the object oriented paradigm for producing robust, reusable, easily maintained software systems. Object oriented databases store information as an object, so the data plus the code needed to perform certain functions on that data are stored as a single unit. This permits very complex data to be stored in a fashion that makes manipulating it from the outside easy, and makes interfacing with object oriented languages very straight forward.

Object-Relational databases are a hybrid of the two previous technologies. They continue to use the table format found in relational databases, but instead of simple data types in each column, they permit the use of a full object as a data type which includes the concept of additional software that knows how to operate on the object. Object-relational databases allow complex queries to be made against these complex data types.

In addition to the general DBMS's, there are also information management systems aimed at specific types of data such as Geographic Information Systems (GIS), Monetary Information Systems, and Document Management Systems for example. These specialized information systems come with additional software designed for analysis of their specific data type. Of these, the GIS's appear to be worth looking at as a mechanism for storing the georeferenced information expected in the CTBT Knowledge Base. Although there are GIS products available that use all of the database types mentioned above, a GIS is much more than just a database product. Their primary strength is provided by the large number of tools in the GIS package that help a user perform spatial analysis on the data. These tools are mostly interactive in nature and designed to help a human analyst with their review of the data. These interactive graphical components may make a GIS an excellent choice for assembling and reviewing the reference information used to build the Knowledge Base, but they impose a large quantity of un-needed overhead for automated access of the distilled Knowledge Base information. As a result, although a GIS may be used for products resulting from the DOE regionalization efforts, it was not considered a viable candidate for the overall information management task required of the Knowledge Base.

2.2 Selecting Candidate Products for the Evaluation

The sheer numbers of Commercial-Off-the-Shelf (COTS) products available for the platforms expected in the US NDC / PIDC environments, meant that some method needed to be found to reduce this number to a manageable size rather than doing a full evaluation on each product. An initial decision was made to actively try to evaluate the broad classes of databases mentioned in section 2.1 “A brief overview of database technology”. Literature and product searches were performed to identify market leading products in each class of databases, and additional factors were considered, such as required computing environment, before selecting a representative product from each class of databases.

Hierarchical databases were eliminated from the evaluation at this stage for two reasons. First, the geo-referenced, gridded nature of the data in the Knowledge Base does not lend itself well to a strict hierarchical organization, so a pure hierarchical database would be unable to perform the tasks required of it. In addition, hierarchical databases have migrated in the commercial world to specialized tasks such as large file system management or image system management and are no longer widely used as general purpose data management systems.

Relational databases were the class with the largest number of reasonable choices. The Oracle database system, in addition to being a market leader, is the system currently in use at both the US NDC and PIDC, so it seemed like a logical choice to represent the relational database technology. Conversations with the Oracle sales staff revealed that they have an add-on product known as the Spatial Data Option (SDO) specifically designed to handle spatial data in Oracle, so Oracle with the SDO was chosen as the representative for relational databases.

Selecting a representative from the Object Oriented database field was less clear cut. The object oriented database field is relatively young, populated with small companies, and a number of products would be a reasonable choice at this time. ObjectStore is a product that has been used at Sandia for several successful projects and was readily available for evaluation purposes, so ObjectStore was selected as the representative for the object oriented database class.

The Object-Relational class of databases is also a relative newcomer and still a fairly sparse field. The continuing push toward object oriented software methodologies coupled with the large installed base of relational database applications makes this a rapidly growing class of databases. Most of the large relational databases have announced plans to add object oriented capabilities to their databases, although these will be implemented as object oriented wrappers on top of a relational database. This approach amounts to a database on top of a database, which will probably adversely impact performance. In the meantime, one of the few products to be designed from the ground up to act as a hybrid object-relational database is Illustra, and it was chosen as the representative object-relational database.

In addition to the class representatives chosen above, another product seemed worthy of evaluation. This was the Spatial Database Engine (SDE) from Environmental Systems Research Institute (ESRI). ESRI is the maker of one of the world's best selling GIS products, and SDE was designed as the "next generation" spatial data management engine. So, although SDE is not a full GIS product, it was designed to support a GIS application, and therefore it offers the management of georeferenced data without the excess GIS overhead. SDE runs as a client of a relational database system, however, so it is technically part of the relational database class, but since it was custom made to handle spatial data, it was included as part of this technology evaluation.

