DATABASES IN HIGH ENERGY PHYSICS – A CRITICAL REVIEW

Jamie Shiers, CERN, Geneva, Switzerland

Abstract

The year 2000 is marked by a plethora of significant milestones in the history of High Energy Physics. Not only the true numerical end to the second millennium, this watershed year saw the final run of CERN's Large Electron-Positron collider (LEP) – the world-class machine that had been the focus of the lives of many of us for such a long time. It is also closely related to the subject of this chapter in the following respects:

- Classified as a nuclear installation, information on the LEP machine must be retained indefinitely. This represents a challenge to the database community that is almost beyond discussion – archiving of data for a relatively small number of years is indeed feasible, but retaining it for centuries, millennia or more is a very different issue;
- There are strong scientific arguments as to why the data from the LEP machine should be retained for a short period. However, the complexity of the data itself, the associated metadata and the programs that manipulate it make even this a huge challenge;
- The story of databases in HEP is closely linked to that of LEP itself: what were the basic requirements that were identified in the early years of LEP preparation? How well have these been satisfied? What are the remaining issues and key messages?
- Finally, the year 2000 also marked the entry of Grid architectures into the central stage of HEP computing. How has the Grid affected the requirements on databases or the manner in which they are deployed? Furthermore, as the LEP tunnel and even parts of the detectors that it housed are readied for re-use for the Large Hadron Collider (LHC), how have our requirements on databases evolved at this new scale of computing?

A number of the key players in the field of databases – as can be seen from the author list of the various publications – have since retired from the field or else this world. Given the fallibility of human memory, the need for a record of the use of databases for physics data processing is clearly needed before memories fade completely and the story is lost forever. It is necessarily somewhat CERN-centric, although effort has been made to cover important developments and events elsewhere. Frequent reference is made to the Computing in High Energy Physics (CHEP) conference series – the most accessible and consistent record of this field.

INTRODUCTION

This chapter traces the history of databases in HEP over the past quarter century. It does not attempt to describe in detail all database applications, focusing primarily on their use related to physics data processing. In particular, the use of databases in the accelerator sector, as well as for administrative applications – extensively used by today's large-scale collaborations – are only covered in passing. However, the famous LEP Database Service – "LEP DB" certainly deserves a mention. Quoting from "LEP Data Base Information note number 1:"

"Oracle version 2 was installed at CERN in the summer of 1981, on a VAX system running VMS version 2. A pre-release of version 3 is presently under test and a production version is expected before the end of the year."

The LEP DB service led to the installation of the first VAX 11/780 into CERN's Computer Centre. This marked another significant change in HEP computing (at least at CERN!), as it marked an important change from batch-dominated computing: the strengths of VAX computing were its interactivity, its excellent (for the time) debugger and its well-integrated networking support. Although it was for the IBM VM/CMS system to introduce the concept of 'service machines', the impact of these changes can still be seen today. Computing for the LEP and LHC experiments was / is largely based on services – experiment-specific or otherwise – of much higher level than the basic batch system and / or tape staging system – a trend that is strongly linked to a database-backend to maintain state, coupled to the rapid developments in computing power that allowed the necessary servers to be setup. A further significant event that occurred around the same time was the introduction of the first Unix system at CERN. Although reference has often been made to early highly conservative estimates of the growth of the Unix installed base, no one at that time predicated that it would soon dominate HEP computing – as it continues to do in its Linux guise today – and let alone on commodity PCs. Indeed, the reluctance to move to Unix – although relatively short-lived – gave a foretaste of the immense and lingering resistance to the demise of Fortran.

The rise of Linux on Intel-compatible platforms has also had a significant impact on database services. After the early popularity of VAX-based systems, Solaris was long the platform of choice (at least for Oracle – the DBMS deployed at CERN). Solaris was displaced by Linux / Intel in recent years and has allowed database services to keep up with at least some of the demand. Not only has the number of database servers or clusters

increased significantly, but also the volume of data thus managed. The great Jim Gray often referred to the "management limit" – somewhere in the low to medium multi-TB region. Whilst only one measure of management complexity – and no one with Jim's great depth of insight would ever have meant otherwise – things clearly cannot scale indefinitely, even given the write-once, read-rarely nature of our bulk data. Early proposals (see below) called for solutions that required much less than one person per experiment for support. The required support level has clearly long passed this threshold, perhaps normal given the scale of HEP experiments in the LHC era. However, alarm bells should possibly be ringing. Are the proposed solutions compatible with the manpower resources that will be available to support them?

Finally, in addition to the core applications identified over 25 years ago, Grid computing has brought new requirements to the database arena – a large number of key Grid applications, such as the reliable File Transfer Service and storage services, are dependent on back-end databases. In reviewing the evolution of Databases in HEP during a quarter century of change, we try to establish the key discontinuities and to answer the many questions that have been raised.

ECFA STUDY

In the early 1980s, the European Committee for Future Accelerators (ECFA), launched a number of study groups into various aspects of HEP computing. One of these groups – subgroup 11 – reported [1] on "Databases and Book-keeping for HEP experiments". The goals of this working group were as follows:

- To provide a guide to the database and bookkeeping packages used at present by HEP groups;
- To find out what future requirements (would) be;
- To make recommendations as to how these (could) best be met.

The working group used the following definition of a database:

"A collection of stored operational data used by the application system of some particular enterprise."

It then goes on to explain:

"In the HEP context, the word 'database' is sometimes used to refer to the totality of data associated with a single experiment… We shall not use the word with that meaning… Instead, we shall use the word for more highly organised subsets of data such as

- *Catalogues of experimental data (with information such as run type, energy, date, trigger requirements, luminosity and detector status);*
- *Information on the status of the analysis (e.g. input and output tapes, cut values and pass rates);*
- *Calibration data;*
- *Summary information from the analysis (e.g. histograms and fitted parameters)."*

Detail aside, such a definition would be instantly recognisable to a physicist of today.

The report also clarifies:

"It is further necessary to distinguish between:

a) Database systems developed within the HEP community, sometimes for a single experiment, which are referred to as 'HEP databases' or 'simple databases';
b) Database management systems (DBMS), which may be classified as hierarchical, network or relational in structure."

Finally, it records that, with very few exceptions, DBMS were not used by HEP experiments at that time.

The report continues with a long list of detailed requirements and surveys of packages in use at that time. We nevertheless include the summary of recommendations made by the working group:

1. *There would be many advantages in using commercially available DBMSs in HEP to reduce the amount of work required to obtain a database or bookkeeping system tailored to the needs of a particular experiment. They will clearly have a place in HEP computing in the future and should be used for LEP experiments in place of complex user-written systems;*
2. *The requirements of flexibility and ease of use clearly point to the need for a relational DBMS;*
3. *Standardisation at the SQL interface level is suggested both for interactive terminal use and embedded in FORTRAN programs. This is an alternative to the implementation of a common DBMS at all centres of HEP computing;*
4. *Greater awareness is needed within the HEP community of what DBMSs offer. Pilot projects should be set up so that some experience can be obtained as soon as possible;*
5. *There is an immediate need for the major HEP computing centres, especially CERN, to make suitable relational DBMSs (e.g. SQL/DS or Oracle) available to users;*
6. *Simple HEP database packages will continue to be needed, especially in the short term. The KAPACK [2] system is recommended for this purpose. However, the basic KAPACK package should not be extended significantly. (If a much more sophisticated system is needed, then a DBMS should be used.);*

7. *A simple bookkeeping system could be written using KAPACK and supported in the same manner as KAPACK;*
8. *Users developing higher level software of a general nature on top of KAPACK or a DBMS should be urged to write as much as possible in the form of a standard add-on packages which can be used by other groups. Central support for such packages should be offered as an inventive to do this;*
9. *Before the development of very sophisticated or complicated packages is undertaken for a given experiment, careful consideration should be given as to whether the advantages to be obtained will justify the work involved. (Considerable effort has been expended in the past in providing facilities that would be standard with a DBMS.);*
10. *A greater degree of automation in the management of tape data would be desirable. If, as at DESY, users do not normally have to worry about tape serial numbers, the need for user tape handling packages is obviated and the problems of bookkeeping are considerable simplified.*

The report also noted that DBMSs and data structure management packages were closely related – a fact borne out by many of the database-like developments for LEP, as we shall see later.

