DØ Database Working Group Report
DØ Note #XXXX

Pushpa Bhat, Stan Krzywdzinski, Lee Leuking, Jerry O'Connell & Laura Paterno

1. CHARGE

The purpose of the Database Working Group, a subgroup of the DØ Software Tools Group, was to evaluate the database management systems currently used by DØ, and come up with a list of choices to be further evaluated in detail. Several appendices have been added to the back of this report which contain a brief description of the features of the top candidates and a cost estimate for the candidates.

2. CURRENT DATABASE MANAGEMENT SYSTEMS

Currently, DØ utilizes several different database management systems.

2.1 DBL3 databases

DBL3 is a ZEBRA based database system used for storing calibration and monitoring information for use in event reconstruction. The information is written during data taking and compressed and copied to offline to be used in reconstruction. Access to information is based on a key or set of keys.

2.1.1 Calibration databases

These databases are used for the various detector groups Calorimeter, CDC, VTX, FDC, WAMUS, SAMUS, LØ, etc. to store calibration information. The largest of them is the calorimeter which consists of 30 (each 300Kbyte uncompressed or 125Kbytes compressed) files. The rest are much smaller in comparison and several were barely used. They are accessed once per run for reconstruction purposes. They are all keyed by run number with a secondary key by time.

2.1.2 Monitoring databases

These databases include DBMON, HV, Luminosity and Run Summary databases. The DBMON database is by far the largest of these with 50 (each 60Kbytes uncompressed) files. The luminosity database was only used for Run 1A. The DBMON database is accessed once per event by the event reconstruction program. They are all keyed by time.

2.2 FATMEN catalog

The FATMEN catalog is a hierarchical, single file, database based on the direct access package RZ of ZEBRA; its current size is 1220 Mbytes. It stores information about raw and processed data files, and Monte Carlo files for Run 1, in a 7-level directory tree; currently it contains 1.6 M files in 13K directories. Besides being an inventory of files and tapes, its main purpose is to provide information about where files are stored and on which media (disks, various kinds of tapes, robots, etc.). It is coupled with a staging server, which is invoked whenever a file is requested by its name, but the file is not available on a disk. There are no explicit parental links for the files, only implicit ones through the file name and/or a comment field. Corruption of the catalog occurs once a year on the average, and it takes a few hours to rebuild the whole catalog from a backup. Backups are done routinely every day.

2.3 DEC Rdb databases

DEC Rdb is a relational database product commercially supported first by Digital Equipment Corporation (when first started using it) and later by Oracle.

2.3.1 HDB

The Hardware DataBase is a single file database, of size 18.4 Mbytes. It has been used in the monitoring and control of the D0 experiment. The database stores all of the essential attributes of the hardware and provides CDAQ processes with the access path required to read from or write to a device. It contains information on detector systems, low and high voltage power supplies, cryogenics and argon purity monitoring, and environmental conditioning. This central database also serves as the master copy of the local databases which reside in the front-end computers. Fast access and reliability are required for this database. It has never had a corruption problem.

2.3.2 PROD_DB

The PRODuction DataBase is a multifile database, of size 1582 Mbytes. It stores information about runs, raw and processed data files, and tapes from Run 1, in particular trigger, filter and luminosity information. Currently it contains 7.5 K runs, 1.6 M files and 48.6 K tapes. There are explicit parental linkages for the files implemented according to the relational model. The database is accessed daily for luminosity information as well as to gather processing statistics.

2.3.3 TAPE_LOG_DB

This database is a multifile database, of size 446 Mbytes. It is used to store raw file, trigger, filter and begin-end run information. Information is written into it by several online processes and it is read by the Production Database three times a day during data taking.

2.4 Oracle databases

Oracle is a relation database similar in style to DEC Rdb. It has many of the features that Rdb has. The training database is implemented using Oracle 7. This database is used to keep track of all the safety training that each DØ person has taken (DØ hazard awareness, etc.)

3. RUN 2 NEEDS

Many questions arise when discussing what we require for future database management system which can satisfy the wide variety of needs that we have for a database system.

3.1 What Databases Do We Keep for Run 2?

Not all of the databases that we used in Run 1 should be used in Run 2.

Several of them contained identical information. The FATMEN Catalog and the Production DataBase could (and should) be merged into one database. They contained almost identical file information and a little more information added to the Production DataBase would allowed it to provide the same functionality as the FATMEN Catalog. The staging part of the FATMEN catalog is actually a separate piece from the catalog itself and could be modified to work with a combined FATMEN and Production Database.

Several of the databases were created but never utilized for Run 1 (i.e. the luminosity DBL3 database). Also DØ has chosen to move away from ZEBRA so all the DBL3 databases will need to be replaced with a new database.

3.2 Centralized or Distributed Database Servers

At this stage of analysis it is hard to say whether we want to configure our database servers as centralized or distributed. All of the products we review and consider should be able to support either configuration.

4. TYPES OF DATABASES AVAILABLE

There are currently 3 database technologies on the market. The following sections describe each on from oldest to newest.