2.3 Evaluation Environment

2.3.1 Hardware and Software

The computing environment chosen for the database evaluation (see Table 1) was selected as one similar to that found at the US NDC and PIDC. Both hardware and software were chosen to be compatible, but the evaluation was also intended to test the products under reasonable conditions, so one of the larger servers was pressed into action as the evaluation platform.

As a result, the evaluation server may not look exactly like machines found at the US NDC or PIDC, but should be very similar.

TABLE 1. Database Evaluation Hardware

Item	Model/Version	Comments
Processor	SPARC station 20	Dual Processor Model 71
Memory	128 Mbytes RAM	
Disk	30 Gbytes	Sparc Storage Array
Operating System	Solaris 2.4	
Display	TGX 8-bit color	Single Display
Graphics	X11/R5	

2.3.2 Network Environment

In modern distributed applications, the network plays a significant role in the response time of the overall system. Unfortunately, networks are also very difficult to control when performing an evaluation unless one is able to strictly control the network traffic, which was not possible for this evaluation without impacting other ongoing research efforts. In addition, the evaluation environment uses an ethernet network rather than the faster FDDI networks in use at the US NDC and PIDC. Consequently, the evaluations were done on a single machine in order to eliminate the network performance from the results. The delays imparted by the network should be relatively independent of the database technology selected.

2.4 Performance Evaluation

The basic goal of this evaluation was to determine how well the particular technologies returned the answer to "give me all the points within X km of this point", where this point was a random lat-lon pair. The datasets used for this evaluation were based on the concept of a grid of georeferenced data points that would hold the knowledge applicable to that point on the earth. For this evaluation, the points were strictly two dimensional with each point defined by a latitude-longitude pair. Two separate evaluations were run for each database on datasets of different size and complexity.

It should be realized that the data point recovery is only part of the overall problem of the Knowledge Base, so the problem has been simplified for the purpose of this evaluation. In the actual system, once the points have been returned, the Knowledge Base system must select the appropriate points, perform an interpolation to calculate the value at the point of interest, and return that value. There are a number of difficult issues such as boundary conditions and interpolant behavior that must be addressed in this area. There are also a number of obvious improvements to performance that can be made to the Knowledge Base. For example, rather than retrieve the entire set of data points for each query, these points can be cached to save time on subsequent calls. All of these issues are independent of the database, however, so eliminating them from consideration at this time makes the evaluation much more straight forward.

This evaluation also focused on 2-D datasets since the phase 0 prototype will be two dimensional. The third dimension can be represented by additional points in the plane (or on the sphere) that contain depth as an attribute. Since the bulk of the data in the Knowledge Base will be surface data, and since depth can be quickly and easily handled with an attribute, limiting the requirements to two dimensions makes the solutions faster and cheaper. As additional information is acquired and requirements are refined, it may be necessary to go to a full three dimensional representation of the Knowledge Base. If that is the case, then a reassessment of the database technology should be considered.

2.4.1 The Datasets

The datasets for this evaluation were constructed from dummy data as simple mock-ups for testing purposes. The small dataset consisted of 41,385 latitude-longitude pairs spaced fairly uniformly around the world. This dataset was based on a uniform 1° grid of the world with about 10,000 additional points added to represent depth in those areas of the globe that have deep seismic events. Each lat-lon pair had a single data value associated with it. In terms of number of points, this dataset is probably close to the lower limit of the Knowledge Base size.

The large dataset consisted of 508,885 lat-lon pairs each of which had a 32 byte data structure associated with it. This was based on the small dataset plus 9350 points arranged around each of the 50 expected primary station locations in the IMS. The amount of data associated with each point in this dataset is still too small, but the total number of points is probably of the right order of magnitude for the Knowledge Base.

Both of these datasets were designed simply as reasonable loads for the purposes of this evaluation. The quantity, quality, distribution, etc. of the data is not intended to imply anything about the distribution of data in the final Knowledge Base.

2.4.2 The Queries

Test routines were written for each database in C or C++ that generated query points, initiated the search, and measured the time from query initiation to data return. Queries against the small dataset involved requests for the nearest points within a 1000 km radius of a random lat-lon point. This run was repeated 10,000 times against different points to develop an average search time. The large dataset involved requests for the nearest points within a 550 km radius of a random lat-lon point, and once again this was repeated with 10,000 different points to develop an average search time.

