Applications Area Report
April 2014 – September 2014

October 2014

Pere Mato

ROOT

The main highlight for the ROOT project for this period has been the release of version 6 at the end of May as scheduled. This version is a major update of the ROOT system. Notably the "good old" CINT C++ interpreter has been replaced by the brand new Cling C++11 interpreter. Cling is built on top of the rock solid Open Source Clang/LLVM compiler infrastructure, which provides the OSX production compiler. Of course, ROOT 6 is backward and forward compatible with respect to the ROOT file format. The great thing of moving to Cling is that it allows future versions of ROOT to support all of the latest C++ language features, focusing on interface improvements and concurrency. To build ROOT 6 from source a C++11 compiler is required (either gcc >= 4.8 or clang >= 3.4). Also a limited number of platforms are supported at the moment due to the porting status of Clang/LLVM (for the time being, Linux i386/x86-64 and OSX 10.9, while Windows 64, AIX/Linux PPC are in the pipeline).

Several follow up versions have been released for the 6 branch and for the 5.34 branch to fix a number of issues detected by end users and the LHC experiments during the integration work done in a very fruitful collaboration with the ROOT team. Most of the major problems have been ironed out and those remaining are related to the increase of memory needs of ROOT 6 with respect to ROOT 5, for which the team is working very hard to reduce it to an acceptable level. The latest release of version 6.02.00 took place at the end of September.

Over the period of reference 575 new issues (bugs, feature requests) have been created and 515 issues have been resolved.

CORAL and COOL

New major releases CORAL 3.0.0 and COOL 3.0.0 have been prepared in the LCG_69root6 configuration, based on the first production release 6.00.00 of ROOT6. They include backward incompatible changes in the APIs of both CORAL and COOL, such as the complete removal of Boost and its replacement by C++11 classes. With respect to ROOT5-based releases, the PyCool package (for interactive Python usage of COOL) is now loading C++ headers at runtime through JIT, rather than at build time through genreflex; one major issue with C++ exception handling in PyCool is still pending, but this will only be fixed when ROOT6 moves from JIT to MCJIT. Tags CORAL 3.0.0a and COOL 3.0.0a have also been prepared for the upcoming LCG_70root6 release that will be built against ROOT 6.02.00; the main difference with respect to LCG_69root6 tags lies in the way PyCool loads C++ headers, which has changed since ROOT 6.00.00.

To allow the experiments to perform detailed comparisons between ROOT5 and ROOT6, CORAL and COOL branches compatible with ROOT5 are still being maintained, without C++11 extensions in their APIs. In particular, release tags CORAL 2.4.3 and COOL 2.9.3 have been prepared for the upcoming LCG_70root5 configuration; their code bases are equivalent to those of CORAL 3.0.0a and COOL 3.0.0a, except for the APIs that are still the same as in CORAL 2.4.2 and COOL 2.9.2.

The migration from the two savannah trackers for COOL and CORAL/POOL to a new single JIRA tracker has been completed. Over 2700 issues have been migrated.

Simulation

The preview release 10.1-beta of Geant4 has been released on June 27th, including most of the developments of the first two quarters to become part of the planned release 10.1 next December. Geant4 10.1-beta introduces explicit use of C++11 types for multi-threading (for thread-local data and atomics) in conjunction with the CLHEP version 2.2 series, with revised use of random number engines directly from CLHEP; this feature is activated only on compilers fully supporting C++11. A new pattern for singletons dynamically allocated is adopted in release 10.1-beta, helping in reducing memory leaked at the end of a job. A new memory allocation strategy for navigation-histories data is also introduced, aiming to reuse memory previously allocated and considerably reduce overall memory churn due to blind copy of vector collections. A bug fix in the fragmentation of hadronic strings used by the FTF model (G4LundStringFragmentation) affects the simulation of hadronic showers: more neutral and less charged pions are now produced, increasing the electromagnetic component, which makes showers more compact and with higher energy response. The detailed list of fixes included can be found in the release notes: http://cern.ch/geant4/support/Beta4.10.1-1.txt.

