EXPERIENCE WITH POOL IN THE LCG DATA CHALLENGES OF THREE LHC EXPERIMENTS

M. Girone*, M. Branco, R. Chytracek*, D. Düllmann, M. Frank, L. Goossens, G. Govi*, V. Innocente, J.T. Moscicki*, I. Papadopoulos, H. Schmuecker (CERN, 1211 Geneva 23, Switzerland)
K. Karr#, D. Malon#, A. Vaniachine# (Argonne National Laboratory, Argonne, IL 60439, USA)
A. Fanfani, C. Grandi (INFN, Bologna, 40147, Italy)
W. Tanenbaum (Fermi National Accelerator Laboratory, Batavia, IL 60510, USA)
L. Tuura (Northeastern University, Boston, MA 02115, USA)
Z. Xie (Princeton University Princeton, NJ 08544, USA)
T. Barrass (University of Bristol, Bristol, BS8 1TL, UK)
C. Cioffi (University of Oxford, Oxford, OX13NP, UK)

Abstract

The POOL project is a common persistency framework for the LHC experiments to store petabytes of experiment data and metadata in a distributed and grid enabled way. POOL is a hybrid event store consisting of a data streaming layer and a relational layer.
This paper summarises the deployment experience gained with POOL during the data challenges of the LHC experiments that were performed in 2004. In particular we discuss integration issues with grid middleware services such as the LCG Replica Location Service (RLS) and the experience with the POOL proposed way of exchanging metadata (such as File Catalog catalogue entries) in a de-coupled production system.

INTRODUCTION

The POOL project (acronym for **P**OOL **O**f persistent **O**bjects for **L**HC) [1] is the common persistency framework for the LHC experiments to store petabytes of experiment data and metadata in a distributed and grid enabled way. The POOL is a hybrid event store combining C++ object streaming technology via ROOT I/O [2] for the bulk data with a transactionally safe Relational Database store such as MySQL or Oracle for file catalog, collection and metadata.

POOL has been started in the context of the LHC Computing Grid (LCG[3]) Application Area in June 2002, as a common effort between the CERN database and software groups and the LHC experiments, for defining its scope and architecture and for the development of its components. The strong involvement of the experiments has facilitated the implementation of their requirements as well as the integration of POOL into their software frameworks. They were also able to benefit from very short release cycles.

COMPONENT ARCHITECTURE

POOL follows a technology neutral approach. As such, it provides a set of service APIs via abstract component interfaces and isolates experiment framework user code from details of a particular implementation technology. A POOL API consists of three main components: the Storage Service, the File Catalog and Collections, as shown in Figure 1.

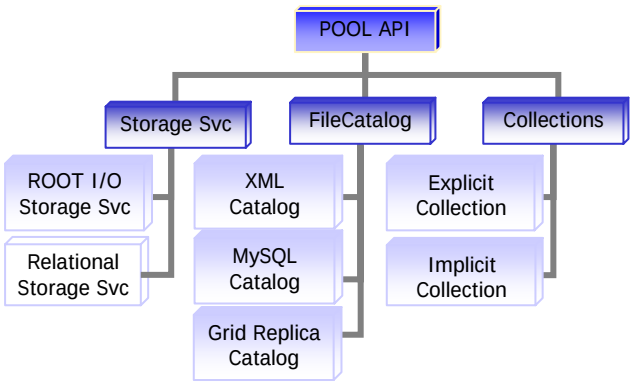

Figure 1: POOL components breakdown

POOL implements a distributed store with full support for navigation between individual data objects across file and technology boundaries. The Storage Service is

* Funded by Particle Physics and Astronomy Research Council, UK.
# Work supported in part by the U.S. Department of Energy, Division of High Energy Physics, under Contract W-31-109-Eng-38.

responsible for streaming C++ transient objects in and from disk. Currently only the ROOT I/O technology is supported but recently a RDBMS storage manager prototype has proven that technology independence has indeed been achieved.
The File Catalog is responsible for maintaining consistent lists of data files or databases mapping the unique and immutable file identifiers, that appear in the representation of the address of an object in the persistent space, to strings that describe the physical locations of the file or database replicas, which are then used by lower level components like the storage service to access file contents. Finally, Collections provide the tools to manage potentially large ensembles of objects stored via POOL.

THE POOL FILE CATALOG

The basic content of a File Catalog, shown in Figure 2, is the many-to-many mapping of logical file names (LFN) to physical file names (PFN). To this standard mapping, POOL has added the system generated FileID, based on so-called Universally or Globally Unique Identifiers (GUID), to provide for stable inter-file reference in an environment were both logical and physical file names may change after data has been written.

