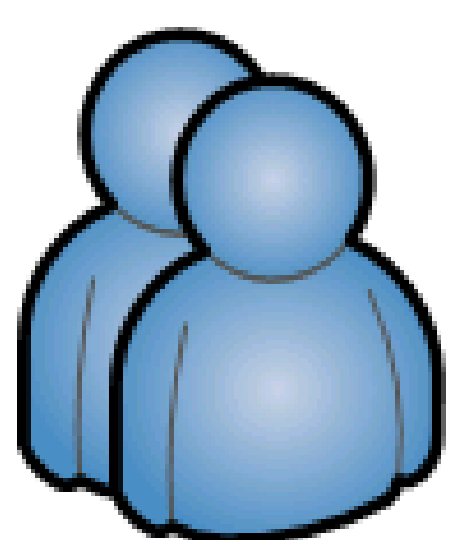# The nightly build and test system for LCG AA and LHCb software
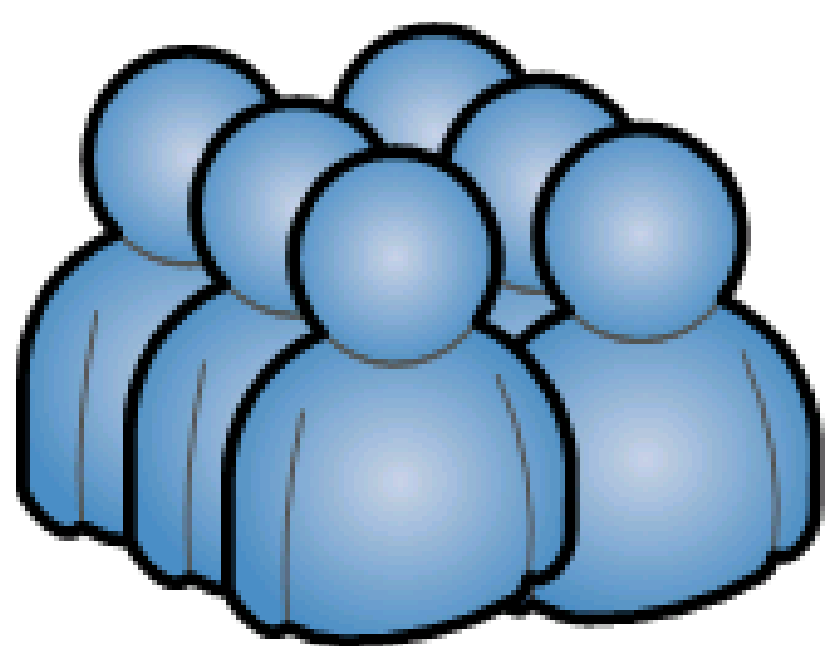