CERN RD45 STATUS REPORT - A PERSISTENT OBJECT MANAGER FOR HEP

Jamie Shiers

CERN, Geneva, Switzerland

The RD45 project at CERN is investigating solutions to the problems of LHC era data management and object persistency in particular. During the first year of the project, a number of possible solutions were investigating, including home-grown and standards-conforming commercial products. During its second year, RD45 has focussed on ODMG-93 compliant ODBMS products, and Objectivity/DB in particular. We present an overview of the project, the current status and future directions. Work on the project milestones and planned future activities is discussed in detail.

1 Introduction

The RD45 project was approved in 1995 to investigate and find solutions to the problems associated with object persistency for the LHC experiments at CERN. The initial milestones, listed below, placed clear emphasis on the use of standards-conforming commercial tools, namely ODMG-compliant ODBMSs.

The milestones for the first year of the RD45 project were as follows.

(i) A requirements specification for the management of persistent objects typical of HEP data together with criteria for evaluating potential implementations.
(ii) An evaluation of the suitability of ODMG's Object Definition Language for specifying an object model describing HEP event data.
(iii) Starting from such a model, the development of a prototype using commercial ODBMSes that conform to the ODMG standard. The functionality and performance of the ODBMSes should be evaluated.

The first milestone was ammended at an interim review by the LHCC Computing Review Board (LCRB) in November 1995 to

Preprint submitted to Elsevier Preprint *22 May 1997*

– Product a “Statement of Probable Capabilities” for a HEP persistent object management system based on commercial ODBMSs and large-market mass storage systems.

One of the main successes of the first year was the identification of a model whereby commercial solutions *could* be used to store and manage HEP event data. The proposed solution was based on the use of distributed database technology, whereby a single, logical database is in fact comprised of many smaller databases, distributed over a number of servers. This solution is theoretically achievable with existing solutions, although a number of questions exist concerning performance and scalability.

These are reflected in the milestones for the second year of the project, which ended in March 1997, namely:

(i) Identify and analyse the impact of using an ODBMS for event data on the Object Model, the physical organisation of the data, coding guidelines and the use of third party class libraries,
(ii) Investigate and report on ways that Objectivity/DB features for replication, schema evolution and object versions can be used to solve data management problems typical of the HEP environment,
(iii) Make an evaluation of the effectiveness of an ODBMS and MSS as the query and access method for physics analysis. The evaluation should include performance comparisons with PAW and Ntuples.

2 Conclusions from milestone 1

Full details regarding the work performed to satisfy this milestone can be found in CERN/LHCC 97-7.

The main conclusions from this milestone are as follows:

– The use of an ODBMS to provide object persistence for HEP applications implies minimal changes to existing applications, and these changes can be further reduced by the provision of a small layer of software. A prototype version of such a layer is under development by the RD45 collaboration, and will eventually be made available through the LHC++ framework.
– With the exception that it is clearly necessary to have consistent object models - both transient and persistent, an ODBMS imposes no restrictions on the object model. The impact on physical data organisation is limited to performance - optimal data clustering will reduce redundant I/Os and result in improved performance. Similarly, the small storage overhead imposed by ODBMSs means that very small objects should be avoided. However, this

overhead is smaller than for existing, Fortran-based systems and is hence not a new constraint.
- With the exception of class libraries providing collections or containers, 3rd-party class libraries can be used freely with applications that use an ODBMS for persistence. In the case of collection/container libraries, changes must be made to avoid storing raw C++ pointers in such collections. However, ODBMS-capable versions of the principle class libraries involved are available and the ODMG is following the C++ standard in this respect.

3 Conclusions from milestone 2

The main conclusions regarding the RD45 investigations into schema evolution, object versioning and data replication are given below. It is important to point out that none of these features are currently covered by the ODMG standard. However, we continue to work with the ODMG to ensure that future versions of the standard eventually include also these features.

It is our conclusion that all of the features are clearly important techniques for solving HEP data management problems. Their implementation in ODBMS products is relatively recent, and it is clear that a number of important enhancements need to be made to existing systems in these areas.