3.0 Spatial Data Option (Oracle®)

Oracle is the world's largest vendor of information management software. With offices in over 93 countries world-wide, Oracle is a widely known and used vendor with numerous applications in both government and the private sector. The Spatial Data Option (SDO) is one component of Oracle's "Universal Server" approach to database services that consists of a number of integrated additions to the base Oracle product with the intent of providing the capability of handling multiple data types with their database system. According to the Oracle datasheets, SDO was designed to handle very large databases with hundreds of gigabytes of spatial data.

3.1 Cost

SDO is a relatively inexpensive addition once the original Oracle product has been purchased. Oracle's pricing structures are complicated, but SDO seems to be about a $1K addition. The original Oracle product, however, is quite pricey with costs ranging around $20K depending on a wide variety of variables.

3.2 Storage and retrieval technique

SDO uses a scheme they call Helical Hyperspatial CODE (HHCODE), which is a hierarchical quad gridding scheme. A given region is first divided into equal quadrants, and a standard relational database table is established for that quadrant. A "high water mark" is also established to define how large the table can grow. Points are entered into the quadrant database table depending on location of the point, and when a table exceeds its high water mark, that quadrant is sub-divided into a new set of quadrants and the points in the table so far are also distributed to this new set of quadrant tables. This process of adding points and sub-dividing quadrants proceeds until all the points have been entered. See Figure 1, "SDO subdivides quadrants of data to limit table size".

A search uses the quadrant relationships as an index to quickly guide the search down to the table that holds the data, and then a standard search of the table returns the row of interest. The level of the high water marks are important in this scheme since they strike the balance between the number of tables that get created and the number of rows in each table. Large numbers of tables have a negative impact on Oracle's performance, but large tables are also difficult to search quickly, so optimal high water mark selection is important.

3.3 Search Evaluation Results

The SDO evaluation was carried out on SDO version 1.3 and Oracle7 release 7.2.

Using an Oracle recommended high water mark of 50 rows, the small dataset resulted in just over 1000 individual tables in the Oracle database. The radial search returned nearly 300 points in an average of 0.55 seconds. This nearest neighbor type of search always crossed

