# THE EVOLUTION OF DATABASES IN HEP

Jamie Shiers, CERN, Geneva, Switzerland

## *Abstract*

The past decade has indubitably been an era of tumultuous change in the area of Computing for High Energy Physics. This paper addresses the evolution of databases in HEP, starting from the LEP era and the visions presented during the CHEP 92 panel "Databases for High Energy Physics" (D. Baden, B. Linder, R. Mount, J. Shiers [1]). It then reviews the rise and fall of Object Databases as a "one size fits all solution" in the mid to late 90's and finally summarises the more pragmatic approaches that are being taken in the final stages of preparation for LHC data taking. The various successes and failures (depending on one's viewpoint) regarding database deployment during this period are discussed, culminating in the current status of database deployment for the Worldwide LCG.

## INTRODUCTION

A panel on Databases for High Energy Physics held at CHEP ’92 in Annecy, France attempted to address two key questions, namely:

1. Should we buy or build database systems for our calibration and book-keeping needs?

2. Will database technology advance sufficiently in the next 8 to 10 years to be able to provide byte-level access to petabytes of SSC/LHC data?

In attempting to answer the first questions, two additional issues were raised, namely:

- Is it technically possible to use a commercial system?

- Would it be manageable administratively and financially?

At the time of the panel, namely in September 1992, it was pointed out that the first question had already been addressed during the period of LEP planning: what was felt to have a technical possibility in 1984 had become at least a probability by 1992, although the issues related to licensing and support were certainly still significant.

In this paper we follow the evolution of the use of Databases in High Energy Physics between two CHEPs – in Annecy and Mumbai – and then revisit these questions in the pre-LHC era.

## CALIBRATION AND BOOK-KEEPING

In 1992, two common projects that attempting to address general purpose detector calibrations (“conditions”) and book-keeping / file catalog needs were the two CERN Program Library packages *HEPDB* and *FATMEN*. At a high-level, these two packages had a fair degree of commonality: both were built on top of the ZEBRA RZ system, whilst using ZEBRA FZ for exchanging updates between client and server (and indeed between servers). Both implemented a Unix file-system like interface – and indeed shared a reasonable amount of code.

Indeed, one of the arguments at the time was that the amount of code – some tens of thousands of lines – would be more or less the same even if an underlying database management system was used. Furthermore, it was argued that the amount of expert manpower required at sites to manage a service based on a DBMS was higher – and more specialized – than that required for in-house developed solutions.

The ZEBRA RZ package had a number of restrictions: firstly, the file format used was platform dependent and hence could not easily be shared between different systems (e.g. using NFS) nor transferred using standard ftp. This restriction was removed by implementing “exchange file format”, in analogy with the ZEBRA FZ package (Burhardt Holl, OPAL). In addition and in what turned out to be a disturbingly recurrent theme, it also used 16-bit fields for some pointers, thereby limiting the scalability of the package. ZEBRA RZ was improved to use 32-bit fields( Sunanda Banerjee, TIFR and L3), allowing for much large file catalogs and calibration files, as successfully used in production, for example by the FNAL D0 experiment.

## CHEP ’92 AND THE BIRTH OF OO PROJECTS

For many people, CHEP ’92 marks the turning point away from home-grown solutions, which certainly served us extremely well for many years, towards “industry standards”. In the case of programming languages, this meant away from “HEP Fortran” together with powerful extensions provided by Zebra and other memory and data management packages, to C++, Java and others. This has certainly not been a smooth change – many “truths” had to be unlearnt, sometimes to be re-learnt, and a significant amount of retraining was also required.

Notably, CERN launched the RD41 “MOOSE” project, to evaluate the suitability of Object Orientation for common offline tasks associated with HEP computing, RD44, to re-engineer the widely-used GEANT detector simulation package, RD45 to study the feasibility of Object-Oriented Databases (ODBMS) for handling physics data (and not just conditions / file catalog / event meta-data), LHC++ (a CERNLIB functional replacement in C++) and of course ROOT.

With the perfect 20-20 vision that hindsight affords us, one cannot help but notice the change in fortunes these various projects have experienced. At least in part, in the author’s view, there are lessons here to be learnt for the future, which are covered in the summary.

## THE RISE AND FALL OF OBJECT DATABASES

This is well documented in the annals of HEP computing – namely the proceedings of the various CHEP conferences over the past decade or so. Object Databases were studied as part of the PASS project, focusing on the SSC experiments. The CERN RD45 project, approved in 1995, carried on this work, focusing primarily on the LHC experiments, but also pre-LHC experiments with similar scale and needs. At the time of writing their use in HEP for physics data is now history, although some small applications – such as the BaBar conditions DB – still remain. To some extent their legacy lives on: the POOL project builds not only on the success of ROOT, but also on the experience gained through the production deployment of Object Databases at the petabyte scale – successes and short-comings – as well as the risk analysis proof-of-concept prototype “Espresso”, described in more detail below.