With the possible exception of object versioning, which can be managed at the application level, it is our opinion that these features should form part of the list of mandatory requirements that must be satisfied by a HEP persistent object manager. Although versioning could be handled by the application, it would clearly be an advantage if this too was directly support by the system.

4 Conclusions from milestone 3

At the time of writing, work on this milestone had not been completed. However, it is expected that detailed results will have been made available via the Web and other mechanisms prior to CHEP 97.

In principle, storing all of the event data using a consistent scheme offers significant benefits at the analysis stage. Today's techniques of data reduction, which have evolved over many years, have been driven by necessity. The cost of random access storage relative to sequential media (tape) was so high that successive data reductions were imperative. Even at the startup of LEP, the idea of providing a mere 100GB of staging space per experiment on the central mainframe was simply unaffordable.

Today, the situation has changed dramatically, and trends suggest that this will continue into the future. Even tape media now offer some degree of random access - typically fast block addressing - and the amount of disk space that can be afforded has increased enormously.

Thus, it is important that the next generation of experiments are now constrained by the technological limits that inhibited previous ones. Nowhere is this more true than in the area of data management.

Today's experiments use a wide variety of data formats for rawdata, DSTs, Ntuples, calibration, meta-data and so forth. This gives rise to extreme difficulty in navigating e.g. from a histogram to the rawdata of the events corresponding to the entries in the histogram. Despite many man-years of effort, both centrally and within the experiments, this is still largely an unsolved problem and results in considerable inefficiency in extracting physics results from the data - in other words, a waste of extremely valuable resources.

An ODBMS approach offers the possibility of revolutionising our approach to physics analysis - offering not only more efficient access to the data, but also permitting more complicated analyses to take place.

5 Risk Analysis

Extrapolating from the current prototyping activities, which are at a scale of GB to hundreds of GB, to a production system capable of scaling to 100PB - an increase of some 6 orders of magnitude - requires detailed and careful analysis of the risks involved.

The main risks of the current strategy were identified and analysed as part of an RD45 workshop, held at CERN in November 1996. Many of these risks, including scalability and performance bottlenecks in the current Objectivity/DB product, are measurable today.

These include issues such as the support for very large collections, the limits imposed by the Objectivity implementation of the 64-bit object identifier, transparent object reclustering and so forth.

Fortunately, almost no problems were encountered during the testing of the limits of the current product. The two problems that were seen have been reported to Objectivity, and are in any case not on the critical path. These problems are:

(i) The number of containers per database is limited architecturally. Although creating new containers one at a time is correctly protected against this limit, the so-called "batch-mode", whereby many containers can be created at a time, is not correctly protected. This is suspected to be due to the mis-coding of a simple comparison,

(ii) Some performance problems were seen when creating very large numbers of physical databases in a single process. Using multiple processes avoided the problem.

A number of non-negligible risks were identified as part of this study. These are listed below in order of decreasing risk.

(i) The market for high-end mass storage systems is almost certainly too small to sustain commodity products. However, it is probable that a product such as HPSS will continue to survive, if only to meet the needs of the US National Laboratories. This is indeed an area of concern, as the man-power required to develop a HEP-specific product is clearly not available today, and cannot be expected in the future,

(ii) The ODBMS market is still extremely small compared with that of RDBMS products. The continued existence of any given ODBMS vendor cannot be guaranteed over the timescales involved for the LHC. However, given that the currently preferred product, namely Objectivity/DB, and the backup-solution, namely Versant, are both strong in the Telecoms market, there is reason to believe that at least one of these two products, or at least one product capable of satisfying the needs of the Telecoms market, will continue to exist,

(iii) The management issues of a very large, distributed, object database combined with mass storage system will be by no means trivial. Many issues related to this problem, such as "rolling-upgrades", handling of unreliable networks and so forth, need to be further investigated. However, it is clear that today's home-grown solutions do not scale in man-power requirements to the LHC, and so it is in any case necessary to find alternative, less man-power demanding, solutions. As one can expect many large, albeit not PB, distributed object databases on the timescale of the LHC, there is reason to believe that this management issues will be at least partially solved.