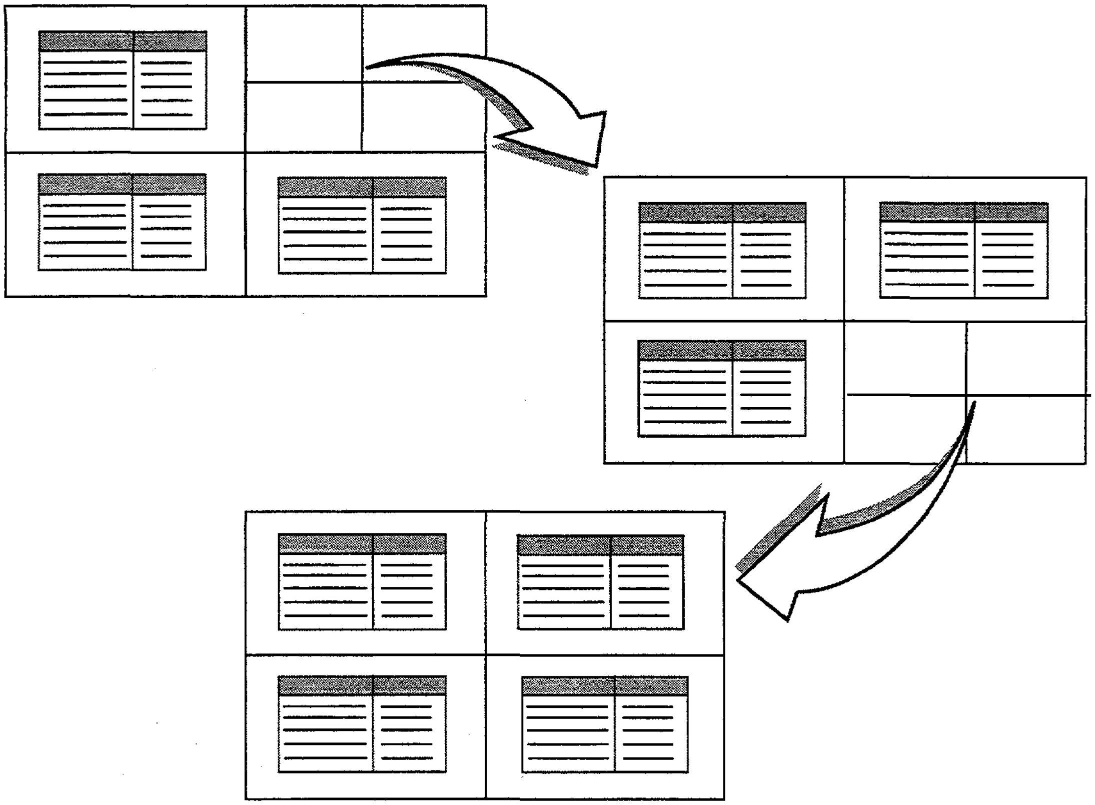

Figure 1. SDO subdivides quadrants of data to limit table size

table boundaries, so searching was slowed by the need to query more than a single table in a single search.

The large dataset presented more serious problems since it could not be loaded into the database without locking up or crashing the database server. Due to a lack of experience with this product, the evaluation team was unable to diagnose the problem and turned to Oracle's helpline for assistance. Unfortunately, Oracle's technicians were unable to solve the problem. The large dataset was, therefore, not evaluated with SDO. Expectations, however, are for upwards of 10,000 tables and slow performance.

3.4 Intangibles

Oracle is the current database of choice for both the US NDC and PIDC, so numerous people in these organizations have worked with Oracle systems. This means that existing personnel have had training and experience with the Oracle product, so the addition of another database would not require additional training. Due to the expected size of the Knowledge Base and the need to avoid impacting the normal CSS database, a separate Oracle license is expected to be needed for the Knowledge Base. Therefore, no cost savings are expected in the licensing area.

Oracle provides a complete set of tools for doing backups, restores, imports, exports, etc. They also have mechanisms for taking advantage of parallel computers, distributed databases, and redundant disk systems, so they provide excellent support functions for their primary product. The experience in this evaluation with software support was not at all satisfactory, although SDO is a new product, so support may improve with time.

Oracle supports Structured Query Language (SQL) queries of their database, and this permits ad-hoc review and examination of the data. This capability has proven to be extremely useful in the past for performing a number of unexpected tasks with the data in the database.

3.5 Further Information

Oracle's web site contains more information about the company and product brochures on all of the product offerings. Oracle's web site is located at: **http://www.oracle.com.**

4.0 Spatial Database Engine (ESRI®)

The Spatial Database Engine (SDE) is an application designed to run as a client to a Relational Database Management System (RDBMS) server, while SDE itself is invoked from an application via a series of C library calls that make up the SDE application programming interface (API). The SDE software provides highly tuned spatial data index and management while still using standard RDBMS technology for actual data storage functions. This approach provides high performance spatial functions while still retaining the benefits of established database systems.

ESRI is the maker of ARC/Info, one of the world's leading geographic information systems (GIS). ESRI got its start as an environmental company and developed one of the first GISs to support their work in pollution and waste management. Today, their work is all GIS software related with a broad range of products, and their software is used in a wide variety of fields.

4.1 Cost

SDE is the most expensive of the technologies in this evaluation since the user is required to purchase both an SDE license and a license for the supporting RDBMS. Licensing costs vary, but a single SDE license should be in the $17K - $20K range.

4.2 Storage and retrieval technique

ESRI considers the details of their data storage and retrieval scheme proprietary, but in general, they are using an indexed polygonal binning system. This scheme indexes each stored point to a non-overlapping polygon and then does a linear search through the polygon to find the point for retrieval. The polygon definitions, points, attributes, indexes, etc. are stored in the database system since no data is stored by SDE itself. However, SDE has been highly tuned to provide very fast access to spatial data, so portions of this information set are cached in memory by SDE to improve the access times to the information.

One downside to the approach, however, is that SDE can only store a single attribute value for each point. This presents a problem when storing the complex data expected at each grid point in the CTBT Knowledge Base. There is a solution, however, and that is to store an identifier value with the point and use that id as an index to the full dataset. This approach works, but requires an additional call to the database to retrieve the data and that results in a decrease in performance.

4.3 Search Evaluation Results

The evaluation was carried out with SDE version 2.1 and Oracle7 release 7.2.

The small dataset worked well with SDE since it only stored a single value at each point. The search returned an average of 295 points at an average speed of 0.078 seconds. Discussions with SDE engineers indicate that this would probably scale in a roughly linear fashion.