In addition to PFN and LFN, the POOL model includes user-defined file-level metadata for production catalog administration, such as querying large file catalogs or extracting partial catalogs (fragments) based on production parameters. In this way, fragments can be shipped to other sites or to decoupled production nodes.

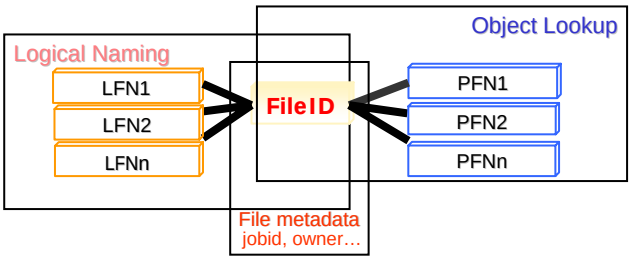

Figure 1: Logical view of the POOL File Catalog

The File Catalog component provides both C++ API and command-line tools, which can be used outside the application process for catalog management operations. Several concrete catalog implementations are provided under single abstract interface. Concrete catalogs are loaded dynamically at run time. End-users can connect to several catalogs at once. Different implementations can be mixed; only one can be updated.

POOL DEPLOYMENT IN THE GRID

The File Catalog component which makes POOL applications coupled to grid services uses the EDG Replica Location Service (RLS) middleware [4], based on Oracle 9i Application Server and Database backend (version 9.2.0.4). File resolution and catalog metadata queries in this case are forwarded to grid middleware requests. Of the EDG-RLS, POOL uses the Local Replica Catalog (LRC) for GUID-PFN mapping for all local files; the Replica Metadata Catalog (RMC) for file-level metadata and GUID-LFN mapping. The Replica Location Index (RLI) component for finding files at remote sites is not used, as it has not been deployed in the LCG-2. Therefore, in the current implementation, only one LRC has been deployed. This resulted in a single centralized service at CERN (provided by the CERN Database group), with scalability and availability concerns.

For grid-disconnected environments, MySQL and XML based implementations of the POOL File Catalog component interface exist, which use a dedicated database server in the local area network (e.g. isolated production catalog servers) or local file system files (e.g. disconnected laptop use cases).

POOL USAGE IN DATA CHALLENGES

POOL has been smoothly integrated in the ATLAS, CMS and LHCb software frameworks [5] and successfully used in their 2004 data challenges [6]. New developments are underway to complete the experiments requirements [7].

CMS DC04

The purpose of the 2004 data challenge (DC04) was to demonstrate the ability of the CMS computing system to cope with a sustained data-taking rate equivalent to 25Hz for a period of one month. DC04 started in March 2004, based on the simulated data produced in the pre-challenge phase PCP04.

POOL has been used during PCP04 to produce about 70TB of simulated data and 24TB of digitized data, corresponding to about 300k jobs. In addition, during DC04, about 4TB of reconstruction data have been produced using POOL.

CMS has used all POOL File Catalog implementations in their production chain. XML catalog fragments were imported in the RLS catalog at CERN. At FNAL, a local MySQL catalog has been used, containing a dump of the RLS. During DC04 the RLS was used both as a file catalog and as a metadata catalog to store file-level

metadata production parameters. A total of about 570k LFNs were stored, each with typically from 5 to 10 PFNs and 9 metadata attributes per file (corresponding to about 1 kB metadata per logical file).

The performance of single-file operations, such as looking up file information by GUID or inserting a physical filename was acceptable. On the other hand, querying information was in general slow. For instance, to find all the files belonging to a given *Owner/Dataset* collection – a LRC-RMC cross-catalog query – was three order magnitudes slower compared to a POOL native MySQL catalog. This problem was caused by missing functionality for bulk operations, such as insert and retrieval for cross catalog operations, together with an overhead introduced by the SOAP-RPC protocol.

Also, as the RLS does not provide transaction support, thus failures during a sequence of inserts/updates require potentially tedious recovery "by hand".

It is worth mentioning that these performance problems occurred only for the RLS backend and they are due to the particular implementation. The proposed POOL model for handling cascades of catalogs, including using production based metadata attributes for shipping catalog fragments, is validated by the good performances of the POOL XML and MySQL based catalogs.

ATLAS DC2

The ATLAS data challenge DC2 has started in July 2004 and is still going on. So far, the total amount of data produced in POOL is of the order of 30TB, corresponding to a total number of 140k files. Anticipated ESD and ASD production using POOL will start in October 2004. At the tag databases level, ATLAS will use MySQL-hosted POOL collections, replicated at many sites.

Production jobs read from and write to local POOL XML file catalogs. The content of XML catalogs is then imported into grid replica catalogs when output files are published. Conversely, XML catalogs are created and shipped with input files when jobs are submitted.