THE CENTRAL ORACLE SERVICE AT CERN

Following the recommendations of the ECFA report, and building on the experience gained with the Oracle service for the LEP construction project, a proposal to establish a central Oracle service on the CERNVM system was made in early 1984 – just a few months after the publication of the report.

Although, from today's point of view, the choice of Oracle appears almost automatic, things were much less obvious at that time. For example, the evaluation of replies to the 1982 LEP relational database enquiry – initially sent out to over 30 firms – resulted in only 6 replies that were considered to be relational systems. Of these, only two (SQL/DS and Oracle) were further considered, although SQL/DS had not yet been delivered to a customer. Furthermore, it only ran under DOS and would have required an additional system to support it. Oracle, on the other hand, was installed at over 70 sites, including 4 in Switzerland!

From such humble beginnings, the service has continued to grow with the years, with physics applications representing a relatively small fraction of the overall service, until the central cluster was logically separated into two in the early 2000's. At this time, a 2-node cluster running Solaris was established – using recycled Sun nodes and a small disk array – to host physics applications, being rapidly complemented by experiment-specific servers built on stovepipe systems, namely "CERN disk-servers". The latter was never an optimal solution and following a lengthy study into Oracle's RAC architecture and its use on Intel systems with SAN storage, such a solution has now been adopted. Numerous additional database servers hosted applications related to the accelerator, experiment controls and AIS / CIS applications, but these are not the main thrust of this chapter.

DATABASE SYSTEMS FOR HEP EXPERIMENTS

In 1987, a review of database systems in HEP [3], primarily but not exclusively within the context of the L3 collaboration, evaluated a variety of database systems and described the L3 database system [4] (later DBL3), then under construction. The systems considered – Oracle, SQL/DS, Ingres, KAPACK and ZEBRA RZ [5] – were evaluated on the basis of the following criteria:

- *Full features;*
- *Efficiency;*
- *Fortran access;*
- *Terminal access;*
- *Concurrent writes;*
- *Portability of Fortran;*
- *Portability of Data;*
- *Robustness;*
- *Security;*
- *Cheapness.*

None of the systems excelled in all categories, although the commercial systems fared best in their feature set and clearly worst in terms of cost. Based not only on these criteria, but also performance measurements, the choice narrowed rapidly to Oracle, RZ or KAPACK – the latter two being part of the CERN Program Library. Given the more extensive feature set of RZ over KAPACK, this left only Oracle and RZ. However, at that time it was not considered realistic to require all institutes that were part of the L3 collaboration to acquire an Oracle license – an issue that has reappeared and been re-evaluated at regular intervals over the past 2 decades. Despite significant advances on this front, the requirement for all institutes in a HEP collaboration to acquire commercial licenses – and not just a strictly limited subset – is still as high a hurdle today as it was 20 years ago.

Thus, the DBL3 package was built using the ZEBRA RZ system – and ZEBRA FZ for the exchange of updates. A system with largely similar functionality – also built on ZEBRA RZ / FZ – was later developed by OPAL (the OPCAL system), whereas DELPHI had already developed a KAPACK-based solution. The ALEPH book-keeping and ADAMO systems are described in more detail below.

Whilst today's computing environment is clearly highly complex, it is worth emphasising that that of LEP startup was, for its time, equally challenging. The degree of heterogeneity – of compilers, operating systems and

hardware platforms – was much greater. Networking was still primitive and affordable bandwidths only a trickle by today's standards. Just as today, every drop of ingenuity was required to squeeze out adequate resources and functionality – requirements that continue to maintain HEP computing ahead of the wave.

COMPUTING AT CERN IN THE 1990S

In July 1989, the so-called 'green book' [6] on LEP computing was published. Amongst the many observations and recommendations made by this report – including spotting the clear trend to distributed computing and the potential use of workstations in this respect (a foretaste perhaps of the SHIFT project), it contained a chapter on Data Base systems. (Historically, the use of "database" as a single word was already common in the previous decade). The book was published simultaneously with the commissioning of the LEP machine and thus by definition covered most of the production systems deployed by the LEP experiments. By that time a central Oracle service – as opposed to the dedicated LEP DB service which continued to run on VAX hardware – had been setup on the central IBM systems. Moreover, two new packages had entered the scene which were set to influence LEP computing significantly. These were the ZEBRA data structure management package – which can somewhat naively be thought of as combining the strengths of the HYDRA and ZBOOK packages before it – and the Aleph Data Model (ADAMO) [7] system. The ADAMO system is particularly notable in that it brought the use of entity-relationship modelling to the mainstream in HEP computing.

The report presents a rather thorough analysis of the areas where database applications were in use, or where the use of such technology would make sense. The list included the following:

- Collaboration address lists;
- Electronic mail addresses;
- Experiment bookkeeping;
- Online databases;
- Detector geometry description databases;
- Calibration constants;
- Event data;
- Bookkeeping of program versions;
- Histograms and other physics results;
- Software documentation;
- Publication lists;
- Other applications.

Specific recommendations were made in a number of these areas, as described below:

Education and training:

"An effort should be made to make physicists in experiments more aware of the potentialities of commercial DBMS for their applications. This could be achieved by intensifying training in the area of data models (software engineering) and DBMS."

Design Support Team:

"Manpower should be made available to support centrally the experiments, starting with the design of the database and continuing during the whole life cycle, including the implementation of the application dependent code. This support team should also ensure the long term maintenance of the General Purpose applications described below."

Data Model Software:

"A package should be provided to design interactively a Data Model and to store the definition in the form of a dictionary in ZEBRA files. The Entity-Relationship Model and related software from ADAMO should be considered as a first step in this direction. This would allow to profit from the experience and possibly from existing tools, including commercial ones."

Portability of Database Information:

"A package should be provided to data from Oracle to a ZEBRA (RZ) data structure. The reverse could also be implemented, providing a data model describes the structure of the data in the DB. A decent user interface should be written on top of these files to allow the users to inquire about the information contained in this structure and to update it. Tools provided with the ADAMO package could be used to learn from the existing experience and could possibly be used directly as part of the proposed package."

Experiment Administrative Databases:

"A data base should be set up covering all CERN (or HEP?) users and other people related to experiments. It should link with information and existing data bases. It should include the functionality required for experiment mailing lists and experiment specific data. Control of the data, i.e. entering and updating the information, should stay within the experiment concerned. We further recommend that a study be made on existing tools and their performance, in order to coordinate any future efforts, such as those that are being made around NADIR and EMDIR. The functionality should cover at least the one of the NADIR and AMSEND systems."

Bookkeeping Databases:

"A solution should be researched and developed urgently, in common between LEP experiments, in the area of tape bookkeeping, to avoid duplication of effort."

Documentation Databases:

"The redesign of existing documentation data bases (CERNDOC, HEPPI, ISIS etc.) into a common data base system (e.g. Oracle) should be envisaged."

Detector description / Calibration Constants Databases:

"This is probably the areas with the largest investment of manpower and the largest savings if a common solution could be found…"

Interactive Data Analysis Databases:

"PAW datasets are expected to play this role…"

ALEPH SCANBOOK

The ALEPH bookkeeping system SCANBOOK [8] was developed starting in 1988. Originally based on CERNVM, it was re-written a number of times, most recently in 1999. It is now implemented using an Oracle database using a web interface written in Java. It is the basis of the LHCb book-keeping system.

Quoting from the abstract of a presentation by ALEPH to LHCb in 2000:

"The Scanbook program has been used extensively over the last 10 years to access Aleph data (Monte Carlo and real data). It enables the users to build a list of tapes suitable for input into the Aleph analysis framework, based on parameters relevant for a given type of analysis.

Selection criteria like year of datataking, detector condition, LEP energy etc... can be combined and transformed into a set of "data cards".

The latest version is based on an Oracle database, a set of stored procedures which perform the selections, and a user interface written in Java."