The large dataset was more of a problem since it required a second table to store and an additional access to retrieve the information. The recommended scheme of storing an index was used, and the results for returning just over 600 points was an average time of 1.237 seconds.

4.4 Intangibles

ESRI is quick to point out that SDE is independent of the underlying database, but at this time their only supported database is Oracle. They are actively working on an interface with Informix, and are considering others such as Sybase. Since Oracle would probably be the vendor of choice for a relational database system, this is not seen as a drawback.

SDE might offer some benefits from being the same engine used to drive the GIS system used at the US NDC and PIDC. At this time, however, ESRI is not using SDE to support their lead product (ARC/Info), although they do provide an SDE interface from ArcView which is a GIS in its own right, but is not as capable as ARC/Info. It is also not clear that ARC/Info will be the GIS of choice for the Data Centers, although it is certainly one of the leading contenders. As of the 1996 ARC/Info User's Conference in May, ESRI has plans to provide an interface between SDE and ARC/Info in version 7.1 which is scheduled for late 1996, so this technology may deserve another examination early in 1997.

The issue of data storage for more than one data value per point is a significant drawback to SDE, and discussions with their marketing folks indicated that they realize that it is a design limitation. The marketing department indicated that the problem would be solved in version 3.0 of SDE which should ship in late 1996 or early 1997. Details on how they would overcome this limitation were not yet available, but a solution to this problem would significantly enhance the way SDE performed in this evaluation.

The full set of Oracle tools would be available to handle database administration functions. This would also mean that relatively little additional training would be required for Oracle trained personnel. Oracle's support organization would be responsible for Oracle related problems. ESRI's support organization would need to address SDE related problems, and the experience with ESRI has been good as far as support is concerned.

Although the SQL queries will work against the data stored by SDE in Oracle, the indexing schemes are not easily unraveled by a casual user. This limits, to some degree, the usefulness of SQL for browsing or querying the database.

4.5 Further Information

ESRI's web site contains marketing information, white papers, release information, and other details. It is located at: **http://www.esri.com,** and includes a white paper on SDE [ESRI 1995].

5.0 Illustra (an Informix company)

Illustra is a unique hybrid database that provides a set of object oriented features while still retaining familiar relational techniques in its data storage mechanisms. This feature set was designed into the database from its inception, so the integration is seamless. In addition, Illustra supports the concept of "DataBlades", which are additional software libraries that get compiled directly into the database to allow it to intelligently manage different types of data such as documents, video, medical data, web pages, audio, spatial data, etc. Users can write their own DataBlades to provide mechanisms for manipulating their unique data types. This combination provides tremendous power while still remaining compatible with relational standards such as SQL.

Illustra was developed by Dr. Michael Stonebraker at the University of California, and it represents the commercialization of the Postgres research project which started in 1982. Dr. Stonebraker is also the original designer of the Ingres database and has a long history of research in the database field. Illustra was first released in August of 1993, but already has offices and applications world-wide. Illustra was recently acquired by Informix Corporation, one of the major relational database vendors.

5.1 Cost

Illustra is priced on a "per concurrent user basis" with floating licenses on the local network. In small numbers, it costs about $1.9K per user. This cost includes the addition of the 2-D DataBlade. There are, of course, a number of different pricing schemes with increasing price breaks for larger purchases.

5.2 Storage and retrieval technique

Illustra sells both a 2-D and 3-D DataBlade that add the ability to manipulate spatial data to the basic Illustra database. The 2-D blade was used for this evaluation, and it uses an R-tree searching mechanism for data indexing. The R-tree mechanism is a well known technique for spatial data indexing, and it involves a hierarchical structure of possibly overlapping bounding polygons. Each polygon splits into multiple, lower level polygons as you descend until you are able to isolate the point of interest. See Figure 2, "R-tree searching mechanism," on page 18. R-tree mechanisms are fast, efficient and work in both 2 and 3 dimensions.

5.3 Search Evaluation Results

The evaluation was carried out with Illustra version 3.2.

Illustra handled both datasets without any problems. The small dataset returned an average of 294 points in an average time of 0.081 seconds.

The large dataset returned 613 points on average with an average search time of 0.823 seconds.

