01/10/97

DØ Note # 3123

DataBase Management System (DBMS) Recommendation for DØ

Pushpa Bhat, Herb Greenlee, John Hobbs, Alan Jonckheere, Stan Krzywdzinski, Lee Leuking,
Qizhong Li-Demarteau, Jerry O'Connell, Laura Paterno, Harrison Prosper, Serban Protopopescu

1. INTRODUCTION

The DØ Database Working Group, a subgroup of the Software Tools Group, was charged with assessing the needs and requirements for Run II DataBase Management Systems (DBMS). In order to evaluate these needs, we have re-examined the uses of DBMSs in Run I. We have considered the potential uses of DBMSs in Run II and established several requirements which we feel must be met by any DBMS which we might use. Finally, we have done preliminary research into several commercial and public domain products which might be suitable to our needs. A detailed summary of the DataBase Management Systems reviewed by the Database Working Group is available in DØ Note #3122.

2. RUN I DATABASE

During DØ Run I, several DBMSs were used to manage a wide variety of information. They include DBL3, RZ, DEC Rdb and Oracle.

DBL3 is a CERN database product which provides keyed access to information using the ZEBRA RZ file structure. It was used for managing detector calibration and monitoring information which was stored on-line, compressed and distributed to the off-line processing cluster. This product was chosen in part because of its platform independence. FATMEN is a hierarchical data management system used to catalog the raw event data and track the off-line data processing. It employs the RZ file structure to provide keyed access to stored information.

DEC Rdb[1], a commercial relational DB product from Digital Equipment Corporation, was used for the Tape Log DB, Hardware DB and Production DB. The Tape Log DB kept track of all run, trigger, filter and raw file and tape information taken on-line. The Hardware DB was used in the on-line system to provide device access information to the front end system for control and monitoring purpose. The Production DB managed all the run, trigger, filter, file and tape information for all runs and processed data for Run I. It also managed all the luminosity information for the runs. Oracle, a relational database designed by Oracle, Inc., has similar features to DEC RDB and was used to track the DØ personnel training information during the run.

3. RUN II NEEDS

Many issues arise when discussing what we need for information management during the next DØ running period. A wide variety of information will need to be maintained during Run II. Calibration, monitoring, luminosity, run summary, trigger, filter and event information will need to be stored when data taking resumes. Access information for hardware devices, used for control

[1] DEC Rdb was purchased in 1995 by Oracle, Inc. and is now supported by them.

and monitoring purposes, will also be needed. Off-line, many of the databases written during running will need to be accessed by the reconstruction farm processors. All data processing(Raw ⇒ STA ⇒ DST ⇒ etc.) and physical location (tape or disk) information will also need to be maintained for all event data.

Not all of the databases that we used in Run I should be used in Run II. The FATMEN Catalog and the Production DataBase should be merged into one database.[2] Several of the databases were created but never utilized for Run I (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(s). Hopefully, this will lead to a more reliable and easily maintained solution. In addition all of the Rdb databases, which are VMS specific, will also need to be replaced with platform independent databases.

4. DBMS REQUIREMENTS

Requirements for a Run II DBMS can, at this point, only be scaled from our Run I experience to the more complex and higher event data rate anticipated for Run II. One DB product to serve all the Run II needs would simplify the effort required to train personnel and manage the information. Due to the complex nature of the information to be stored, it appears that an Object-Oriented DBMS is best suited to our needs. It is felt that a commercial product would provide a more reliable service and higher quality tools for access and performance monitoring than a public domain software product. A list of requirements and preferences for such a product follows.

Requirements

- ◊ Be commercially supported.
- ◊ Allow automatic replication(copying) of a database across DØ supported platforms.
- ◊ Support Object & Schema versioning.
- ◊ Meet performance requirement needs for Run II
  - On-line calibration data must be accessible no later than 5 minutes after a calibration run has finished.
  - Robust and corruption free performance
  - Others as we are told what they are.
- ◊ Performance must scale for the expected number of users and size of the database. It should take into account that the database may be replicated to help improve performance.
- ◊ Work in a fully-distributed, multi-platform environment
- ◊ Be standards compliant
  - ODMG, SQL, C++
- ◊ Must have sufficient Database administration tools to do optimization and setup of the database.
- ◊ Provide monitoring capabilities.