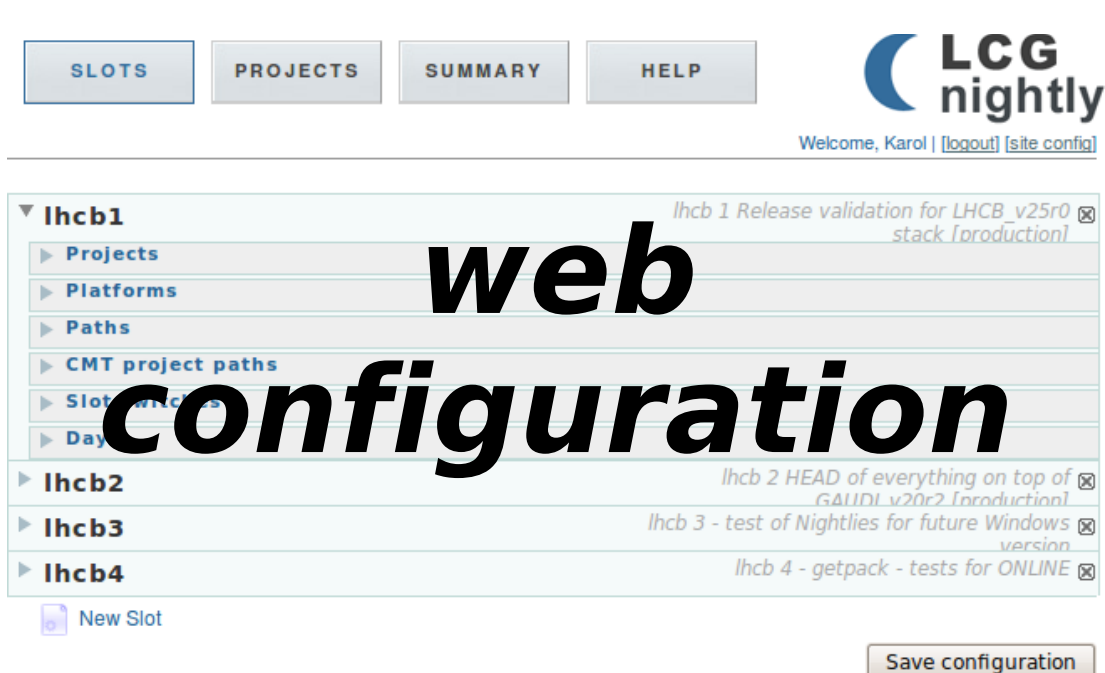
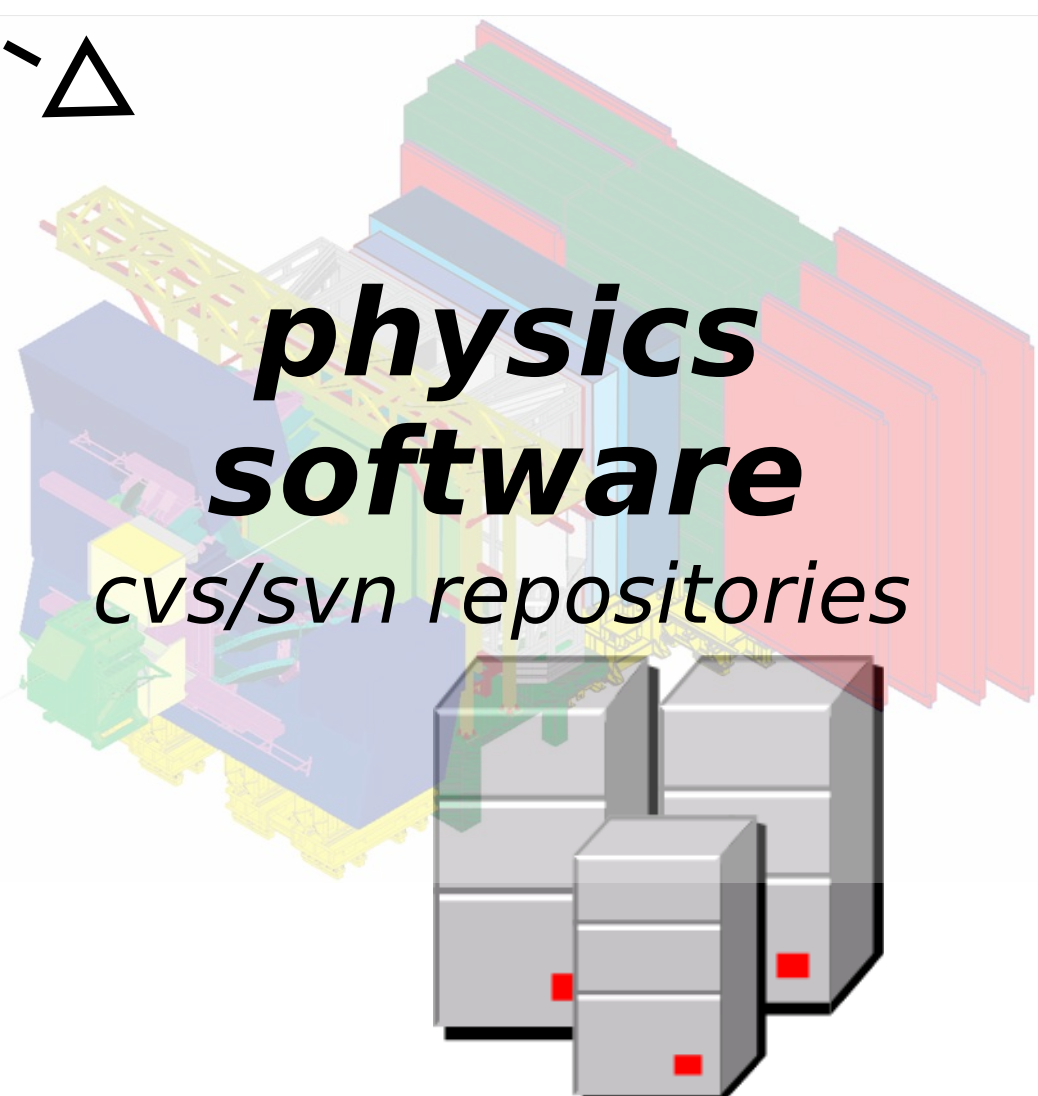Karol KRUZELECKI*, Stefan ROISER**, Hubert DEGAUDENZI*
CERN PH/LBC*, PH/SFT**