4.1 Relational (RDBMS)

RDBMSs provide highest performance for applications that primarily access data by value of simple, pre-defined, data types stored in flat tables. RDBMS are not good at complex data like video, audio, compound documents, animation, arrays, geographic information and composite objects. The ability to share data across programming languages (by embedding the standard SQL in them) is considered a major benefit of relational DBMS approach. SQL serves as the programming language neutral interface to the database.

RDBMSs tend to be used for database applications with the following characteristics: simple data that can be represented in tables using built-in data types supported by the relational DBMSs flexible access which can be phrased simply using the value-based constructs of SQL simple transactions that the DBMS can process in short periods (i.e., milliseconds).

Examples:

- Oracle, latest version 7.? - Oracle Rdb, latest version 7 with 64-bit support announced in March 1996
- IBM's DB2 and SQL/DS
- SyBase's SQLServer
- Ingres
- Informix
- MiniSQL by David J. Hughes from Australia (free and runs on UNIX)

4.2 Object (ODBMS)

ODBMS standard is still evolving; not long ago ODBMS were not suitable for multi-user, mission-critical environments. ODBMS provide higher performance when data access is primarily navigational via object references. The primary contribution of object databases is object model support. Object DBMSs tend to be used for database applications with the following characteristics:

1. complex data that is highly interrelated and/or includes multimedia data types
2. high-performance needs with predictable access patterns and localized working sets
3. distributed computing environments, which commonly involve collaborative work, long transactions, and versioning.

Examples:

- Objectivity
- ObjectStore by Object Design
- O_2
- UniSQL
- Versant

4.3 Hybrid (ORDBMS)

Hybrid ORDBMS are advertised as the "best of both worlds", the ones which combine the security, integrity and reliability of RDBMS with the power and ability to manage complex data of object database technology.

Examples:

- Illustra (bought by Informix from Illustra Info Technologies)
- Odapter by Hewlett-Packard
- Oracle version 8, awaited but not released yet ...

5. WORKING GROUPS DATABASE THOUGHTS

After reviewing all the databases from Run I and the needs for Run II it became clear to us that a relational database would not be sufficient for our purposes.

Currently the Production Database access time is greatly impeded by having the information stored in a relational format. The number of join tables (a way to link one table to another in order to get a one-to-many or many-to-many relationship) required to retrieve information from the database pushes the limit of what is recommended for relational databases.

DØ has also chosen to use C++ for Run II coding. While the commercial relational databases used for Run 1 performed very well and have C++ interfaces, you cripple the C++ in order to work within the relation scheme. You have to have a translation performed each time to go from the relational format to the object format.

6. DATABASE TESTING

No testing of any of the databases was actually done. This would have taken a great deal of time and effort to accomplish on such a short time schedule. However, we did determine that most of the vendors did provide free evaluations of their products and would even provide a minimum amount of people support as well.

7. DATABASES REVIEWED

We have reviewed six database companies. Two have products which are hybrid and the other four are pure Object databases. These companies include:

- Illustra - Illustra, Inc. (now a subsidiary of Informix)
- O_2 - O_2 Technology, Inc.
- Object Store - Object Design, Inc.
- Objectivity - Objectivity, Inc.
- UniSQL - UniSQL, Inc.
- Versant - Versant Object Technology

After careful consideration we have removed two of the candidates from our list. They are Illustra and Versant. Illustra was removed because it is not a true Object Database or even a hybrid. It is a relational database with C++ wrapped around it. It also has just recently been purchased by Informix and will be undergoing major modifications over the next several years. Versant was removed because it did not provide enough basic database tool support.

We also removed MSQL from consideration because the current version allowed for only a single key (Run Number, Date) to access data and most of our databases needed multiply keys. Also this is a freeware product and it was agreed but everyone in the group that we would not consider any freeware due to the tremendous problems we encountered using DBL3 (corrupted data, manpower to support it, etc.).

8. OVERALL COMPARISON OF THE TOP DBMS CANDIDATES

Database Service	**Object Store**	**Objectivity**	**O_2**	**UniSQL**
Native ANSI SQL Support	**Y**	**N**	**N**	**Y**
Mission-critical DBMS Services	**Y**	**Y**	**Y**	**Y**
Scaleable Performance	**Y**	**Y**	**Y**	**Y**
Standards-based Data Model	**Y**	**Y**	**Y**	**Y**
Triggers, Views and Authorization	**Y**	**?**	**?**	**Y**
On-line Catalog Management	**Y**	**?**	**?**	**Y**
Complex Object Navigation	**Y**	**Y**	**Y**	**Y**
Methods @Inheritance	**Y**	**Y**	**Y**	**Y**
C++, Smalltalk & OQL Support	**Y**	**Y**	**Y**	**Y**
Real-world Data Modeling	**Y**	**Y**	**Y**	**Y**
ODMG Compliant	**Y**	**Y**	**Y**	**?**
WEB Connectivity	**Y**	**Y**	**Y**	**Y**
Object/Schema Versioning	**?/?**	**Y/Y**	**Y/Y**	**?/?**

Open Multimedia Framework	**?**	**?**	**?**	**Y**

9. SUMMARY

The database working group has evaluated the features of six databases in addition to discussing the ones we already use. We have narrowed down the list of candidates to four: UniSQL, Objectivity, Object Store and O_2.