The HERA-B database services

for detector configuration, calibration, alignment, slow control and data classification.

A. Amorim[1,2]*, Vasco Amaral*[2]*, Umberto Marconi*[3]*, Tomé Pessegueiro*[2]*,*
Stefan Steinbeck[4]*, António Tomé*[2,5]*, Vincenzo Vagnoni*[3] *and Helmut Wolters*[2,6]
The HERA-B Collaboration

[1] DESY Hamburg, CFNUL Lisboa
[2] LIP Coimbra/Lisboa
[3] INFN - Bologna
[4] University of Hamburg
[5] UBI Covilhã
[6] UCP Figueira da Foz

Abstract

The database services for the distributed application environment of the HERA-B experiment are presented. Achieving the required 10^6 trigger reduction implies that all reconstruction, including calibration and alignment procedures, must run online, making extensive usage of the database systems. The system integrates the DAQ client/server protocols with customized active database servers and relies on a high-performance database support toolkit.

Keywords: Information system; Databases; Slow control; online calibration and alignment; Data quality and tag; Persistent object services

1 Introduction

To measure the CP violation angles at HERA-B implies that the chain of online processing, including full reconstruction, has to achieve a 10^6 reduction of the original 40 MHz interaction rate, with good signal efficiency. The system deals with 600k channels, 0.4k computers, 1k DSPs and more than one thousand processes running online. The experiment is facing many of the challenges of the LHC like environment, in particular, the absolute requirement to perform very sophisticated online reconstruction.

The DAQ system is a servant to the trigger system and houses the trigger processing elements distributing data to them as required. Trigger levels 2, 3 and 4[2], the latter including full event reconstruction, are running on two LINUX/PC farms with more than 200 CPU's each. The event data were kept outside of the database management system.

The necessity of full online reconstruction blurs the classical distinction between online and offline software. The 4th level trigger, and consequently the final event selection depends on the full event reconstruction. ARTE[3] is the common framework for Monte Carlo detector simulation and reconstruction (tracking, vertexing, particle identification), and event display.

2 The software environment and technology opportunities

While the HEP applications of RDBMS were restricted to slow control and run bookkeeping systems, object technologies and the emergence of the new standards led to new approaches. While ODBMS databases were used, simultaneously, CORBA based distributed systems were introduced with persistent object services requirements. There is the need to access, through an open client/server interface, the management of persistent objects, considering analysis, tagging and data-processing by large computing farms.

It is worth to remark that some of the concepts and functionality of the more standard solutions are already present in their HERA-B counterparts. The message passing system (RPM/RPS)[1] is a single integrated message passing interface that includes a naming service for locating processes, and a translation service which hides byte alignment and floating point formats from the user. It is extensively used in the database client/server architecture and allows the optimization of the processing between database servers and clients.

The information systems in the HERA-B environment can be characterized using the parameters (Fig. 1) of inter-process data-flow, data-volume and complexity. The most import parameter of the HERA-B database systems is the managed inter-process data-flow rate generated by update requests from the 2 and 3/4th level trigger farms. Accessing clusters of objects with sizes of the order of megabytes lead immediately to gigabytes of requests.