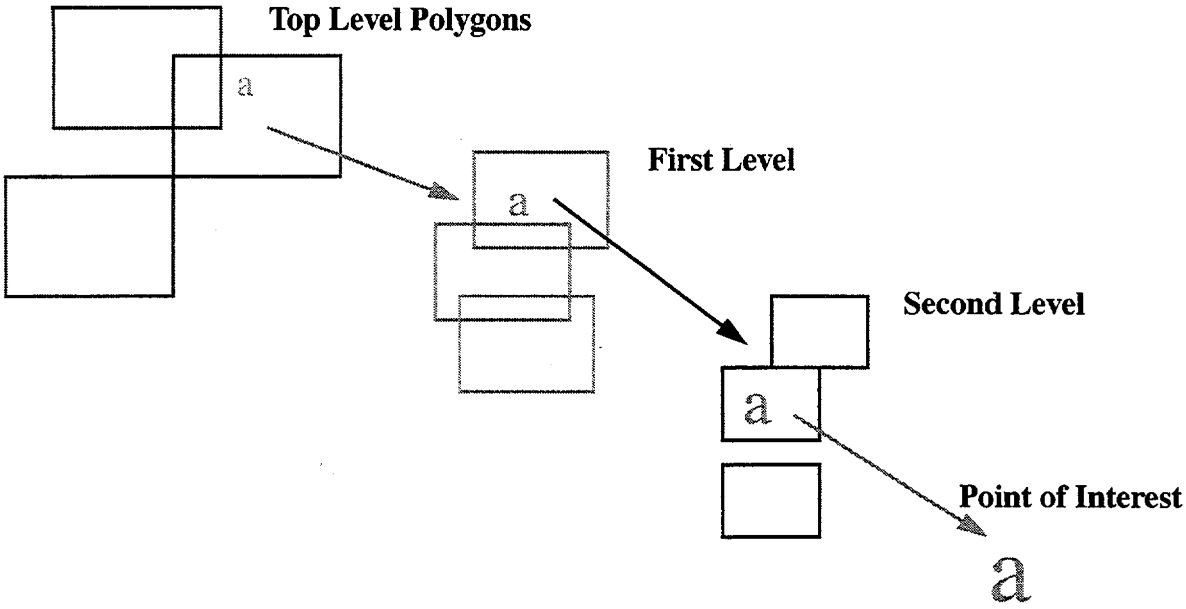

Figure 2. R-tree searching mechanism

5.4 Intangibles

Illustra's ability to add code to the database server provides tremendous potential for dealing with complex data structures in an intelligent fashion. This could provide opportunities for dealing with items such as time and frequency variant corrections or frequency dependent dispersion curves. This capability also offers the opportunity to enhance the performance of the spatial searching to take advantage of the uniqueness of Knowledge Base data structures.

Illustra is a fairly young database, so although it has a set of database administration tools, they are not yet fully mature. The recent purchase of Illustra by Informix should help in this regard since Informix does provide a full set of tools with their product.

Although it is a different database technology from Oracle, Illustra retains some similarities. The table structures are similar, and the SQL access is identical across the technologies. This provides a high degree of compatibility in environments that support both databases. Programs can use the same SQL scripts to access Oracle or Illustra databases and can even access both databases simultaneously if needed. Many of the software processes in use at the US NDC and PIDC use the Generic Database Interface (GDI) from SAIC which has used an Illustra interface in the past. Illustra will fit easily into the environment at the data centers, will not disrupt the existing Oracle database operations, and could be easily interfaced to from existing software.

Using Illustra will require additional training, but the Illustra training should come easily to personnel trained on Oracle. Illustra has a complete set of training programs available either at their locations or at the customer site that cover the complete range of skills from administration to application development.

The experience with Illustra's technical support was excellent during this evaluation. They were quick to provide answers to questions and provided direct support from the designers and developers when needed to solve problems. They also provide a network of independent value added resalers across the country who can provide consulting and support services as needed.

Illustra provides full SQL support for its database product. This permits the flexible, adhoc queries that have proven useful in the relational database environment. Since this is identical to the SQL for RDBMS, no additional learning curve is required for accessing the database in this fashion.

