POOL, the LCG Persistency Framework.

Ioannis Papadopoulos
CERN, IT Division, Geneva, Switzerland

(for the POOL team)

abstract

The POOL project is currently developing a common object storage framework for the LHC Computing Grid (LCG) application area. The POOL software is developed for the storage of experiment and meta data in the multi-petabyte scale in a distributed and grid computing environment. The project follows a hybrid approach combining C++ object streaming technology such as Root I/O for the bulk data with transactionally safe relational database such as MySQL for collections and meta data. POOL is based on a component model. It provides navigational access to distributed data without exposing details of the particular underlying storage technology. The POOL software is expected to be already used in test production by the time of the conference.

Summary

The POOL software project started in 2002 within the context of the Applications Area of the LHC Computing Grid (LCG) project. Its scope and the main directives for its architecture had been defined in by the LHC experiments in collaboration with the relevant group in the IT Division at CERN.

POOL addresses the needs of the LHC experiments in software infrastructure for storing and accessing data from the detectors, their conditions and their environment, as well as data generated with computing processes of physics and detector simulation, event reconstruction, calibration and high level physics analysis. All these data categories are associated to different scales of recording rates and volume sizes. The categories also differ in their access patterns. Given the lifetime of the LHC experiments, which is rather long with respect to the cycle of a typical technological trend in software, it is expected that the back-end storage solutions will be upgraded or even replaced with time. At the same time all data shall always be accessible in a completely transparent way for the user independently of their recording time and their access path. The latter could be as simple as locating a file in the hard disk of a portable computer running locally an interactive application, or as complex as opening a replicated database in a batch job in a grid environment.

The POOL architecture had to primarily reflect the solution satisfying the above main requirements. Three major domains have been identified, each of them comprising a set of components interacting weakly via abstract interfaces: the Storage Manager, the File Catalog, and the Collections.

The Storage Manager domain is the heart of the POOL system. It is responsible

for providing I/O for the data that come from the detectors, simulation and reconstruction programs. These categories of data are characterized by their large volume size and by the fact that they are written once, very often read back, and very seldom updated. It has been recognized that this access pattern is best served using techniques of C++ object streaming. Therefore, ROOT I/O has been the obvious choice for the first production implementation of the storage back end. This does not necessarily exclude other possible implementations based on RDBMS solutions. In fact, the representation of the address of an object in the persistent space includes a technology identifier, thus allowing complex data structures spanning several files or databases of different technologies. The translation of the transient to persistent representation and vice versa is done using the LCG Dictionary software, allowing for the access of the persistent data from different LCG applications. The components in the Storage Manager domain take care of the necessary technology dispatching so that the user interfaces that need to be exposed are minimal. The latter consist mainly of smart pointers to transient objects, generic cache management services and high level transactions.

The File Catalog domain is responsible for maintaining consistent lists of data files or databases mapping the unique and immutable file identifiers, which appear in the representation of the address of an object in the persistent space, to strings which describe the physical locations of the file or database replicas. POOL provides three different implementations of the File Catalog interfaces: An XML-based implementation, which serves the needs of being able to run simple applications even on a computer disconnected from the network, a MySQL-based implementation, which is best suited in a production farm serving multiple processes and users, and an EDG-RLS-based implementation, which serves an entire Virtual Organization in a data and computing grid environment.

The purpose of the Collections domain is to provide an infrastructure to support the definition, creation, population, use, and management of ensembles of objects stored by means of the persistence services of POOL. These collections of objects are associated to a set of meta data which are used for the rapid selection of the data that need to be analysed, as well as for high-level analysis based on the values of major and summary attributes of the selected data. They are equivalent to the concepts of the event tags and the ntuples that have been heavily used for physics analysis in HEP. Current implementations are based on MySQL tables and ROOT object structures. Moreover, the same interfaces are exposed to the users for implicit collections, which are defined by physical containment of objects in their actual store.

POOL made the first public release in May 2003 aiming at functional completeness in June 2003. By the time of this conference POOL will already be used in preproduction mode for the needs of the data challenges performed by the LHC experiments.