Contribution ID: **19**

Type: **Oral Presentation**

CORAL - A Relational Abstraction Layer for C++ or Python Applications

Wednesday, 13 April 2011 17:00 (30 minutes)

Conclusions

The CORAL software is widely used for accessing from C++ and python applications the data stored by the LHC experiments using a variety of relational database technologies (including Oracle, MySQL and SQLite). It provides generic software libraries and tools that do not specifically target the data models of the LHC experiments and could therefore be used in any other scientific domain.

Overview

The huge amount of experimental data from the LHC and the large processing capacity required for their analysis has imposed a new approach involving distributed analysis across several institutions. The non-homogeneity of policies and technologies in use at the different sites and during the different phases of the experiment lifetime has created one of the most important challenges of the LHC Computing Grid (LCG) project. In this context, a variety of different relational database technologies may need to be accessed by the C++ client applications used by the experiment for data processing and analysis. The Common Relational Abstraction Layer (CORAL) is a software package that was designed to simplify the development of such applications, by screening individual users from the database-specific C++ APIs and SQL flavours.

Impact

CORAL provides generic software libraries and tools that do not specifically target their data models and could therefore be used in any other scientific domain to access relational databases from C++ or python applications.

Description of the work

CORAL is a C++ software package that supports data persistency for several relational database backends. It is one of three packages (CORAL, POOL and COOL) that are jointly developed by the CERN IT Department and the LHC experiments within the context of the LCG Persistency Framework project. The CORAL API consists of a set of abstract C++ interfaces that isolate the user code from the database implementation technology. CORAL supports several backends and deployment models, including local access to SQLite files, direct client access to Oracle and MySQL servers, and read-only access to Oracle through the Frontier/Squid and CoralServer/CoralServerProxy intermediate server/cache layers. Users are not required to possess a detailed knowledge of the SQL flavour specific to each backend, as the SQL commands are executed by the relevant CORAL implementation libraries (which are loaded at run-time by a special plugin infrastructure, thus avoiding direct link-time dependencies of user applications against the low-level backend libraries).

Primary authors: Mr LOTH, Alexander (CERN); Dr TRENTADUE, Raffaello (CERN)

Presenter: Dr TRENTADUE, Raffaello (CERN)

Session Classification: User Environments

Track Classification: Data Management - Technology