SOFTWARE MANAGEMENT INFRASTRUCTURE IN THE LCG APPLICATIONS AREA

A.Pfeiffer, S.Ashby, Y.Patois, E.Poinsignon, CERN, Geneva, Switzerland

Abstract

This In the context of the SPI project in the LCG Application Area, a centralized software management infrastructure has been deployed. It comprises of a suite of scripts handling the building and validating of the releases of the various projects as well as providing a customized packaging of the released s/w. Emphasis was put on the flexibility of the packaging and distribution solution as it should cover a broad range of use-cases and needs, ranging from full packages for developers in the projects and experiments to a minimal set of libraries and binaries for specific applications running, e.g., on grid nodes. In addition, regular reviews of the QA analysis of the releases of the projects are performed and fed back to the project leaders to improve the overall quality of the software produced. The present status and future perspectives of this activity will be presented and we will show examples of quality improvement in the projects.

INTRODUCTION

The huge amount of data expected for the experiments at the Large Hadron Collider (LHC) at CERN [1] (several Petabyte/year for each of the four experiments) in combination with the large amount of CPU resources to reconstruct the data (100 kSI2000) provides a unique and unprecedented challenge to reconstruction and analysis software involved. In addition, the lifetime of the experiments (from the start of data taking to the end of the analysis phase) is expected to be of the order of 20 to 25 years, several major changes in the software environment can therefore be expected during this large time-span.

The LCG project

The LHC Computing Grid (LCG) project [2] was set up to coordinate the substantial work and efforts to meet this challenge. The Applications Area [3] is one of the five working areas of the LCG, designated to work on application software common to the LHC experiments. Several projects were set up to work on separate aspects of this common software.

The projects established in the LCG Applications Area cover the following activities (in the order of creation of the projects):

- The POOL project covers the overall persistency framework for the experiments, providing general object persistency in a technology independent way (hybrid ROOT/relational data store, Conditions Database).
- The SPI project provides general software process support and infrastructure services and tools.
- The SEAL project provides core tools and foundation and utility, as well as commonly used mathematical libraries.
- The PI project provides interfaces and tools by which the physicists will directly use the software.
- The SIMU project has several subprojects covering generator services (GENSER), Geant-4, physics validation of MC generators, general framework services and integration of FLUKA.

The projects and their relations to each other and other projects and the experiments are shown in Figure 1.

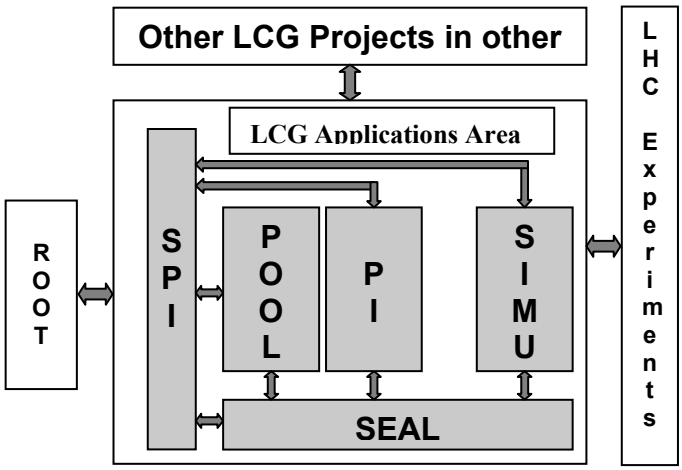

Figure 1: The projects in the LCG Applications Area and their relations.

The SPI project in the Applications Area

The SPI project [4] provides general software process support as well as infrastructure services and tools (such as the Librarian, QA, testing, developer tools, documentation, and training) needed by all projects in the Applications Area as well as for projects in other areas and the LHC experiments.

One of the services provided by the SPI project is to provide “external software" needed by the LCG projects and the experiments. The purpose of this service is to provide all the software packages (libraries and applications) in the versions and platforms needed by the LCG projects; so that they do not need to perform the installations. SPI has defined a standard procedure to provide these installations and a well-documented web site for documenting the external software installed.

The web-based "project portal" for software development based on Savannah is available and has proven to be very popular, and is now in use by the LCG Applications Area projects and by more than 100 other projects in the LHC experiments and projects in other areas of the LCG.

SPI also provides the infrastructure for code documentation and assists the projects with their QA by providing testing frameworks (CppUnit and QMtest) to standardize the way projects perform unit, regression and integration testing. Work is ongoing to ease the use of the QA tools developed in SPI, allowing the projects to enhance the quality of the software by frequent QA inspections and automated generation of QA reports.

SOFTWARE MANAGEMENT INFRASTRUCTURE

LCG AA s/w management infrastructure

Distribution of the projects’ software in the form of tarfiles of the (binary) libraries and source code as well as the dependent (“external”) packages – together with the corresponding documentation – is the main area of work for the software management infrastructure provided by the SPI project.

To ease the load on the developers in the various projects, builds of the projects’ software on the supported platforms – presently there are five supported platforms in addition specific versions, e.g., for debugging and profiling. This environment for automatized builds is also useful in the context of builds (at external sites) on similar, but non-supported platforms.

Other aspects of the s/w management infrastructure cover the centralized packaging of the binaries (and sources) for off-site deployment as well as providing a central quality assurance (QA). The former aims at supporting the various download and installation tools as used in the experiments and the other LCG projects, thus providing a flexible way of installing and upgrading the software. The centralized QA allows detecting cross-project problems in the early phases of (pre-) releases, optimizing the time to deployment for the releases.