A new patch release of Geant4 for the 10.0 series, Geant4-10.0.p02, has also been released in June. The patch includes mainly fixes/improvements to multi-threading, porting to new version of the clang compiler and corrections to the CMake system for proper installation of static libraries. Several fixes have been introduced in the USolids module to cope with problems detected when running on realistic/complex detector geometries. The detailed list of fixes included can be found in the release notes: http://cern.ch/geant4/support/Patch4.10.0-2.txt. Validation of Geant4-10.0.p02 has been made by CMS, planning to start production this year with it; confirmed performance speedup of the order of 5-10% when running in sequential mode, compared to the previous release based on Geant-9.6; managed to achieve additional 10% speedup by adopting static libraries dynamically loaded; in aggregate the CPU speedup obtained from Run-I to Run-II is on the order of 50%, thanks also to improvements in CMSSW simulation code and adoption of Russian-Roulette for neutron and gamma particles. Targeting first production release for MT in November; MT runs have been tested with TBB up to 20 threads successfully; few thread-safety issues have been detected and reported; fixes planned for inclusion in a the next Geant4 patch.

Validation of release 9.6.p03 in ATLAS is almost completed; planning to use this version for the next production, including few private patches on top of it ported from more recent releases; these include also the improved allocation strategy for navigation-histories included in 10.1-beta, which, from first measurements, seems to provide up to 75% memory churn reduction and noticeable CPU speedup.

CernVM file system

For the file system the main activity of the past months was the consolidation and deployment of the 2.1 version. The current production version 2.1.19 has been released on May 28th and the overall experience is so far quite positive with stable operations. The migration of the CernVM-FS servers from 2.0 to 2.1 was accomplished according to schedule between beginning of August and mid-September. No major problems were encountered.

A couple of potential problematic situations that can deadlock the system have been identified and fixed in the repository, with workarounds available from the CernVM site.

Some targeted developments also took place during this period. The Parrot module was improved to better cope with multiple CVMFS repositories concurrently; this is particularly interesting for the opportunistic usage of super computing resources. Following the request from the Architect Forum, the priority for an automatic garbage collection system has been raised and a first version is available in the repository since mid-August and is being tested together with the requesting experiments, LHCb and CMS. Finally, work on the configuration bootstrap repository has also continued. These bug fixes and new features and improvements will be available in version 2.1.20, foreseen for before the end of 2014.

CernVM appliance

The period was devoted to fixing the issues found by the users of the first production release 3.1. The current version 3.3 was released end of May; several security hotfixes have been released following the SL6 security updates releases, which can be directly picked-up from CernVM-FS, thanks to the new micro-bootloader technology adopted for the appliance. Work on streamlining the different contextualization options and technologies has been carried on via a GSoC student project and is being integrated in the reference repository.

Data Preservation

The main activity in this context has been the support for the bootloader technology-based SL5 image used for the CMS Open data project which is used to provide access to the available datasets and for their analysis. The official release of the project is foreseen for mid October 2014.

SPI

In the last period, one major LCG release (LCG 69root6) was created. This release served as another validation release for the ROOT 6 developments, but contains other changes as well. The production compiler for the release was gcc48. During the first week of October, a new ROOT 5 vs. ROOT 6 validation will be made available. The 70root5 and 70root6 releases will be based on gc48 and gcc49. In addition to the ROOT 5 vs. ROOT 6 validation, they contain a few updated external software packages and an upgraded version of most of the MC generators.

The technical infrastructure of the SPI project was migrated to using Jenkins for build automation. The build cluster was migrated to the CERN Openstack/puppet infrastructure.

The Savannah-to-JIRA migration activities have almost finished. All projects that requested a data transfer from the old to the new issue tracking services were successfully migrated, in total 91 projects. The projects that requested a static export of their data will get the data provided to them in October. Shortly after the Savannah service will be shut down and be replaced by a redirection facility to the new JIRA based issue tracking portal.