FILE AND TAPE MANAGEMENT (EXPERIMENTAL NEEDS)

Given all the discussion above, the situation was ripe for a more formal study into the needs of the LEP experiments for bookkeeping and data cataloguing and a possible common solution. Initiated by a discussion in the LEP computing coordination meeting, MEDDLE, a working group was setup in early 1989. This task force, which had the unfortunate acronym FATMEN (for File And Tape Management: Experimental Needs) had the following mandate:

"At the MEDDLE Meeting held on 6/12/88 it was decided that a small task force be performed to review with some urgency the needs of various experiments for a file and tape management system to be available from LEP start-up. The following is the proposed mandate of this task force.

1. *The composition will be one representative from each major collider experiment (4 LEP, 2 UA), one representative for LEAR, one for the SPS fixed target programme, and 3-4 representatives from DD Division. You should feel free to seek advice and assistance from other experts as appropriate.*
2. *The task force will endeavour to specify the needs of the experimental program in the area of automatic tape and file management systems. It is suggested that three levels be specified:*
   - *Minimal and absolutely urgent requirements: solution needed by September 1989;*
   - *Minimal (less urgent) longer term requirements: solution needed by March 1990;*
   - *Optimal (perhaps too deluxe?) specifications of what we would really like, but which may not be available on a desirable timescale.*
3. *The task force will review the approach of the experiments to*
   - *The generation of production jobs;*
   - *The location of events at all stages of production;*
   - *The location of magnetic cartridges, both at CERN and outside*

   and make any recommendations that seem useful to avoid unnecessary duplication of effort. Transportability between operating systems, between sites, and between experiments should be considered.
4. *In view of the short timescale before LEP data starts flowing, and the limited resources available, the task force is encouraged to look very seriously at basing the overall approached on a commercially available storage management package. If that proves unrealistic the task force should take all possible steps to encourage common development between the experiments.*
5. *Taking into account the probably diversity of tape management software that is likely to be installed at LEP processing centres (not all of which work exclusively for HEP), the task force should make recommendations for interfaces to be respected.*
6. *The committee is asked to reports its conclusions to MEDDLE at its meeting scheduled for 4th April 1989."*

The most concrete outcomes of the report – dated April 6, 1989 – were as follows:

- A Tape Management System (TMS), based on SQL/DS, was imported from RAL. This system was also deployed at other HEP sites, such as IN2P3. The CERN version was later ported to Oracle – a non-trivial task, considering that parts of the RAL original were written in IBM 370 assembler with embedded SQL. Oracle had no plans for a pre-compiler for this language, nor was one ever produced. The TMS lived for many years, eventually being replaced by the volume manager component of today's CASTOR(2);
- A File Catalogue, based on ZEBRA RZ, and introducing the so-called *generic name* –

equivalent to today's logical name – was written. This had both command-line and Fortran callable interfaces, hiding much of the underlying complexity and operating system specifics.

Despite such a late start to this project, a first pre-alpha release was made during the summer of 1989 for VM/CMS only. This allowed users to perform basic catalogue manipulations and access (i.e. stage-in) catalogued files. How was it possible to produce even an alpha version so rapidly? (An initial release, covering also VMS and Unix systems, was made in time for the MEDDLE meeting of October that year, although it was still several years before the full functionality was provided – partly due to the ever changing environment at that time, including the migration from mainframes to SHIFT). This was no doubt partly due to the mature and extensive CERN Program Library but also to the excellent and fertile working environment that existing at that time. Young programmers could discuss on a peer basis with veritable giants of HEP computing and rapidly assimilate years of experience and knowledge by adopting a widely-used programming style, as well as debugging and testing techniques and an informal documentation process. This allowed a 'jump-start' in proficiency and highlights the value of mixing experienced and less experienced developers in the same teams. A further concrete step in this case was an informal code review by a very experienced developer – Hans Grote – who highlighted key issues at an early stage. This practice would surely be equally valid in today's complex world of the Grid.

Once again, the considerable heterogeneity of the early LEP computing environment has perhaps been forgotten. A simple program allowed a user to forget operating system and staging system details and access data, be it disk or tape resident, in a uniform many across a host of incompatible platforms. Three main platforms (VM/CMS, VMS and Unix, in all its many flavours, as well as also MVS) were supported, together with many times as many incompatible variants. The need for a standard and consistent interface to storage lives on today, albeit in a rather different guise.

As a file catalogue, the FATMEN package [9] of the CERN Program library was used by DELPHI, L3 and OPAL (ALEPH having their own SCANBOOK package), as well as many other experiments outside CERN (notably at DESY and FNAL). The CERN based server was only closed down in April 2007, with read-only use continuing only from OPAL. At both DESY and FNAL there was strong collaboration between the CERN and local teams – integrating with DESY's FPACK system and D0/CDF's computing environments respectively. The latter involved multi-laboratory collaboration, with the STK robot control software for VMS systems coming from SLAC.

Originally, FATMEN supported both Oracle and RZ back-ends, although the Oracle version was later dropped, for reasons discussed under the CHEP '92 section below.

The way that users were able to update the FATMEN catalogue and the techniques used for distributing updates between sites was extremely similar to that adopted by other packages, such as DBL3 (and hence HEPDB), and OPCAL, and is discussed in more detail below.

Some 2 million entries from all catalogues at CERN were used relatively recently to stress test the European DataGrid "Replica Location Service" catalogue.

Given the Oracle backend, the package attracted quite some interest from Oracle corporation, which led in turn to regular visits to their headquarters to argue for product enhancements – such as those delivered with Oracle 10g – for the HEP community. One of the first such proposals was for a distributed lock manager – now a key feature behind Oracle's Real Application Cluster architecture.

The FATMEN report also recommended that mass storage systems built according to the IEEE Computer Society's reference model be studied. Indeed, several such systems are used today in production – notably HPSS at BNL, IN2P3 and SLAC and OSM at DESY and Thomas Jefferson lab. The CASTOR system is also based on this model.

Originally designed to handle disk or tape resident files – the latter by invoking the appropriate staging system or requesting direct access to a mounted volume – the package was extended to support 'exotic opens', whereby the underlying system – such as those mentioned above – hid the gory details of file recall or equivalent operations. This was done using a syntax eerily similar to today's storage URL (SURL) – namely *protocol:path*.

The system proved extremely stable over many years, although younger 'administrators' preferred the technique of dumping the entire catalogue and manipulating it with their preferred scripting language – by far from the most efficient mechanism but one that is echoed today with the LCG File Catalogue (LFC), as is described later. The final 'change' to the system was to relink one of the utility programs (which made a backup of RZ catalogues) that had been omitted from regular rebuilds as part of CERNLIB and was hence not Y2K safe.

CHEOPS

The computer centre at CERN boasts a large satellite dish on the roof, marking one of several attempts to distribute scientific data by such means. Requests to transfer files – aka today's FTS – could be made through the FATMEN API or CLI to the CHEOPS system – a batch data dissemination system based on the OLYMPUS satellite.

CHEOPS was a collaboration between CERN, LIP and INESC in Portugal, SEFT in Helsinki and four Greek institutes – the uplink station being in Athens. The CHEOPS earth stations had access to the Olympus satellite on an overnight schedule, each site having a local Unix management server.

It entered operation early in 1992, but was destined to be somewhat short-lived. Unfortunately, after an earlier incident due to operator error was recovered, the satellite was silenced forever in a freak meteorite shower.

DATA STRUCTURES FOR PARTICLE PHYSICS EXPERIMENTS

A workshop held in Erice, Sicily in November 1990 – the 14th Eloisatron project workshop – covered many of the data structure / data base managers in HEP at that time. It included not only position papers from the authors of the various systems, but also experience papers from the user community. In addition, future directions and the potential impact of new programming languages were hotly debated. Quoting from the proceedings [10]:

"The primary purpose of the Workshop was to compare practical experience with different data management solutions in the area of:

- *Simulation of Interactions and their Detection;*
- *Data Acquisition, On Line Management;*
- *Description of Detector and Other Equipment;*
- *Experiment and Data Processing Bookkeeping;*
- *Reconstruction Algorithms;*
- *Event Display and Statistical Data Analysis."*

One paper at this workshop described *"A ZEBRA Bank Documentation and Display System"*, known as DZDOC. This was an initiative of Otto Schaile, then of the OPAL collaboration, and consisted of

"a program package which allows to document and display ZEBRA bank structures. The documentation is made available in various printed and graphical formats and is directly accessible in interactive sessions on workstations. FORTRAN code may be produced from the documentation which helps to keep documentation and code consistent."