LCG AA s/w coordination

In addition to the more technical aspects of the infrastructure management mentioned above, the coordination of the project releases and pre-releases is an important part of the work. This coordination needs to be done both, with the experiments as well as across the projects in the LCG Applications Area.

For the latter, the coordination ensures coherent releases already early on in the pre-release phase of the projects. It also adds flexibility in the pre-release phase as through the coordination the librarians in the experiments can get early access to pre-released relevant for them and perform specific tests to ensure a new feature is working correctly or a specific bug-fix is correct. This coordination can be done on a case-by-case basis such that the overhead of testing can be minimized.

In coordination with the experiments the configuration information is provided for the various specific build tools used. To ease the maintenance on this and to allow for future developments SPI maintains the package (version) dependencies in XML format. This will be extended to also cover the build instructions specific for the projects in XML format; allowing to easily transform this information into any desired format as needed by existing or future build tools.

External s/w packages

This service of the SPI project provides the (mainly) public domain software needed by the LCG AA projects (and the experiments). These packages (“external s/w”) are maintained centrally with first level user support; providing feedback to the authors/maintainers whenever needed. The software is provided for all the supported platforms of the LCG AA (with the exception of debug and profiling versions). The service also provides packaging of the binary versions and has recently been extended to provide installations from source tarballs as well. Present work is concentrating on providing the source installation information in formats suitable for installation with other tools used by the experiments, such as pacman, extracting the information from XML files.

THE LCG AA PROJECTS

The main aims of automating the build and release process of the LCG AA projects are to lower the burden of the developers and to increase the efficiency of the overall release process, providing releases on as short a time scale as possible.

Building and releasing

For building and releasing the LCG AA projects’ software configuration management and control is essential for reproducibility and traceability of the production software of the experiments. While a given release once it is released only configuration control is needed, strict configuration (change) management is important during the development of new releases. During this phase new versions of external s/w packages will be needed or new packages will be introduced and other (dependent) packages will need to be updated correspondingly.

The software of the LCG AA projects is being built on a set of supported platforms which is agreed between the

projects and the experiments. In this context a platform is defined as the combination of a version of an operating system and a compiler version. The supported platforms at present are RedHat 7.3/GCC 3.2, RedHat 7.3/GCC 3.2.3, ScientificLinux3/GCC 3.2.3, MacOSX10.3/GCC 3.3 and Windows/VisualStudio 7.1. In addition to these five "basic" platforms, builds are done for specific versions such as debugging, profiling and/or QA. Not all of the specific versions are required on the entire set of basic platforms, limiting the total number of builds needed.

Given this rather large number of builds needed for each release and the intrinsic dependencies between the packages the releases clearly benefit from some overall coordination. A centrally managed build of the releases of all projects also makes central QA possible; e.g., the experiments can provide their acceptance tests which will be run for each (pre-) release. This way turn-around times can be significantly reduced and early feedback can be given on new features and bug-fixes provided.

DEPLOYMENT AND INSTALLATION

In order to allow other build/installation tools to use the information collected in SPI easily, the various information entities are stored in XML. This way, the information can be easily transformed into any format needed by a given build/installation tool (e.g. pacman).

Distribution formats

The two main requirements for the formats of the distribution/deployment of the LCG AA software emerge from the following use-cases:

1. On platforms which are directly supported or at least fully binary compatible, the pre-compiled binary packages can be downloaded and used directly after their installation.
2. On all other platforms the user wishing to use the s/w needs to download it in source format. In this case, an easy to use tool is required which allows the compilation, test, and installation of the downloaded sources in the users' area without the need for higher (root) privileges.

Deployment use-cases

A set of three use cases has been identified concerning the various viewpoints for s/w installation:

1. The user wants to ***develop*** (part) of an LCG Applications Area project and therefore needs the distribution of the dependent s/w. The project's s/w is then typically checked out of CVS for the development work.
2. The user wants to ***use only a selected part*** of a project, for example the parts of SEAL, POOL and PI needed in batch production within an experiment, or in the context of a Geant-4 advanced example.
3. The user wants to ***use all*** of the Applications Area software, typically to install it for the experiment's collaborators at a remote site.

Configuration control and management

In the use cases above typically the configuration needs strict *control*, allowing only one fixed set of depending packages/versions to be installed for a given release of LCG AA s/w. In case the user is developing part of the LCG AA s/w in question, a less strict *control* is needed; the more complex requirements – the need of the developer to change/add/remove versions or packages – in this case require a strict configuration *management*.

Depending on the role as user, developer or installer, the needs for the download change and need to be taken into account properly.

STATUS AND OUTLOOK

Initial work concentrated on providing an easy to use tool to download and install the pre-compiled binary packages of the LCG AA s/w and their dependent (external) packages. The user needs only to specify which version to download, where to install and which platform to use. Presently, the packages are downloaded from the web, a disk (CD) based installation is in preparation.

For the source based installations a set of python scripts is being developed to install the packages required with their dependencies from source tarballs. The detailed information on how to build the package is extracted into XML format, allowing to convert the information easily into formats needed by other installation managers used in the experiments.

Presently this system is working and under test for the external packages; work is ongoing to extend this to the LCG AA project builds, the prototype version is planned to be ready in late November 2004.

ACKNOWLEDGMENTS

The authors would like to thank the developers within the LCG projects POOL, SEAL and PI and the ATLAS, CMS and LHCb experiments for their collaboration and feedback.

REFERENCES

[1] CERN: http://www.cern.ch/
[2] LCG: http://cern.ch/LCG/
[3] Applications Area: http://lcgapp.cern.ch/project/
[4] SPI: http://spi.cern.ch/