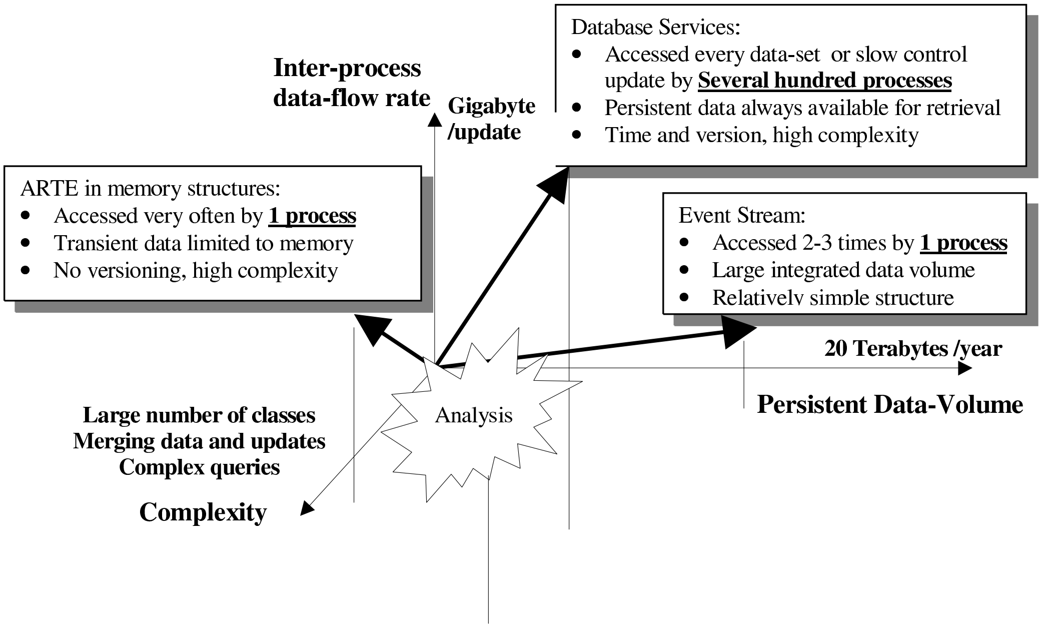
Database Services:
• Accessed every data-set or slow control update by **Several hundred processes**
• Persistent data always available for retrieval
• Time and version, high complexity
Inter-process data-flow rate
Gigabyte /update
ARTE in memory structures:
• Accessed very often by **1 process**
• Transient data limited to memory
• No versioning, high complexity
Event Stream:
• Accessed 2-3 times by **1 process**
• Large integrated data volume
• Relatively simple structure
20 Terabytes /year
Persistent Data-Volume
Analysis
Large number of classes
Merging data and updates
Complex queries
Complexity

Figure 1: The HERA-B information system domains.

3 The architecture and technologies used

HERA-B databases allow selections on time intervals or versions like in the conditions database[5]. To accommodate this requirement, our main software interfaces are implemented on top of the "key" based embedded transactional database, Berkeley DB (also used in Netscape and OSF/DCE among others).

The associations with the events must be efficient and flexible. While the use of time was rejected because it would force accessing the database servers for each event, setting bidirectional associations from events would lead to an unacceptable overhead. The event is instead associated to a calibration/alignment index object, in the database, that refers to all related objects. The association from events to the key object is dynamic to allow simultaneously reproducing the trigger conditions and improved re-processing.

The packages that define the general architecture of the HERA-B database system are depicted in Fig. 2. All client/server methods are built on top of the DAQ communication packages.

The thin SDB layer encapsulates the Berkeley DB API and provides the infrastructure for

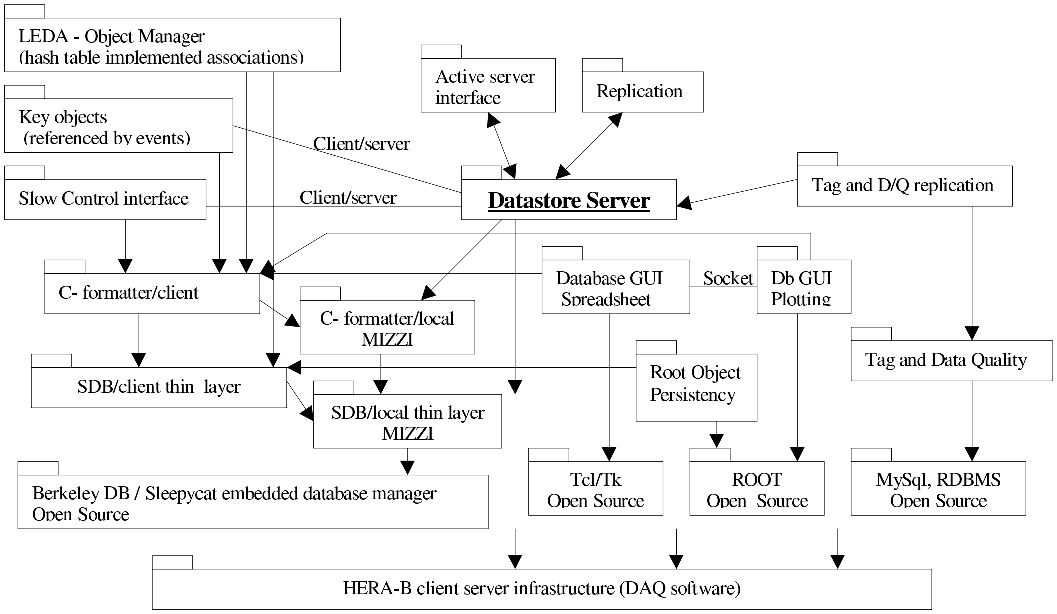

Figure 2: The General Architecture of the HERA-B database system.

client/server requests for indexed unformatted objects. The MIZZI[6] C interface, used since long in the collaboration, provides simple efficient persistence to sets of formatted variable length arrays. The schema is managed together with the data allowing the database servers to optimize the queries internally.