Another idea that (re-)arose during this workshop was that of a common "HEPDDL". Some discussions – particularly between ZEBRA and CHEETAH – took place, but the great tsunami of object oriented programming and design was soon to engulf us.

ADAMO

The following description of the ADAMO system is copied verbatim from the abstract of the corresponding paper presented in Erice by Paolo Palazzi:

"The ADAMO (Aleph DAta MOdel) system was started in the early eighties in the ALEPH experiment as an attempt to apply state of the art concepts of data modelling and data base management systems to algorithmic FORTRAN programs, especially particle physics data reduction and analysis chains for large experiments.

The traditional FORTRAN + memory manager style of programming had several drawbacks that limited programmer's productivity and made projects difficult to manage: obscure reference to data objects by offsets in a large vector, arbitrary use of pointers and no automatic correspondence between data structures and their documentation.

ADAMO adopted the principles of database systems, separating the internal representation of the data from the external view, by reference to a unique formal description of the data: the Entity-Relationship model…"

CHEP '91

At CHEP '91 two important papers were presented summarising the status of databases in HEP. One of these papers – Database Management and Distributed Data in HEP: Present and Future [11], by Luciano Barone, described the current state of deployment of database applications and raised the issue of "event databases" – somewhat akin to today's event tag databases but with a very reduced amount of information per event, as a key challenge for future work. The other – Database Computing in HEP [12], by Drew Baden and Bob Grossman – introduced the idea of *"an extensible, object-oriented database designed to analyse data in HEP at the Superconducting SuperCollider Laboratory (SSCL)".* This was clearly not "business as usual" and was subject of much – often heated – debate during the rest of that decade. To skip ahead, the end result – seen from the highest level – was that both viewpoints could be considered correct, but for different domains. For the applications identified at the time of the ECFA study group, the "classical approach" is still largely valid. However, for event data, we have – according to the prediction of Jim Gray *"ended up building our own database management system"*. Will these two domains ever converge, such that a single solution can be used across both? Is this even desirable, given the markedly different requirements – e.g. in terms of concurrency control and other database-like features?

Barone's paper summarised the key characteristics of databases in HEP, as well as describing the experience of the 4 LEP experiments. The similarities between the global approaches of DELPHI, L3 and OPAL were stressed, as well as the close resemblance in many ways of the L3 and OPAL solutions. ALEPH was different in that the initial (see also the discussion on this point in [8]) size of the database was significantly smaller – some 5MB as compared to 60MB for OPAL and 400MB for L3. He also high-lighted ALEPH's use of ADAMO and its DDL for building their system.

Finally, he summarised the work on event directories / tags, as well as event servers. This activity was relatively young at the time, but set to become an important component of future analyses. Event directories were

typically very concise – a given file of run / event numbers – together with their offsets in a file – corresponded to a specific selection. Today's tags are significantly larger and correspond to the input to the selection, rather than the result set.

His definition of databases is interesting in that it had already expanded somewhat from that of the ECFA report. This is primarily in his final (4th) criterion, namely:

> *"A HEP database is accessible and used on different computers and different sites. This is inherent to the nature of present collaborations, geographically very distributed, and with relevant computing resources at home institutes."*

HEPDB

Following on from the discussions in Erice and at CHEP, a small group was setup to study the possibility of a common solution to the experiments' needs in terms of calibration databases – much as proposed by the 'green book'. As had already been revealed, there was a high degree of commonality not only in the requirements but also in specifics of the various implementations – some 20 packages were reviewed at that time. It was fairly quickly – although not unanimously – agreed to build a package based on either OPCAL or DBL3, re-using as much code as possible. In the end, the DBL3 base was preferred, due to its additional functionality, such as client-side caching, and both OPCAL and DBL3 compatibility interfaces were produced. Sadly, neither of these experiments ever migrated to the new code base. However, possibly 20 experiments worldwide went on to the use the system – with continued use by NA48 for its 2007 data-taking. The central server is no longer maintained by IT, with an AFS-based copy of the previous RZ database available for both R/O and update access – the latter under control of NA48 experts.

The main 'added-value' of the central service was to:

- Run a centrally monitored (console operators) service, with operations procedures;
- Provide regular backups and data integrity checks of the DB files;
- Perform recovery if required.

Due to unfortunate bugs in the area of record re-allocation, the latter primarily plagued FATMEN – it being a mantra of DBL3 and hence HEPDB [13] "never delete". FATMEN – on the other hand – by default updated the catalogue on each file access with the last use date and use count. Whether this was ever more than academic interest is far from clear, but it certainly helped to debug the record allocation routines!

HEPDB was supported on VM/CMS, Unix and VMS systems, the latter being plagued by a host of TCP/IP implementations, some of which were not available at CERN and hence could not be fully tested.

In terms of a common development, it represents an interesting example of a package almost entirely developed within an experiment that is subsequently taken over centrally. In this respect, as well as the benefit that it gave to smaller experiments, unable to devote the manpower to (unnecessarily) develop their own solution, it can be considered a success.

As suggested above, the update mechanism for all of these packages was via the exchange of FZ files between client and server. On VM/CMS systems, these files were sent to the virtual card reader of the corresponding service machine, prompting the server to leap into action. On VMS and Unix systems they were written into a special directory which was polled at regular (configurable) intervals. The updates could be replayed if required and similar queues – i.e. directories – were established to exchange updates with remote servers, typically configured in a similar arrangement to that later proposed by MONARC and adopted by WLCG in its Tier0/Tier1/Tier2 hierarchy. DBL3 and hence HEPDB had a concept of a 'master' server – which assigned a unique key and timestamp – and hence updates made at remote sites were first transferred – using the above mentioned routing – to the master site before redistribution. In the case of FATMEN, all servers were equal and updates were processed directly and then dispatched to remote sites. This update mechanism also allowed for recovery – a not uncommon operation in the early days was the excision of a complete directory or directory tree that was then recreated by replaying the corresponding update or 'journal' files. To reduce overhead, the journal files could be batched as required. However, although essentially any manipulation was possible through the API and CLI, global changes were performed much more efficiently by writing a special program that worked directly on the catalogue / database. Such a change would typically come from the change of name of a host or to perform bulk deletions or other operations – a requirement that still exists today. The results could be dramatic – one listing operation that took many hours when using the standard (necessarily general) API took only seconds using a program optimised for that sole purpose.

CHEP '92

A panel [14] 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 SSCL/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.

We follow below 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

At the time of this panel and as described above, two common projects that attempted to address general purpose detector calibrations ("conditions") and book-keeping / file catalogue needs were the two CERN Program Library packages *HEPDB* and *FATMEN*. At a high-level, these 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 catalogues 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" and Object Orientation. 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 catalogue / event meta-data), LHC++ (a CERNLIB functional replacement in C++) and of course ROOT[15].

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, and 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 [16] 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 that of 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 and 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).

WHY EVENT DATA?

The footnote to the first milestone given to the RD45 collaboration deserves some explanation. At the time, it was not felt realistic to use a single solution for the full problem space – from simple objects, such as histograms, to the event data of LHC-era experiments. The initial ideas – as borne out by paper-only records from that time – were to use a common interface, with a backend tailored to the particular domain. There was strong interest in the ODMG 93 standard at that time and this was rapidly proposed as such an interface. It was upon discovering more than one database with an architecture that scaled on paper – borne out by initial functionality and scaling tests – that the focus on a single solution appeared.

CERN joined the vendor-dominated ODMG standards body with "reviewer" status. Meetings were held quarterly, with CERN representation at least twice per year. One such meeting was held in *Providenciales* – an island in the Caribbean, named after a ship that had wrecked off its coast. The group of islands is so remote that a former flag of the currently British colony lying between the Bahamas and Cuba – which was intended to depict a pile of salt (the islands then main source of income) – was retouched to represent an igloo. Even in as remote a location as this – far from any hadron collider – HEPDB support questions were to be found on the sparsely populated beach.