6 Proposed Mass Storage Interface

In the 1996 RD45 status report to the LCRB, a number of possible ways of interfacing and ODBMS to an MSS were presented. More recently, an alternative approach has been developed, that is currently specific to Objectivity/DB and HPSS, although could be adopted to other products if required.

HPSS provides a client API, which provides I/O routines such as *hpss_read(),* *hpss_open()* and so forth. These provide functionality equivalent to calls such as *read(), open()*, etc. although operate on HPSS-managed space. In particular, a call to *hpss_open()* will cause the target file to be staged to the top of its associated storage hierarchy - typically, but not necessarily, disk. This allows a clean interface to the mass storage system to be provided. All Objectivity/DB databases (or rather, all except those, such as the tag databases, which need not reside permanently on disk), would reside in the HPSS-managed storage hierarchy. As objects were accessed, Objectivity would automatically issue *open()* calls for the associated databases, which are seen as files by the operating system. This would result in the corresponding databases being automatically loaded from tape, as and when necessary.

Rather than modify the Objectivity/DB client, it would be possible to provide this interface at the level of the Objectivity server. This server is simply a page server and, as such, has no knowledge of objects. It uses a small number of I/O routines, for which HPSS equivalents exist. It also provides a clean break between the HPSS-managed world, and the Objectivity/DB-managed persistent object space. Finally, it should be possible to implement, at least at the proof-of-concept level, in a very short time.

We are currently working with both Objectivity and the HPSS consortium to ensure that both the initial proof-of-concept version, and the eventual productised version, are made available in a timely manner.

7 Future Directions

After discussions with the LHC experiments, the following milestones have been proposed to the LCB for the next year of investigations.

(i) Demonstrate, by the end of 1997, the proof of principle that an ODBMS can satisfy the key requirements of typical production scenarios (e.g. event simulation and reconstruction), for data volumes up to 1TB. The key requirements will be defined, in conjunction with the LHC experiments, as part of this work,

(ii) Demonstrate the feasibility of using an ODBMS + MSS for Central Data Recording, at data rates sufficient to support CMS test-beam activities during 1997 and NA45 during their 1998 run,

(iii) Investigate and report on the impact of using an ODBMS for event data on end-users, including issues related to private and semi-private schema and collections, in typical scenarios including simulation, (re-)reconstruction and analysis.

Furthermore, it is felt important to that a proof-of-concept MSS interface should be demonstrated by mid-1998, in order to provide input to the ATLAS and CMS Technical Design Reports. In addition, it is expected that further investigations of the areas covered by the 1996 milestones will be required. Examples include further performance investigations, a study of the feasibility of wide-area data replication, and so on.

8 Summary and Conclusions

The proposed solution to object persistence for LHC event data, based upon a commercial ODBMS and MSS, has been accepted as the baseline solution for both ATLAS and CMS, pending the final results of the RD45 investigations. The performance and scalability of such a solution, together with the associated risks, have been analysed. Whilst more work needs to be done, particularly in the area of MSS integration and efficient data access, we believe that this solution is still by far the most promising of those considered, offering both the functionality and scalability that is required.

References

[1] CERN RD45 Status Report, CHEP 1995, Rio de Janerio, Brazil.

[2] RD45 - A Persistent Object Manager for HEP, LCRB Status Report, CERN/LHCC 96-15, February 1996

[3] Object Databases and Mass Storage Systems - the Prognosis, the RD45 collaboration, CERN/LHCC 96-17

[4] RD45 Status Report to the LCB, February 1997, CERN/LHCC 97-6

[5] Object Databases and their Impact on Storage-Related Aspects of HEP Computing, the RD45 collaboration, CERN/LHCC 97-7

[6] Object Database Features and HEP Data Management, the RD45 collaboration, CERN/LHCC 97-8

[7] Using an Object Database and Mass Storage System for Physics Analysis, the RD45 collaboration, CERN/LHCC 97-9

[8] DBMS Needs Assessment for Objects, Barry and Associates

[9] The Object Database Standard, ODMG-93, Morgan Kaufmann

[10] ATLAS Computing Technical Proposal, CERN/LHCC 96-43

[11] CMS Computing Technical Proposal, CERN/LHCC 96-45