With a total input of 600k channels, and a rather complex detector, HERA-B faces challenges in the area of the slow-control databases that require special solutions. To avoid the presence of too many small objects, the system incorporates the possibility to update individual channels that are part of large collection objects. The slow control interface defines schema, data and update objects. Upon request, the history of values is re-clustered on the database server before being sent to the clients under the control of a specific API.

A simplified object manager layer (LEDA) was developed to provide object persistence and manage associations between objects in related containers. The implemented many-to-many associations are navigated, with the help iterators, using hash tables. Keys are used as object identifiers that have the scope of classes, and associations can only be followed when the classes of objects have explicitly been loaded or saved. It has been extensively used in the configuration databases. The fact that it includes, in parallel to the C++ binding, a simple C interface, together with its availability on LynxOS machines have proven to be very useful. The LEDA package does not allow the database server processes to optimize queries by following object associations. A relational database is used in the case of tag data-quality databases where this limitation has turned out to be problematic.

In the contact with the sub-detector experts, rather than concentrating on the details of the implementation, the UML class diagrams have been used to design improovements in the database schema. An example of the complexity is provided by the DAQ configuration databases with ≈40 classes distributed over 6 packages.

The database server application simultaneously provides persistence to data and performs locally operations that require accessing large numbers of objects. It notifies clients upon storage of subscribed data and provides information on the databases and file systems. The calibration, alignment and setup information is broadcast simultaneously to a large number of machines. It uses the SHARC network to the second level trigger farm, and a tree of cache database servers to the 4$^{\text{th}}$ level trigger processors. This is triggered by the active servers, which propagate update messages notifying the distributed system that constitutes the event stream. The management of the database servers is done through special tools that use configuration databases for the distributed system. Offline access to the databases is achieved by a replication mechanism that isolates the online servers from the load fluctuations induced by offline processing. This mechanism also provides incremental backup of the data.

The database browsing/editing tools resulted form a R&D project that investigated several technologies, and have been implemented by reusing a spreadsheet Tcl/Tk tool, achieving good performance by completely hiding the data from the scripting language.

With an event rate that prevents the efficient use of event tag databases, and the runs extending for periods in which conditions change many times, we have introduced the concept of tagged data sets in the data quality databases. They are managed by a relational database system.

The HERA-B database software is supported on systems running LINUX, Sun/Solaris, SGI/IRIS, and on embedded LynxOs PowerPC's. Mainly the GNU compiler C/C++ has been used in its gcc and egcs versions. The commercial packages used, i.e. Berkeley DB, MIZZI, ROOT and MySql, are all maintained under Open Source policy, a fact that has proved to be crucial in all platform/compiler migration processes.

4 Conclusions and Outlook

The HERA-B database system incorporates special distribution mechanisms to accommodate with a large number of clients in the trigger farms. It also correlates each event with information on the databases, establishing dynamic associations that allow improvement during offline re-processing. Optimizing the information retrieval in the server, before sending it to the client application, has been crucial for the database system performance.

The database system that is involved in online data-taking has been successfully commissioned. The online database cluster includes 5 machines that are able of high network throughput. The status of the database servers is being constantly monitored online as part of the detector slow control alarm system.

A ROOT interface to the HERA-B databases, incorporating persistence to the ROOT analysis objects through the serialization mechanisms, is currently being implemented. A similar solution using Java serialization is under evaluation. Due to the huge number of events produced by HERA-B, the relations between the event directories and event tag databases are still subject of an R&D project.

References

1 V. Rybnikov and the HERA-B Collaboration, "DAQ Online Software and the Run Control System in the HERA-B Experiment", CHEP'98, Chicago, Autumn 1998.

2 A. Gellrich et al., "Full Online Event Reconstruction at HERA-B", CHEP 2000.

3 Hartwig Albrecht, "The Computing Model for HERA-B", CHEP'97, Berlin, Spring 1997.

4 The Sleepycat Software Home Page, `http://www.sleepycat.com`

5 David Brown et. al. "The BaBar Conditions Database", CHEP'98, Chicago, Autumn 1998.

6 MIZZI Computer Software, Thomas Kihm, Mannheim, Germany, 1995