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 important architectural choices.

The Espresso proof-of-concept prototype was delivered, implementing an ODMG compliant C++ binding. Various components of the LHC++ suite were 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.

VLDB '97

A paper [17] presented at this conference on "Critical Database Technologies for High Energy Physics" by David Malon and Ed May addressed the following issues:

"A number of large-scale high energy physics experiments loom on the horizon, several of which will generate many petabytes of scientific data annually. A variety of exploratory projects are underway within the physics computing community to investigate approaches to managing this data.

There are conflicting views of this massive data problem:

- *there is far too much data to manage effectively within a genuine database;*
- *there is far too much data to manage effectively without a genuine database;*

and many people hold both views."

The paper covered a variety of projects working in this area, including RD45, the Computing for Analysis project (CAP) at FNAL, the PASS project and a recent Department of Energy "Grand Challenge" project that had recently been launched.

The paper included a wish-list of DBMS systems, which included:

- *Address at least tens-eventually, hundreds-of petabytes of data.*
- *Support collections of 10^9 or more elements efficiently.*
- *Support hundreds of simultaneous queries, some requiring seconds, some requiring months to complete.*
- *Support addition of 10 terabytes of data per day without making the system unavailable to queriers.*
- *Return partial results of queries in progress, and provide interactive query refinement.*

as well as a number of requirements related to mass storage systems, either as back-ends or else integrated into the DBMS.

This confirmed that there was some commonality in the approaches of the different projects but that there were still many issues that remained still unresolved – the stated goal of the paper being

"…to begin a dialog between the computational physics and very large database communities on such problems, and to stimulate research in directions that will be of benefit to both groups."

In passing, it is interesting to note the relatively modest ATLAS event sizes foreseen at that time, with 100KB/event at the event summary data (ESD) level, compared with 500KB/event at the time of writing.

LC(R)B WORKSHOPS

During this period a series of workshops focusing on LHC computing was organized by the LHC Computing (Review) Board. These took place in Padua in 1996, in Barcelona in 1998 and in Marseille in 1999. For a short period, it looked as though the combination of Objectivity/DB together with HPSS might even become a semi-standard across HEP laboratories, with experiments from many sites investigating these as potential solutions. However, with time, opinions began to diverge, fueled in part by the slowness in delivery of important features – such as a non-blocking interface to mass storage (designed by SLAC), the full Linux port, support for the required compilers and so forth. The mass storage interface – which would probably never have been delivered had it not been for SLAC's design and indeed proximity to Objectivity's headquarters in Mountain View, allowed the system to be deployed in production. This interface was both powerful and flexible and allowed CERN to later move the backend to CASTOR in a largely transparent way.

CHEP 2000

"All is not well in ODBMS-land". This quote from Paris Sphicas in his summary talk [18] at CHEP 2000 effectively acted as a death knell for object databases in HEP.

One of the key presentations at this conference was BaBar's experience in scaling to full production level. Many adjustments had to be made to achieve the required degree of performance and scalability, leading to the conference quote *"either you have been there or you have not"* – and at the time of writing, there are a number of important aspects of the LHC experiments' computing models – not just limited to database services – that have not yet been demonstrated at full production load, let alone for all experiments at all relevant sites concurrently.

Also during this CHEP, not only were the various aspects of the RD45 risk analysis presented, but also a number of experiments presented their experience with hybrid or non-ODBMS solutions. Questions were clearly raised as to whether an ODBMS solution was the only path ahead or even a useful one. Although the formal decision to change the baseline persistency solution was still some distance away, the community in general had lost confidence in this approach and by this stage it was simply a question of time. As more and more effort was devoted to investigate alternative solutions, a swing back in favour of a commercial ODBMS became increasingly unlikely. The only remaining issues being:

- How to rapidly identify and if necessary provide such an alternative;
- What to do with existing data.

LCG RTAG1

The newly formed LHC Computing Grid project setup its first Requirements and Technical Assessment Group (RTAG1) in February 2002 with the following mandate:

"Write the product specification for the Persistency Framework for Physics Applications at LHC:

- *Construct a component breakdown for the management of all types of LHC data;*
- *Identify the responsibilities of Experiment Frameworks, existing products (such as ROOT), and as yet to be developed products.*
- *Develop requirements/use cases to specify (at least) the metadata/navigation components.*
- *Estimate resources (manpower) needed to prototype missing components.*

RTAG may decide to address all types of data, or may decide to postpone some topics for other RTAGs, once the components have been identified. The RTAG should develop a detailed description at least for the event data management. Issues of schema evolution, dictionary construction/storage, object and data models should be addressed."

Based on the final report of this RTAG and the recommendations of the LCG, the POOL project was established, which is now the baseline persistency solution for ATLAS, CMS and LHCb – ALICE using native ROOT for this purpose.

THE TRIPLE MIGRATION

Following the decision to move away from Objectivity/DB at CERN, the data of the experiments that had used this system had to be migrated to a supported alternative. The needs of the pre-LHC – i.e. running – experiments were somewhat more urgent and could not wait for a production release of the POOL software. Hence, the following strategies were proposed:

- The data of the LHC experiments would not be migrated but maintained until rendered obsolete by a sufficient quantity of newly simulated data in the agreed LHC persistency format;
- The data of the pre-LHC experiments would be migrated to a combination of Oracle (for the event headers / tags / meta-data) and DATE (ALICE raw data format).

More than 300TB of data was migrated in all – a triple migration [19] as it involved:

1. Migration from one persistency format to another;
2. Migration from one storage medium to another;
3. Migration of the associated production and analysis codes.

It also required a degree of R&D on the target solution – not only Oracle as a database system but also Linux/Intel as a hosting platform. This work is described in more detail below.

This triple migration required a significant amount of human effort and computer resources. However, as we shall see later regarding maintaining long-term scientific archives, such migrations need to be foreseen if data is to preserved even in the medium term – it is far from guaranteed that the media chosen at the beginning of LHC will be readable by the end, and a migration of tape format is a convenient time to perform other pending migrations.

"…we describe the migration of event data collected by the COMPASS and HARP experiments at CERN. Together these experiments have over 300TB of physics data stored in Objectivity/DB that had to be transferred to a new data management system by the end of Q1 2003 and Q2 2003 respectively. To achieve this, data needed to be processed with a rate close to 100MB/s, employing 14 tape drives and a cluster of 30 Linux servers. The new persistency solution to accommodate the data is built upon relational databases for metadata storage and standard "flat" files for the event data. The databases contain collections of 10^9 events and allow generic queries or direct navigational access to the data, preserving the original C++ user API. The central data repository at CERN is implemented using several Oracle9i servers on Linux and the CERN Mass Storage System CASTOR."

SECURITY ISSUES

A well known security incident in recent years drew attention to the amount of responsibility a site such as CERN can have for database servers deployed at external sites. The clear answer is *none*. Although there are a number of well documented practices that can significantly reduce exposure to typical security exploits – and the consistent use of *bind variables* is one of them – the responsibility for site-local services must run with the site concerned. Nevertheless, in the aftermath of this event it was agreed that response to severe security threats must receive top priority – even if it meant stopping the accelerator. This was the first time that such agreement was reached but can be expected to have similar consequences to other Grid-related services and beyond.

LESSONS LEARNT IN MANAGING A PETABYTE

BaBar's experience in managing a PB database using Objectivity/DB and HPSS, the enhancements that they

found it necessary to introduce and their subsequent migration to a 2nd generation solution provide an extremely valuable case study in this story [20]. Of particular note:

> *"The commercial ODBMS provided a powerful database engine including catalogue, schema management, data consistency and recovery, but it was not deployable into a system of BaBar's scale without extra effort. Half a million lines of complex C++ code were required to customize it and to implement needed features that did not come with the product."*

The paper describes in detail the enhancements that were required to run a production service and – of particular relevance to the Grid community – how to deal with planned and unplanned outages. Less than three full time DBAs were required to manage the system – although this in itself raises scalability concerns for the LHC, where each experiment is expected to generate roughly this amount of data per year. Hiring an additional 3 DBAs per experiment per year would clearly not be affordable.