5.5 Further Information

Illustra's web site is stored entirely in an Illustra database and provides a number of demos and white papers in addition to marketing information. One of the white papers [Colton 1994] specifically addresses spatial data storage. Illustra's web site is located at:
http://www.illustra.com

6.0 ObjectStore (ODI)

ObjectStore is a full object oriented database that provides reliable, high-performance data storage and retrieval in a seamless fashion for object oriented applications in C, C++, and Smalltalk. ObjectStore provides persistent objects that do not go away when the application quits running. Otherwise the database is almost invisible since the objects used in the application are the same objects that are stored in the database. This permits tight integration of application and database, and due to ObjectStore's multi-threaded architecture and light weight thread implementation, developers can achieve very fast throughput.

Object Designs, Inc. (ODI) claims that ObjectStore is the world leader in the Object Database Management System (ODBMS) market, and they are one of the most popular ODBMS products on the market today. A number of successful projects have been completed based on object oriented design methodologies and centered around the ObjectStore database.

6.1 Cost

ObjectStore is priced with a floating licensing scheme that requires a license for each simultaneous user. The licenses are sold in packages, with price breaks for larger package purchases, but in small quantities, licenses are about $1.5K each.

6.2 Storage and retrieval technique

Unlike all of the other database systems examined in this evaluation, ObjectStore does not provide specialized spatial data management software. Instead, they provide some very general features that make it easy for a developer to work with the persistent objects that are stored by ObjectStore. It is left up to the developer to create the searching technique that handles the data correctly.

For the purpose of this evaluation, an implementation of the R-tree searching mechanism was developed. This mechanism is fast, efficient, fairly easy to code, and it is functionally identical to the mechanism used by the 2-D spatial DataBlade in Illustra. More detail on how the R-tree search works can be found in section 5.2 "Storage and retrieval technique"

6.3 Search Evaluation Results

ObjectStore was easily the fastest of the databases in this evaluation. For the small dataset, the search returned an average of 293 points in an average time of 0.0012 seconds. The large dataset searches took 0.015 seconds on average to complete. A significant reason for this speed is the fact that the developer had complete control over the search code. There is no excess overhead or provisions to support unusual or difficult data types, so the code can be optimized to run very fast. There was also limited support for error checking and correcting, so operational code would be expected to run a little slower.

6.4 Intangibles

Although ODI is a relatively young company, they have shown steady growth and an increasing market share. They appear stable enough to provide long term support for the Knowledge Base product. It seems likely that they will eventually become a buy-out candidate for one of the larger database companies.

Tools for database administration are sparse with the ObjectStore package. Although this is an area that is receiving a great deal of effort, at this time there are some tools, but far from a complete set.

One of ObjectStore's great features is also a significant drawback, and that is the database's lack of embedded code. Although this means that there is almost no excess overhead in Object Store, it also means that almost nothing can be done without developing the code. This makes for a very flexible and efficient system, but requires a longer development cycle to get that system ready to go.

ObjectStore integrates tightly with the C++, C, and SmallTalk languages, but these are also the only methods of interfacing with the database. This means that SQL will not work for ad-hoc queries format this database. ODI is developing an SQL interface, however, with a release date of Summer '96. If this product works as advertised, then ObjectStore will have the same type of ad-hoc query structure as the other databases in the evaluation.

ObjectStore represents a significant paradigm shift in the way database systems are designed, developed, and maintained. The object oriented approach has proven to produce much more robust, reliable, and reusable software if the method is followed correctly. Unfortunately, this shift in thinking also requires a large retraining effort. Little of the current database administration skills are useful for dealing with ObjectStore, so new training will be required.

6.5 Further Information

Additional information on ObjectStore can be found on Object Design's web site at:

http://www.odi.com

7.0 Summary

This evaluation has taken a rough mock-up of the type of data structures expected in the CTBT Knowledge Base and made an initial evaluation of how well each of four different database systems handled the problem. This summary section will provide a brief recap of the results seen with each database system and present a recommendation for a product to be used in the initial prototypes of the Knowledge Base.

7.1 Evaluation Summary

7.1.1 Spatial Data Option

SDO is not a good solution for the Knowledge Base data storage problem. Although it may be useful for certain applications, its mechanism for storage and retrieval of data is poorly suited to the type of data and queries expected in this application. Search times were slow and significant problems were encountered with the system. Software support was less than adequate.

7.1.2 Spatial Database Engine

SDE showed promise in this evaluation, but it's biggest drawback for the Knowledge Base application is the limitation of storing only one word of data with each spatial point. Given the expected complexity of the data needed at each point, this drawback is significant. This technology deserves watching, however, since this limitation may be removed in future releases.

