



Contribution ID: 41

Type: **Oral Presentation**

The gLite Data Management Continuous Integration and Testing Process

Tuesday, 12 April 2011 14:30 (30 minutes)

Impact

Currently the DM product team runs every night this procedure on the latest code produced by developers. As such, all changes committed by developers during the day are immediately built and tested every night. Furthermore reports are automatically produced and sent to developers informing them on the impact of their changes. Potential problems are discovered in advance and the code quality increases.

The same process can also be executed during the release of new software versions. In this case the advantage is not on the continuous verification of the code but rather on the easiness to verify if the new release passes all existing tests. This directly contributes to a decrease in the time needed to complete the release process.

Overview

The continuous execution of builds and tests on newly developed software is a key activity for any software development team. To properly implement such procedure it is fundamental to automatize all tasks related to software building, deployment, and testing.

The gLite Data Management product team has put in place a continuous integration and testing procedure to properly support the development and release of the product team software products: DPM/LFC, FTS, and GFAL/lcg_util.

Description of the work

A continuous integration and testing procedure must integrate the execution of source code builds, the deployment of packages, and the execution of different types of tests under a common continuous execution plan. The execution plan defined for the DM product team defines the following tasks: building the source code from SVN (taking code from three repositories: LCGDM, LCGUTIL, FTS); creating platform specific YUM repositories containing all generated packages (including meta packages which represent the services provided by the product team); verification of the correct deployment and configuration of all services; execution of a number of functional, regression, and stress tests; and finally the creation of reports for the product team members. This execution plan is executed n-times on all supported platforms (currently slc4_ia32_gcc346, slc4_x86_64_gcc346, sl5_ia32_gcc412, and sl5_x86_64_gcc412).

The automatic execution of these tasks is managed by a new tool, called SAKET (Swiss Army Knife for ETICS Test), which not only orchestrates the entire process but provides reports to the product team members. These reports are sent by email and archived in a web accessible location for historical reference. SAKET also relies on other existing tools:

- ETICS: to run builds and tests on remote infrastructure nodes;
- YUM: to make available the packages generated by the build;
- YAIM: to deploy the available packages and to configure the services;

- YAIMGEN: to execute the pre-defined tests and to generate reports;

Besides these tools, the continuous integration and testing procedure also relies on a well defined controlled environment to ensure the reproducibility of the build and test executions. This environment is composed by the ETICS infrastructure nodes where the builds and tests actually run and the EMI certification testbed which provides other external services, such as BDII.

Conclusions

A procedure to continuously verify the integrity and stability of the products being developed the gLite Data Management PT has been put in place and is being exploited on a daily basis. This procedure relies and integrates a number of existing tools making more efficient and easier the development and release of new gLite DM releases.

Primary authors: AYLLON, Alejandro (CERN); KEEBLE, Oliver (CERN); ANDRADE, Pedro (CERN)

Presenters: AYLLON, Alejandro (CERN); KEEBLE, Oliver (CERN); ANDRADE, Pedro (CERN)

Session Classification: EMI: Software for Distributed Computing Infrastructures

Track Classification: Producing & Deploying Technology