Again, the lessons learned from the 2nd generation refactoring can clearly be expected to have some importance for the LHC programme, particularly as BaBar 'led' by 'following' the LHC decision.

The paper concludes (penultimate sentence) with:

"Planning for change makes inevitable migrations practical."

A lesson we would clearly be advised to follow for the LHC.

VLDB 2000 PREDICTIONS

The 26th Very Large Database (VLDB) conference, held in Cairo in September 2000, included a panel on predictions for the year 2020. One of these was that yotabyte (10^{24}B) databases would exist by that time. Now a yotabyte is a lotta bytes. By 2020, the LHC might have generated around 1EB – 10^{18}B of data. 1YB is 10^6 times larger – and would require not only significant advances in storage but also in processing capacity to handle effectively. In particular, we cite Jim Gray's work on the need for balanced systems. Finally, 2020 is perhaps 3 – maybe 4 – product cycles away. Today's largest databases are perhaps scraping a PB. What will be the driving forces behind the need for such massive data volumes?

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 considered a success. There were certainly limitations – which is something to be expected. The fact that the current persistency solutions for all LHC experiments (which differ 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 transient and persistent data models – the purist ODBMS view was that they were one and the same. 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

Following the recommendations above at the end of the Espresso study, and 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.

At the time of writing, the CERN physics database services consists of:

"Over 100 database server nodes are deployed today in some 15 [TB sized] clusters serving almost 2 million database sessions per week. [21]"

OPENLAB & ORACLE ENHANCEMENTS

Although the explosion in Oracle database applications had yet to happen, a concerted effort was made to ensure that any necessary enhancements were delivered in production well ahead of LHC data taking. The main areas targeted were:

- Support for native IEEE *float* and *double* data types;
- Removal of any scalability limitations, such as 16bit fields etc.;
- Support for Linux and commodity hardware;
- Improvements in the area of transportable tablespaces – foreseen not only for bulk data exchange between sites, but also for building a potential interface to mass storage systems;
- Reduction in administrative overheads.

Work on these issues was initially started as a continuation of the longstanding relationship between the company at CERN and then continued more formally as part of CERN's openlab – designed to foster exactly such industrial partnership in Grid-related areas. As part of the openlab work, a variety of high-availability and related techniques were evaluated and prototyped, with the clear goal of production deployment (where appropriate) in the short to medium term. Areas studied included the use of commodity Linux systems to host database clusters, Oracle's DataGuard for high availability and to help perform transparent upgrades, as well as Oracle Streams for data distribution. All of these solutions are now routinely used as part of the production services deployed at CERN and elsewhere.

Indeed, at the time of the Oracle 10g launch in San Francisco, CERN was publically acknowledged for its contribution in driving the database area forward.

CLUSTERS

Clusters have played an important role in database deployment at CERN throughout this quarter century. From the first VAXCluster in the mid-eighties, which hosted the LEP DB and other services, through the Oracle Parallel Server some ten years later, to today's Real Application Clusters (RAC). These systems are linked by more than name: the clusterware of VMS was later made available on Digital Unix systems, and is now used on Linux systems in RAC environments. Architecturally, a RAC and VAXCluster have a number of similar features – not only the distributed lock manager but also a dedicated interconnect for cluster communication. Indeed, many of the centres of excellence for VAXClusters – such as Valbonne in southern France and Reading in the UK – are now centres of excellence for RAC systems. The LEP DB service also implemented disk-resident backup – again close to two decades before its time.

The use of clusters has a number of advantages – not only a high(-er) availability solution, they also allow more flexible CPU and storage allocation than in a single server solution, such as a conventional diskserver. However, not all applications scale well in a cluster environment: conventional wisdom being that those that perform well on an SMP will adapt well to a cluster.

ENTER THE GRID

The LHC Computing Grid (LCG) has a simple hierarchical model where each Tier has specific responsibilities. There is a single Tier0 – CERN, the host laboratory, with O(10) Tier1 sites and O(100) Tier2s. To first approximation, the sum of resources at each level is roughly constant. The roles of the different Tiers are as follows:

- Tier0: safe keeping of RAW data (first copy); first pass reconstruction, **distribution of RAW data and reconstruction output to Tier1**; reprocessing of data during LHC down-times;
- Tier1s: safe keeping of a proportional share of RAW and reconstructed data; large scale **reprocessing** and safe keeping of corresponding output; **distribution of data products to Tier2s** and **safe keeping** of a share of simulated data produced at these Tier2s;
- Tier2s: Handling **analysis** requirements and proportional share of **simulated event** production and reconstruction.

Whilst databases are not explicitly mentioned in this high level view, one does not have to dig very deep to find that they are behind virtually all services in the Grid. Many, as we shall see, had their counter-part in the LEP era. Some – in particular in the case of workload management and the handling of Grid certificates – are new and – at least when all relevant components handle roles and groups correctly – can be considered defining elements of the Grid.

EDG-RLS DEPLOYMENT

One of the first Grid services to be deployed that required an Oracle database (in fact also the Oracle Application Server) was the EDG Replica Location Service. This was a critical service, which, if unavailable, meant that:

- Running jobs could not access existing data;
- Scheduling of jobs at sites where the needed data was located was not possible.

The Grid – if not down – was at least seriously impaired. As a result this was taken into account when designing the service deployment strategy & procedures. In addition to trying to define a service that was highly available and for which all possible recovery scenarios were tested and documented, an attempt was made to package the software – together with the underlying Oracle components – in a manner that made them trivial to

install, both on CERN instances and at Tier1 sites outside. This proved to be an extremely difficult exercise – in part as many of the sites involved had at that time little or no experience with the technologies involved. Furthermore, despite repeated attempts at producing some sort of "appliance" that simply ran unattended, such a self-managing, self-healing database system still seems to be as far off today as when first suggested more than ten years ago. The only possible alternative to in-house expertise is for 'hosted applications', as has been done successfully at CERN for the Oracle*HR service. Could this ever be extended to Grid middleware services?

JIM GRAY'S VISIT

Having followed the progress in HEP on using databases for physics applications for many years, he visited CERN in 2001 and attempted to convince us to:

"Put everything online, in a database".

One concrete proposal that he made at the time was for a *geoplex* – namely where data is stored (online) in two or more places (as is largely done in the LHC Computing Grid) and to *"scrub it continuously for errors"* (as is not).

He continued:

"On failure, use other copy until repaired – refresh lost copy from safe one(s)."

As a further potential advantage, the copies may be organized differently, e.g. optimized for different access patterns. As we are now witnessing 'silent corruption' at a level that is bound to impact the large volumes of data already collected – let alone those that will be produced when the LHC starts up – this wisdom now seems particularly pertinent.

He also argued:

"In reality, its build versus buy. If you use a file system you will eventually build a database system:

- *metadata,*
- *query,*
- *parallel ops,*
- *security,….*
- *reorganize,*
- *recovery,*
- *distributed,*
- *replication,"*

Finally, his top ten(?) reasons for using a database were:

1. Someone else writes the million lines of code
2. Captures data and **Metadata**,
3. Standard interfaces give **tools** and **quick learning**
4. Allows **Schema Evolution** without breaking old apps
5. **Index** and **Pivot** on multiple attributes space-time-attribute-version….
6. **Parallel terabyte searches** in seconds or minutes
7. Moves **processing & search close to the disk arm** (moves fewer bytes (qestons return datons).
8. **Chunking** is easier (can aggregate chunks at server).
9. Automatic **geo-replication**
10. **Online update and reorganization**.
11. **Security**
12. If you pick the right vendor, ten years from now, there will be software that can read the data.

Jim is well known for his work on databases in astrophysics, where he demonstrated that quite complex queries can indeed be expressed in SQL. Some examples include:

Q1: Find all galaxies without unsaturated pixels within 1' of a given point of ra=75.327, dec=21.023

Q2: Find all galaxies with blue surface brightness between and 23 and 25 mag per square arcseconds, and -10<super galactic latitude (sgb) <10, and declination less than zero.