ATLAS uses the RLS to master the POOL file catalog on LCG. Other grids (e.g. Grid3) use the Globus RLS as master catalog. An ATLAS data management tool (Don Quijote [8]) knows how to communicate with multiple catalog flavors. Don Quijote adjusts for the fact that the Globus RLS does not support file GUIDs natively.

Differently from CMS, file-level metadata are maintained in a separate ATLAS bookkeeping service (AMI) that supports queries on datasets and returns LFN lists. Therefore, there is no use of POOL for file- or dataset-level metadata, nor are pattern-matching queries on LFNs performed on POOL file catalogs. Therefore, the above mentioned RLS performance problems do not affect ATLAS, as the RLS metadata functionality was not used.

LHCb DC'04

The LHCb data challenge DC'04 aims to test the robustness of the software and production system. It is composed of three phases: production, event pre-selection and analysis. The production phase ran for 4 months and completed at the end of August 2004. LHCb has produced a large set of data using POOL, which amounts to about 300 TB.

Various POOL components have been used: the Storage Service, with the ROOT I/O backend, the File Catalog, with the XML backend and the persistency service. LHCb does not use the RLS component. POOL is mostly hidden from end-users, being dynamically loaded within the experiment's software framework GAUDI.

POOL related deployment issues concern ROOT and rfio protocol problems, possibly caused by network problems, which are difficult to debug or reproduce.

DEPLOYMENT ISSUES FOR 2005

The deployment of POOL has required setting up Oracle 9i Database and Application Server Services to sustain the current 2004 experiments' data challenges. This service has been deployed at production level for the Virtual Organizations of the 4 LHC experiments, plus parallel set-ups for testing and certification test-beds. The service has been stable throughout the data challenges. Valuable experience has been gained possibly allowing in the near future service consolidation in terms of scalability, manageability, isolation and redundancy using Oracle 10g Database cluster technologies.

On the other side, the experience gained in 2004 data challenges has shown some weakness in the file-level metadata area of the grid aware RLS catalog area. These and other issues are being addressed by the developers of the CERN IT Grid Deployment group [9]. The POOL abstract file catalog interface will allow interfacing to more than one grid catalog. Other possible implementations could be the EGEE gLite or the Globus-RLS.

Other database deployment requirements might come via the newly Relational Abstraction Layer and the ConditionsDB [10] services, which could be set-up on Oracle 10g.

These POOL related services will most likely need to be in line with the distributed environment being proposed by LCG Distributed Database Deployment (3D) project [11].

CONCLUSIONS

Overall, the experience gained in LHC data challenges using POOL in 2004 has been positive: no major POOL-related problems were reported during the challenges of ATLAS, CMS and LHCb, neither from the software nor from the service point of view. The close collaboration between POOL developers and experiments has proven to be invaluable.

Whilst a number of performance problems have been uncovered with the LCG File Catalog implementation, the catalog services themselves have been stable and reliable throughout the data challenges. The performance issues have provided valuable feedback on requirements for future catalog services and are currently being addressed.

The total amount of data stored using POOL was of the order of 400TB. Given that this volume is of the same order of magnitude as that recently migrated out of the previous baseline persistency solution for the LHC experiments, this can be considered as a significant achievement and demonstration of the validity of the POOL model and implementation.

ACKNOWLEDGEMENTS

We would like to thank J. Wu and I. Fisk from the CMS collaboration for providing benchmark values for POOL MySQL at FNAL and the CERN RLS backend.

REFERENCES

[1] T. Wenaus *et al.*, Report of the LHC Computing Grid Project Architecture Blueprint RTAG. http://lcgapp.cern.ch/project/blueprint/BlueprintReport-final.doc .

[2] R.Brun and F.Rademakers, Nucl. Inst.&Meth. in Phys.Res.A389(1997)81-86.

[3] The LHC Computing Grid, http://lcg.web.cern.ch/ .

[4] D. Cameron et al., "Replica Management in the European DataGrid Project", Journal of Grid Computing 2004, in print.

[5] G. Govi, POOL Integration into three Experiment Software Frameworks, this conference.

[6] L. Goossens, ATLAS Production System in ATLAS Data Challenge 2, this conference
A. Fanfani, Distributed Computing Grid Experiences in CMS DC04, this conference
N. De Filippis, Tier-1 and Tier-2 Real-time Analysis experience in CMS DC04, this conference
J. Closier, Results of the LHCb experiment Data Challenge 2004, this conference

[7] D. Düllmann, POOL Development Status and Plans, this conference.

[8] M. Branco, Don Quijote - Data Management for the ATLAS Automatic Production System, this conference.

[9] J.P. Baud, The Evolution of Data Management in LCG-2, this conference.

[10] A. Valassi, LCG Conditions Database Project Overview, this conference.

[11] D. Düllmann, On Distributed Database Deployment for the LHC Experiments, this conference.