7.1.3 Illustra

Illustra's 2-D DataBlade shows promise for use in the CTBT Knowledge Base. The technology provides flexibility for dealing with complex data structures, and there is significant opportunity available for performance improvement. Illustra also supports SQL queries which helps lower the learning curve for accessing this database. Support for the product was excellent.

7.1.4 ObjectStore

ObjectStore showed the best performance in this evaluation by a substantial margin, and it would be a clear winner if speed was the only issue. Unfortunately, prototyping the Knowledge Base with ObjectStore would require a great deal of code development since ObjectStore does not provide anything more than the very basic code for object management. This would prove to be a serious drawback in the cyclical prototyping effort expected for the Knowledge Base.

7.2 Recommendations

After completing this evaluation, the recommendation of the design team is to develop the initial prototype using the Illustra database management system. This technology offers a good balance of performance with the expected data, flexibility, and built-in features that should aid in rapid prototype development. In addition, its SQL support and commonality with relational systems should make it easy to develop for the target environment and provide excellent compatibility with the existing Oracle environment at the data centers.

The information management field is marked by rapid change, so at the end of the first phase of the Knowledge Base prototype, the performance, experience, and requirements for the Knowledge Base should be reassessed. If changes in the database are required, they can be easily addressed in the evolutionary prototype approach to the Knowledge Base.

References

Colton, M. Illustra, Relational Databases and Spatial Data, *Illustra Technical White Paper*, November 1994.

Environmental Systems Research Institute. Spatial database engine (SDE), *ESRI White Paper Series,* July 1995.

Le Bras, R., H. Swanger, T. Sererno, G. Beall, R. Jenkins, W. Nagy, and A. Henson. Global association final report, *SAIC document #SAIC-94/1155*, 28 pp., 1994.

Keyser, R., H. Armstrong. Proposed conceptual requirements for the CTBT knowledge base, *Proceedings 17th Ann. Seismic Research Symp., PL-TR-95-2108 pp 774-784,* 1995.

Distribution

1 Wendee M. Brunish
MS F659
Los Alamos National Laboratory
PO Box 1663
Los Alamos, NM 87545

1 Leslie Casey
U.S. Department of Energy
NN-20
1000 Independence Ave. SW
Washington DC 20585-0420

1 Dan N. Hagedorn
MS K5 12
Pacific Northwest National Laboratory
PO Box 999
Richland, WA 99352

1 Dave Harris
MS L205
Lawrence Livermore National Laboratory
PO Box 808
Livermore, CA 94551-0808

1 George E. Randall
EES-3 / MS C335
Los Alamos National Laboratory
PO Box 1663
Los Alamos, NM 87545

1 Dr. Stan Ruppert
MS L-205
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94550

1 Tom Sereno
SAIC
10260 Campus Pt. Drive
San Diego, CA 92121

1 Henry Swanger
SAIC
10260 Campus Pt. Drive
San Diego, CA 92121

1 Ray Warner
MS K6-48
Pacific Northwest National Laboratory
PO Box 999
Richland, WA 99352

1 Jay Zucca
MS L205
Lawrence Livermore National Laboratory
PO Box 808
Livermore, CA 94551-0808

Sandia National Laboratories

1	MS1138	Hillary Armstrong, 6533
1	MS1138	Judy Beiriger, 6532
1	MS1138	Larry Ellis, 6531
1	MS1138	Tony Edwards, 6532
1	MS1138	Jim Hipp, 6532
20	MS1138	Ralph Keyser, 6531
1	MS1138	Bruce Malm, 6532
1	MS1138	Susan Moore, 6532
1	MS1138	Ellen Shepard, 6532
1	MS1138	Randy Simons, 6532
1	MS1138	Julian Trujillo, 6532
1	MS0979	Dale Breding, 5704
1	MS0979	Don Shuster, 5704
1	MS0979	Larry Walker, 5704
1	MS0655	Eric Chael, 5736
1	MS0655	Dorthe Carr, 5736
1	MS0655	Mark Harris, 5736
1	MS 0750	Chris Young, 6116
1	MS9018	Central Tech Files, 8523-2
5	MS0899	Technical Library, 4414
2	MS0619	Review & Approval Desk, 12630 for DOE/OSTI