Q3: Find all galaxies brighter than magnitude 22, where the local extinction is >0.75.

Q4: Find galaxies with an isophotal surface brightness (SB) larger than 24 in the red band, with an ellipticity>0.5, and with the major axis of the ellipse having a declination of between 30" and 60"arc seconds.

Q5: Find all galaxies with a deVaucouleours profile ($r^{1/4}$ falloff of intensity on disk) and the photometric colors consistent with an elliptical galaxy.

DATABASE APPLICATIONS IN THE LHC ERA

Whilst the database applications for the LHC experiments can be broadly categorized as was done for LEP in the green book, there are a number of distinguishing characteristics that require additional attention:

- Those applications that are critical to the experiments production processing and data distribution;
- Those that require some sort of distributed database solution.

(Some may fall in both categories).

In this section we focus on the latter, as the techniques for handling the former are largely the same as for production Grid services and are hence discussed below.

To date, the only application in this category is that of detector calibrations / conditions (for LHCb, a replicated

file catalogue [22] is also made available using the same technologies that we shall describe, once again echoing the situation in the LEP era).

ALICE have chosen to base their conditions data on ROOT files, distributed in the same way as for event data, together with the Alien file catalogue.

CMS have implemented their own conditions application on top of Oracle, which uses caching techniques to make conditions data available to Tier1 sites and thence out to Tier2s. Based on experience at FNAL, the overall system consists of an Oracle database together with a FroNTier [23] server at the Tier0 and Squid web caches at the Tier1 and Tier2 sites. Data is also exchanged between online and offline systems, using the same Oracle Streams [24] technology that is used in a wider sense by ATLAS and LHCb.

ATLAS and LHCb have adopted a common solution based on the COOL package. The data maintained in the backend databases is replicated using Oracle Streams to Tier1 sites, with data flows also to / from the online systems. ATLAS has the largest number (10) of Tier1 sites and also has 3 special "muon calibration centres" that are not Tier1s but play a specific role in this exercise, with calibration data flowing back to CERN and then out again.

"To enable LHC data to flow through this distributed infrastructure, Oracle Streams, an asynchronous replication tool, is used to form a database backbone between online and offline and between Tier-0 and Tier-1 sites. New or updated data from the online or offline database systems are detected from database logs and then queued for transmission to all configured destination databases. Only once data has been successfully applied at all destination databases is it removed from message queues at the source."

The distributed solutions for all experiments except ALICE are coordinated by the LCG 3D project [25]:

"describes the LCG 3D service architecture based on database clusters and data replication and caching techniques, which is now implemented at CERN and ten LCG Tier-1 sites. The experience gained with this infrastructure throughout several experiment conditions data challenges and the LCG dress rehearsal is summarised and an overview of the remaining steps to prepare for full LHC production will be given."

Whilst extensive testing of these solutions continues, full scale LHC production experience is needed to iron out any remaining issues.

Since the adoption of Objectivity/DB by the BaBar experiment at SLAC, a whole host of conditions database implementations have been produced. The first such implementation, by Igor Gaponenko [26], was introduced at CERN and eventually migrated to Oracle. A new implementation – COOL [27] – was subsequently made at CERN, this being the baseline choice of ATLAS and LHCb. The COOL system itself is based on CORAL [28] – the The COmmon Relational Abstraction Layer:

"the LCG Conditions Database Project ... COOL, a new software product for the handling of the conditions data of the LHC experiments. The COOL software merges and extends the functionalities of the two previous software packages developed in the context of the LCG common project, which were based on Oracle and MySQL. COOL is designed to minimise the duplication of effort whenever possible by developing a single implementation to support persistency for several relational technologies (Oracle, MySQL and SQLite), based on the LCG Common Relational Abstraction Layer (CORAL) and on the SEAL libraries."

EVENT TAGS REVISITED

At the time of writing, ATLAS is the only LHC experiment potentially interested in storing event tags in an Oracle database. The experiment with the most experience in this respect is COMPASS, who currently store some 6TB in Oracle, following their migration from Objectivity/DB. However, the COMPASS tag database is maintained centrally at CERN, with a small subset of the data copied to Trieste. (BaBar also maintain a bookkeeping database that is replicated to some 10 sites and even some laptops, but it is at a much higher level and only contains a few GB of data.) Until recently, ATLAS foresaw maintaining tag databases at least at all of their 10 Tier1 sites. It is unclear whether the currently used database synchronization mechanism would be able to handle the volumes (6TB of data in a nominal year of LHC running) and rates involved, and other techniques – such as transportable tablespaces – are also being considered. Recently, this model changed and the latest proposal is to store the tags at those Tier1 sites that volunteer to host them. This is still very much work in progress – the data volumes involved still need to be confirmed and the exact distribution mechanisms agreed and tested.

DATABASE DEVELOPERS' WORKSHOPS

Given the very large number of database applications – and indeed database developers – foreseen for the LHC, a workshop focusing on LHC online and offline developers was organized for early 2005. Around 100 developers signed up for this week-long session, consisting of both lectures and hands-on exercises. Although previous and subsequent training events have taken place, this workshop was unique in focusing on the needs of the physics community.

All attendees at the workshop were given a copy of Tom Kyte's excellent book – "Effective Oracle by Design". Shortly after the workshop, Tom himself visited

CERN and gave a series of tutorials, including one on 'The Top 10 Things Done Wrong Over & Over Again'. Such events are essential given such a large and geographically distributed community and are to be encouraged if they do indeed reduce the support load on the DBA teams, as well as producing applications that are both more robust and performant. It certainly goes in the direction of the ECFA recommendation, although it is unlikely that a DB developer community of more than 100 was imagined at that time. Given that the *type* of application is largely as predicted, can the growth in number of applications – and hence developers – be purely explained by the magnitude of today's detectors? It is surely also related to the fact that databases are a well understood and widely taught technology, whereas the number of true experts in the dark arts of ZEBRA were closer in number to those in the early days of relativity.

GRID MIDDLEWARE AND STORAGE SOLUTIONS

A number of the Grid middleware and storage solutions that are deployed in the LCG rely on a database backend. However, there is no unique solution: IBM's HPSS now used DB2 internally. Sites running dCache typically use PostgreSQL, whereas those deploying DPM use MySQL (Oracle is also supported). CASTOR2 sites run Oracle. The gLite FTS is only supported on Oracle, whereas the LFC can use either Oracle or MySQL backends – the former being preferred for larger sites, i.e. Tier1s and the Tier0. The use of databases in these applications is described in [29].

The VOM(R)S applications were recently ported to Oracle, whereas some Grid components – in the particular the Resource Broker – still only support MySQL.

Given the impressive degree of standardization elsewhere, why is there so much diversity at this level? In the case of IBM's storage solutions, the choice of DB2 is mandated by the vendor. For dCache, PostgreSQL is preferred for licensing reasons. For the other data management middleware, MySQL makes more sense for smaller sites, whereas the additional features of Oracle are required for larger scale production services.

Despite this seeming diversity, there appears to be a set of problems that affect many of the implementations and this is largely related to database housekeeping. Unless maintained – preferably by the application – some tables grow indefinitely until queries first become inefficient and later grind to a halt. Whilst not explicitly covered by the ECFA recommendations, there is clearly a list of 'best practices' that it would be useful to establish to guide not only existing sites but also those yet to deploy the above storage and data management solutions.

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, there is at least the possibility of falling into the same trap.

Food for thought for CHEP 2030 or thereabouts?

THE ECFA REPORT REVISITED

It would be hard to argue that there was a concerted effort to systematically address the recommendations of the ECFA report (apart from in the initial years – leading to the first central Oracle services and some specific

enhancements to KAPACK). Nevertheless, there has been significant progress on all of the issues raised. As described above, databases are now an integral part of current experiments on- and off-line environments and an essential component of the overall production and analysis chain. Perhaps two of the ECFA recommendations deserve further attention:

- Further effort in training for database developers could reduce the amount of effort required to solve key implementation mistakes, such as the infamous lack of use of "bind variables";
- The cost of administering the databases for the experiments is significant. Anything that can be done to reduce this effort – over and above the reduction in support load that would come from better design and implementation – would be welcome.