## RD45 – THE BACKGROUND

Of the various OO projects kicked off in the mid-90’s, the RD45 project was tasked with understanding how large-scale persistency could be achieved in the brave new world. At that time, important bodies to be considered were the Object Management Group (OMG), as well as the similarly named Object Data(base) Management Group. The latter was a consortium of Object Database vendors with a small number of technical experts and end-users – including CERN. Whilst attempting to achieve application-level compatibility between the various ODBMS implementations – i.e. an application that worked against an ODMG compliant database could be ported to another by a simple re-compile – it had some less formal, but possibly more useful (had they been fully achieved) goals:

- That the Object Query Language (OQL) be compliant with the SQL3 DML;
- That no language extensions (thinking of C++ in particular) would be required for DDL.

ODMG-compliant implementations were provided by a number of vendors. However, as was the case also with relational databases, there are many other issues involved in migrating real-world applications from one system to another than the API.

## RD45 – MILESTONES

There is a danger when reviewing a past project to rewrite – or at least re-interpret – history. To avoid this, the various milestones of the RD45 project and the comments received from the referees at the time are listed below.

- [The project ] should be approved for an initial period of one year. The following milestones should be reached by the end of the 1st year.

  **1.** A requirements specification for the management of persistent objects typical of HEP data together with criteria for evaluating potential implementations. **[ Later dropped – experiments far from ready ]**

  2. An evaluation of the suitability of ODMG's Object Definition Language for specifying an object model describing HEP event data.

  3. 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.

- It should be noted that the milestones concentrate on **event data.** Studies or prototypes based on other HEP data should not be excluded, especially if they are valuable to gain experience in the initial months.

The initial steps taken by the project were to contact the main Object Database vendors of the time – $O_2$, ObjectStore, Objectivity, Versant, Poet – and schedule presentations (in the case of $O_2$, Objectivity also training). This lead to an initial selection of the two latter products for prototyping, which rapidly led to the decision to continue only with Objectivity – the architecture of $O_2$ being insufficiently scalable for our needs. Later in the project, Versant was identified as a potential fallback solution to Objectivity, having similar scalability – both products using a 64 bit Object Identifier (OID). Here again we ran into a familiar problem – Objectivity’s 64 bit OID was divided into 4 16 bit fields, giving similar scalability problems to those encountered a generation earlier with ZEBRA RZ. Although an extended OID was requested, it was never delivered in a production release – which certainly contributed to the demise of this potential solution.

The milestones for the 2nd year of the project were as follows:

1. 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;

2. 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;

3. 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.

These were followed, for the third year, with the following:

1. 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,

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

3. 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.

Finally, the milestones for 1998 were:

1. Provide, together with the IT/PDP group, production data management services based on Objectivity/DB and HPSS with sufficient capacity to solve the requirements of ATLAS and CMS test beam and simulation needs, COMPASS and NA45 tests for their '99 data taking runs.

2. Develop and provide appropriate database administration tools, (meta-)data browsers and data import/export facilities, as required for (1).

3. Develop and provide production versions of the HepOODBMS class libraries, including reference and end-user guides.

4. Continue R&D, based on input and use cases from the LHC collaborations to produce results in time for the next versions of the collaborations' Computing Technical Proposals (end 1999).

## RD45 – RISK ANALYSIS

The CMS Computing Technical Proposal, section 3.2, page 22), contains the following statement:

*"If the ODBMS industry flourishes it is very likely that by 2005 CMS will be able to obtain products, embodying thousands of man-years of work, that are well matched to its worldwide data management and access needs. The cost of such products to CMS will be equivalent to at most a few man-years. We believe that the ODBMS industry and the corresponding market are likely to flourish. However, if this is not the case, a decision will have to be made in approximately the year 2000 to devote some tens of man-years of effort to the development of a less satisfactory data management system for the LHC experiments."*

As by now is well known, the industry did not flourish, so alternative solutions had to be studied. One of these was the Espresso proof-of-concept prototype, built to answer the following questions from RD45's Risk Analysis:

- Could we build an alternative to Objectivity/DB?
- How much manpower would be required?
- Can we overcome limitations of Objectivity's current architecture?
- To test / validate import architectural choices.

The Espresso proof-of-concept prototype was delivered, implementing an ODMG compliant C++ binding. Various components of the LHC++ suite was ported to this prototype and an estimate of the manpower needed to build a fully functional system made.

The conclusions of an IT Programme of work retreat on the results of this exercise were as follows:

- Large volume event data storage and retrieval is a complex problem that the particle physics community has had to face for decades.
- The LHC data presents a particularly acute problem in the cataloguing and sparse retrieval domains, as the number of recorded events is very large and the signal to background ratios are very small. All currently proposed solutions involve the use of a database in one way or another.
- A satisfactory solution has been developed over the last years based on a modular interface complying with the ODMG standard, including C++ binding, and the Objectivity/DB object database product.