[2] The tape staging system was VMS specific and will also change. An interface between the new staging system and the combined FATMEN catalog/Production Database will need to be written.

Preferences

- ◊ Provide query access via the Web.
- ◊ Licensing be site-wide and include remote institutions.

An additional minimal database solution may be required if the database chosen cannot meet the performance requirements necessary for the on-line calibration and off-line reconstruction requirements. During Run I because the RDB database could not be accessed from a UNIX platform DBL3 was used to store all the calibration information. However, DBL3's reliability was not very good. The off-line DBL3 databases had regularly occurring corruption problems (and in fact still do occur). It also required 8 people to manage the DBL3 databases while all the DEC Rdb databases only required 2 people. If a minimal database solution becomes necessary, we feel that the product should also meet most of the above requirements to avoid the problems encountered in Run I. However, the product does not necessarily need to be commercial.

5. DATABASES REVIEWED

We have explored several commercial and public domain database products in the context of Run II usage. The databases reviewed can be categorized into three different types: relational (RDBMS), object-relational (ORDBMS), or object-oriented (ODBMS). A relation database is tabular in format and uses index tables to allow for one-to-many or many-to-many relationships. An object-relational database can be a true object-oriented database that can "mimic" a relational database (has SQL access) or it can be a relational database with C++ wrappers around it. An object-oriented database stores its information in object format. This means that one-to-many and many-to-many relationships are part of the object (via a pointer or object ID) instead of separated from it. The following table summarizes how each product reviewed is categorized and who supports them.

PRODUCT	SUPPORTER	PUBLIC / COMMERCIAL	TYPE
DEC Rdb	Digital Equipment Corporation	Commercial	RDBMS
Illustra	Illustra, Inc.[3]	Commercial	ORDBMS
MiniSQL	Minerva Network Management	Public Domain	RDBMS
O_2	O_2 Technology, Inc.	Commercial	ODBMS
Object Store	Object Design, Inc.	Commercial	ODBMS
Objectivity	Objectivity, Inc.	Commercial	ODBMS
OBST	Forschungszentrum Informatik	Public Domain	ODBMS
Oracle	Oracle, Inc.	Commercial	ODBMS
POSTGRES	UC - Berkeley (Database	Public Domain	ORDBMS

[3] Now a subsidiary of Informix.

	Research Group)		
SHORE	University of Wisconsin	Public Domain	ODBMS
UniSQL	UniSQL, Inc.	Commercial	ORDBMS
Versant	Versant Object Technology	Commercial	ODBMS

After careful consideration we have removed several candidates from the list. They include MSQL, SHORE, POSTGRES, DEC Rdb, Oracle, Illustra and Versant.

MSQL only allowed single key access (can only access by run or time but not both) and does not meet most of our requirements for a database commercially supported. SHORE and POSTGRES are not commercially supported products. DEC (Oracle) Rdb and Oracle are relational in nature and we feel that while they served us well in Run I an object-oriented solution will serve better (pending testing of the ODBMS solution). 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. This leaves us with the following candidates: UniSQL, Objectivity, Object Store and O_2.

- It should be noted that any of the databases removed could be considered for a minimal solution if the testing of the ODBMS shows that a minimal solution is necessary.

6. DATABASE RECOMMENDATIONS

◊ *We recommend that we minimize the kinds of databases DØ uses (2 at most if possible).*

◊ *We recommend that the database chosen meet at minimum the above DBMS requirements.*

◊ *We recommend that anyone who wishes to store data in a database should contact the Database Working Group to let them know the amount of data to be stored and the access requirements for said data.*

◊ *We recommend that a test project be defined and used to test the top choices found by this committee to answer questions such as performance issues, tool support, network access, scalability, ease of setup, etc.*

◊ *We recommend that the final candidates be further evaluated in detail by the DØ Database Working Group. However, it would be our preference to have a joint CD/CDF/DØ group formed to do the performance testing.*

◊ *We recommend a strictly OO database solution due to the complexity of the calibration database and the fact that the Production and FATMEN database will be merged and need the additional functionality. It would also be easier to work with the DØ Language Group's recommendation of C++ as the primary language.*

- **NOTE:** This recommendation is contingent upon testing of the candidates. This test should include both a high speed and normal access test. If a candidate passes both tests then a minimal database solution is not required.