Table of Contents

LHCb Core Software Meeting

Date and Location

01-Sep-2010
10:30 - 12:30
CERN (2-R-030)

Attendees

Gloria, Hubert, Karol, Marco Ca., Marco Cl. (minutes), Markus, Nicolas, Paloma, Rob

Subjects

News

- New version of the Tag Collector in production
    - Marco Ca.: it should be presented during the Software Week, in the advanced tutorials session.
- The Coverity training took place
    - The system is simple to use and can be integrated in the nightly builds (Karol already working on it)
    - If the set up is advanced enough, it could be presented during the Software Week
- New release of ROOT foreseen for September
    - When do we need it? We can adopt it during the winter shutdown (the reprocessing for the Winter conferences will still be done using the old version).

Specific Topics

Discussion on POOL Support (Markus)

We have to decide if we want to keep using POOL or switch to the approach suggested by the ROOT developers, i.e. a single tree with several top level branches.

From the technical point of view the switch will require some special handling of the *optional items*. When storing events in a TTree all the branches must be filled, which means that if an object is not present in the Transient Store, we must anyway store something in the TTree, like a *null* pointer. This difference, with respect to the many-trees approach (POOL), where objects are not stored if not present, implies that the files should be slightly bigger and that it will take more CPU to handle the extra data. It is not clear which of the two approaches is better.

We could also use Atlas' approach (write a single TTree from POOL), but it has the same disadvantages of the switch to ROOT, plus the additional overhead of POOL and its support.

If we decide to switch, we can do it only during the 2012 shutdown, of course preparing and testing the code in the meantime.

Marco Ca. will report to the OPG, but we need solid arguments in favor of the ROOT approach to hope that it is approved.

Abandon Shared Directories in the InstallArea (Marco Cl.)

Some time ago we had a problem in the nightly builds due to fact that files produced in a failed build were replacing the good versions. It is not a problem anymore because now the copies of the shared files are done only if the build is successful, but a long term solution was envisaged in that context.

The proposal was to install all the generated files in platform specific directories. This means duplication of Python script and modules and header files in the InstallArea (one copy per binary platform).

Marco Cl. modified GaudiPolicy to support the new schema (currently enabled with a special CMT tag). This structure introduces also big advantages in the generation of the distribution tarballs, because the *shared* tarball could be reduced to a plain *source* tarball (no need to build all the platform to make it) and each binary tarball is just the content (dereferencing symbolic links) of the binary directory in the InstallArea. In fact, all Gaudi test can be run with an installation produced the just described tarballs.

After this change, we could move toward a build that doesn't require the sources at run-time (installing options files etc.). Of course this is just a possibility for a future (also taking into account the work of Pere on CMake, see next Core Software Meeting.

We agreed to test the new build in one of our nightly slots, aiming for an adoption during the Winter shutdown.

Supported Platforms (Marco Cl.)

In the last few weeks, Marco Cl. addressed the pending problems in the build of Gaudi on Linux+ICC, Windows + VC9, MacOSX (`llvm` needs fixes to the compiler).

Linux+ICC platform (`x84_64-slc5-icc11-dbg`) is currently being built in the lhcb-lcg-head slot thanks to Karol. The terminal server `lhcbts02` is going to be reconfigured in a way similar to `lhcbts03`, then we can properly set up the VC9 nightly builds. We can also start to think about a MacOSX nightly slot.

Recently, it has been suggested in a mailing list to stop deploying the Windows builds, keeping only the nightly builds for validation. We, anyway, agree that such a change will just make the Windows nightly builds ignored more that what they are at the moment.
If we really want to drop a platform, it is probably better to drop MacOSX since we cannot learn anything from its compiler (gcc) that we do not already know from Linux.

The current plan is:

- enable ASAP builds on Windows + VC9 (releases will have to wait for next Gaudi/LCGCMT)
- try to get a full build on ICC, aiming for some performance tests and possibly make it a supported platform
  (we should also test inter-procedural optimizations, IPO, in gcc, together with link time optimizations)
- start a nightly build on MacOSX (to be confirmed with a poll on mailing lists)

Side Note: during the discussion about Windows, it was raised the point of the Visual Studio Debugger.
Eclipse could be used as an alternative on Linux.
Marco Cl. did some more steps to simplify the adoption of Eclipse (EclipseTutorial), which he should try to present it during the Software Week, in the advanced tutorials session.

Round Table

Rob

Marco Ca.

Hubert

Gloria

-- MarcoClemencic - 01-Sep-2010

This topic: LHCb > CoreSoftwareMeetingMinutes20100901
Topic revision: r2 - 2010-09-02 - MarcoClemencic