- The pure object database market has not had strong growth and the user and provider communities have expressed concerns. The “Espresso” software design and partial implementation, performed by the RD-45 collaboration, has provided an estimate of 15 person-years of qualified software engineers for development of an adequate solution using the same modular interface. This activity has completed, resulting in the recent snapshot release of the Espresso proof-of-concept prototype. No further development or support of this prototype is foreseen by DB group.
- Major relational database vendors have announced support for Object-Relational databases, including C++ bindings.
- Potentially this could fulfil the requirements for physics data persistency using a mainstream product from an established company.
- CERN already runs a large Oracle relational database service.

This was accompanied by the following recommendation:

- The conclusion of the Espresso project, that a HEP-developed object database solution for the storage of event data would require more resources than available, should be announced to the user community.
- The possibility of a joint project between Oracle and CERN should be explored to allow participation in the Oracle 9i beta test with the goals of evaluating this product as a potential fallback solution and providing timely feedback on physics-style requirements. Non-staff human resources should be identified such that there is no impact on current production services for Oracle and Objectivity.

## ODBMS IN RETROSPECT

It would be easy to dismiss Object Databases as a simple mistake. However, their usage was relatively widespread for close to a decade (CERN and SLAC in particular). Was there something wrong in the basic technology? If not, why did they not “take off”, as so enthusiastically predicted?

Both of the two laboratories cited above stored around 1PB of physics data in an ODBMS, which by any standards has to be a success. There were certainly limitations – which is something to be expected. The fact that the current persistency solutions for all LHC experiments (which differs in some important respects in detail) have much in common with the ODBMS dream – and less with those of the LEP era deserves some reflection.

There was certainly some naïvety concerning data and persistency models – the purist ODBMS view was that there was one. As a re-learnt lesson, RD45 pointed out very early that this was often not viable. More importantly, the fact that the market did not take off meant that there was no serious ODBMS vendor – together with a range of contenders – with which to entrust LHC data.

## ORACLE FOR PHYSICS DATA

Based on Oracle’s 9i and later 10G release, the feasibility of using Oracle to handle LHC-era physics data was studied. This included the overall scalability of the system – where once again 16 bit fields raised their ugly heads (since fixed) – as well as the functionality and performance of Oracle’s C++ binding “OCCI”. As a consequence of this work, the COMPASS event data was migrated out of Objectivity into flat files for the bulk data together with Oracle for the event headers – of potential relevance to LHC as this demonstrated the feasibility of multi-TB databases – similar to what would be required to handle event tags for LHC data.

However, the strategy for all LHC experiments is now to stream their data into ROOT files, with POOL adopted as an additional layer by all except ALICE.

In parallel, the database services for detector related and book-keeping applications – later also Grid middleware and storage management services – were re-engineered so as to cope with the requirements of LHC computing. A significant change in this respect was the move away from Solaris for database servers to Linux on PC hardware. Initial experience with the various PC-based systems at CERN showed that the tight coupling between storage and CPU power inherent in a single box solution was inappropriate and a move to SAN-based solutions, which allow storage and / or processing power to be added as required, has since been undertaken.

## THOSE QUESTIONS REVISITED

After more than a decade it seems that the questions posed at CHEP ’92 still have some relevance. Today, it is common practice that applications in the area of storage management, experiment book-keeping and detector construction / calibration use a database backend. However, the emergence of open-source solutions and indeed much experience has changed the equation. Nowadays, it is common practice to use a database backend (where the distinction between object / object-relational / pure-relational is very much blurred). However, the licensing, support and deployment issues are still real.

So in summary:

1. Should we buy or build database systems for our calibration and book-keeping needs?

- It now seems to be accepted that we ***build*** our calibration & book-keeping systems ***on top*** of a database system.
- ***Both*** commercial and open-source databases are supported.

2. Will database technology advance sufficiently in the next 8 to 10 years to be able to provide byte-level access to petabytes of SSC/LHC data?

   - We (HEP) have run production database services up to the PB level. The issues related to licensing, and – perhaps more importantly – ***support***, to cover the full range of institutes participating in an LHC experiment, remain.
   - Risk analysis suggests a more cautious – and conservative – approach, such as that currently adopted. (Who are ***today*** the concrete alternatives to the market leader?)

As regards lessons for the future, some consideration of the evolution of the various OO projects – RD45, LHC++ and ROOT – is deserved. One of the notable differentiators of these projects is that the former were subject to strict and frequent review. Given that the whole field was very new to the entire HEP community, some additional flexibility and freedom to adjust to the evolving needs – and indeed our understanding of a new technology – would have been valuable.

As we now deploy yet another new technology for LHC production purposes, we are possibly falling into the same trap.

Food for thought for CHEP '30 or thereabouts?

## ACKNOWLEDGEMENTS

Many people have contributed to the story of Databases in HEP, including the many who worked on various aspects of the CERN Program Library and the ZEBRA, FATMEN and HEPDB packages. Members of the PASS and RD45 projects, together with the ROOT and POOL and related projects as well as all those who have contributed to database deployment at the various HEP sites throughout the years also played key roles in this story.

## REFERENCES

[1] Computing in High Energy Physics '92. CERN 92-07, D. Baden, B. Linder, R. Mount, J. Shiers.