ISSUES ON LONG TERM ARCHIVES

Very long term data archives are far from a solved problem – maintaining scientific or other data in a way that it is still usable hundreds or thousands of years hence is still not understood. However, there is recent experience in maintaining scientific data with the specific goal of a reanalysis in the light of new theories and / or experimental results. Such a reanalysis was performed on data from the JADE collaboration at the PETRA accelerator at DESY was made in the mid-1990s. Apart from the rather obvious issues of maintaining the data (the tapes in question were found abandoned in the corner of an office), there are issues related to programming languages, which may be obsolete after even a few years – as happened in this case – or more likely the program execution environment. However, the biggest problem as seen in the JADE case and rediscovered in the various attempts at a LEP data archive, has been in the area of metadata – maintaining enough information about the detector and the experiments' bookkeeping so that the bits – even if they can be read – can be meaningfully used. This is a big challenge for the database area, in that the necessary care to identify and preserve all of the necessary metadata must be made well in advance. Waiting until the necessary experts have retired or moved on is simply too late. There are many arguments that scientific data – such as from LEP or the LHC – should be maintained for posterity. However, if we are unable to analyse it even a few years hence, there is little chance of achieving such a notable goal. Arguably, however, this is tantamount to destroying our scientific legacy and is an area that should be addressed with priority.

THE STATE OF THE GRID

As experienced by BaBar and indeed many other experiments beforehand, operating reliable distributed services is a challenge. In the case of a number of the middleware services, redundancy is provided by load-balanced servers, deployed in such a way as to avoid single points of failure, such as power, network switches and so forth. Whilst high availability database technology is well understood in theory, it can be both costly and complex to implement. Indeed, unnecessary complexity – such as cross-site services – may do little to enhance actual availability and may even make it worse. A further element in the equation is that Grid users typically care about much higher level applications than the core Grid services. Often an experiment-level service may be built on a combination of a number of experiment-specific services – some of which may have a database component – as well as Grid services likewise. On the positive side, the Grid is basically a batch environment and so resilience to shortish-term glitches is acceptable and even 'transparent'. However, it is not sufficient to list the basic technologies involved – an in-depth study of the key services and their criticality, followed by a specific implementation consisting of hardware, middleware, procedures and application are required to achieve this goal. At the time of writing this work is clearly 'in progress', but it is well understood that the benefits to both service providers and service users is significant and well worth the effort. The immediate goal is to perform an analysis of the services required by CMS and – once deployed at an acceptable level – perform the equivalent analysis with the other LHC VOs. Clearly, the experience of previous experiments, together with high-availability database techniques, will be essential components of this strategy. The target is to have the key services deployed in this manner early enough to be reported on at CHEP 2009 (March 2009 in Prague).

CHEP 2006

A review of earlier technology predictions highlighted:

"Object databases may change the way that we view storage".

It is hard to guess exactly what was behind this remark. If it was that we would be using commercial object databases to manage all LHC data, then the story is told above. If, however, it was intended to mean that we would finally treat disk storage as random-access, and not just "fast tapes", then indeed the prediction can be considered correct. Furthermore, based on the definition from the early ECFA report, and indeed Jim Gray's analysis of our work, it also correct that we are using databases (commercial or open source) to manage book-keeping and other non-event data, whereas we have build a powerful – albeit not fully featured – object-oriented database system in which the full event data – from raw to tags – is maintained. Indeed, in many aspects this is very similar to the work reported on at CHEP '91 - 'Database Computing in HEP'.

FINAL REMARKS

There is no doubt that the era described above was at times turbulent – both the move to distributed computing

and from "Fortran to OO" resulted in heated debates and often diametrically opposed opinions. However, the ECFA report turned out to be remarkably prescient – apart from relatively minor details, such as the use of ZEBRA RZ in most cases for home-grown solutions, rather than KAPACK. A number of official or semi-official joint projects were established addressing the areas raised by the report – it being in many cases the smaller experiments that benefited most from this work. At the same time, the emergence of commodity computing and a convergence of technologies have made a new era of computing possible, namely that of Grid computing.

We have not yet gained sufficient experience in this environment for a fully objective analysis – this must wait another few years, including the onslaught of full LHC data taking and analysis.

The full story of databases in HEP is worthy of a much longer treatise and an event modelled on the "SQL 25 year reunion" held in Palo Alto in the mid-90's is clearly called for.

ACKNOWLEDGEMENTS

Numerous 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. Particular thanks are due to Harry Renshall for his careful reading of this and many other documents during the past twenty five years, together with frequent guidance and mentoring. Finally, the "Grid Guys & Gals" (3G), working on the various middleware components and associated services, for deploying a host of database applications and in anticipation of the many years of LHC data taking to come. *LHC data archive anyone?*

REFERENCES

[1] *Databases and bookkeeping for HEP experiments*, ECFA Working Group 11, ECFA/83/78 (Sept. 83).
[2] *KAPACK – Random Access I/O Using Keywords*, CERN Program Library Entry Z303.
[3] Database systems for HEP Experiments, Richard P. Mount, Computer Physics Communications 45 (1987) 299 – 310.
[4] The L3 database system, Nuclear Instruments and Methods in Physics Research A309 (1991) 318 – 330.
[5] *ZEBRA Data Structure Management Package*, CERN Program Library Entry Q100.
[6] Computing at CERN in the 1990s, July 1989.
[7] The Aleph DAta MOdel – ADAMO. In the proceedings of [10].
[8] ALEPH bookkeeping and SCANBOOK, Jacques Boucrot, in "The ALEPH Experience", p 128.
[9] *FATMEN – File and Tape Management Package*, CERN Program Library Entry Q123.
[10] Data Structures for Particle Physics Experiments: Evolution or Revolution? ISBN 981-02-0641-0.
[11] Database Management and Distributed Data in High Energy Physics: Present and Future, L.M. Barone, in the Proceedings of CHEP '91, ISBN 4-946443-09-6.
[12] Database Computing in High Energy Physics, in the Proceedings of CHEP '91, ISBN 4-946443-09-6.
[13] HEPDB *Package*, CERN Program Library Entry Q180.
[14] B. Linder, R. Mount, J. Shiers, "Databases for High Energy Physics", in the proceedings of CHEP92, ISBN 0007-8328.
[15] ROOT – an Object-Oriented Data Analysis Framework – see http://root.cern.ch/
[16] POOL – Persistency Framework – see http://pool.cern.ch/
[17] David Malon and Ed May, "Critical Database Technologies for High Energy Physics", in the proceedings of VLDB97, available from vldb.org.
[18] Paris Sphicas, conference summary talk, CHEP 2000, Padua, Italy.
[19] *Objectivity Data Migration*, in the proceedings of CHEP 2003, La Jolla, California.
[20] *Lessons learnt from Managing a Petabyte*, Jacek Becla, Daniel Wang, SLAC, in the proceedings of the 2006 CIDR conference.
[21] Database services for Physics at CERN, M. Girone, submitted to CHEP 2007.
[22] LHCb experience with LFC database replication, B. Martelli et al, submitted to CHEP 2007.
[23] L. Lueking et al, "FroNtier: High Performance Database Access Using Standard Web Components in a Scalable Multi-Tier Architecture", proceedings of CHEP04, Interlaken (September 2004).
[24] Oracle streams – see http://www.oracle.com/
[25] Production experience with Distributed Deployment of Databases for the LHC Computing Grid, D. Düllmann et al, submitted to CHEP 2007.
[26] I. Gaponenko et al, "Using Multiple Persistent Technologies in the Conditions Database of BaBar", proceedings of CHEP06, February 2006.
[27] A. Valassi et al, "COOL Development and Deployment: Status and Plans", in the proceedings of the CHEP 2006 conference.
[28] *CORAL, A Software System for Vendor-neutral Access To Relation Databases,* I. Papadopoulos et al, in the proceedings of the CHEP 2006 conference.
[29] Grid-Enabled Standards-based Data Management, L. Abadie et al, submitted to the 24th IEEE Conference on Mass Storage Systems and Technologies.