The core software stack both from the LCG Application Area and LHCb consists of more than 25 C++/Fortran/Python projects build for about 20 different configurations on Linux, Windows and MacOSX. To these projects, one can also add about 20 external software packages (Boost, Python, Qt, CLHEP, ...) which have also to be build for the same configurations. It order to reduce the time of the development cycle and increase the quality insurance, a framework has been developed for the daily (nightly actually) build and test of the software. Performing the build and the tests on several configurations and platform allows to increase the efficiency of the unit and integration tests. Main features: - flexible and fine grained setup (full, partial build) through a web interface - possibility to build several "slots" with different configurations - precise and highly granular reports on a web server - support for CMT projects (but not only) with their cross-dependencies. - scalable client-server architecture for the control machine and its build machines - copy of the results in a common place to allow early view of the software stack The nightly build framework is written in python for portability and it is easily extensible to accommodate new build procedures.

project managers

web configuration

acrontab

The process of nightly builds and tests **STARTS HERE** with the acron jobs (and Task Scheduler on Win.).

Several minutes after midnight, each night, the AFS directories and directories for local builds are being cleaned on all build machines.
The next 'job' starts 'slots' to be build one by one, for platforms set in the slots' configuration.

Four times per hour, summary webpage generation is also launched to keep logs available on the web up to date.

The Nightly build and test system for LCG AA and LHCb software (core)

universal plugin for LHCb software (LHCb, Lbcom, Boole, Rec, Analysis, ...)

dedicated plugins for LCG AA projects:
ROOT   POOL
RELAX   COOL
CORAL   GAUDI

Dedicated build servers running SLC4/SLC5, Windows Server 2003, OS X 10.5 are taking care of software builds from midnight to late morning 7/7 days.

SLC4/5, Win, MacOS

software developers

physics software
cvs/svn repositories

Software is organised in a set of CMT packages and all configuration tasks are done by CMT (Configuration Management Tool - http://www.cmtsite.org)

CMT actions

checkout

configuration

build nodes

After compilation, software is installed in the shared directories on AFS. Compiled versions from the last seven days are always available.

AFS

@

build summary

build logs

test logs

test summary

build

tests

installation

Per subdirectory locking mechanism is used for copying to avoid writting the same file (e.g. source files which are common on all machnes) at the same time.

Project	Version	slc4_amd64_gcc34 [Tue Mar 17 07:45 2009]		slc4_ia32_gcc34 [Tue Mar 17 09:25 2009]	
LHCb	LHCB_v26r3	build	tests	build	tests
Lbcom	LBCOM_v6r27	build	tests	build	tests
Boole	BOOLE_HEAD	build	tests	build	tests
Rec	REC_v6r3	build	tests	build	tests
Brunel	BRUNEL_v34r3	build	tests	build	tests
Phys	PHYS_HEAD	build	tests	build	tests
Hlt	HLT_HEAD	build	tests	build	tests
Online	ONLINE_HEAD	build (51)	tests	build (50)	tests (1)
Analysis	ANALYSIS_HEAD	build (4)	tests	build (4)	tests
DaVinci	DAVINCI_HEAD	build (2)	tests (13)	build (2)	tests (13)
Panoramix	PANORAMIX_HEAD	build (2)	tests(1)	build (2)	tests (1)
Bender	BENDER_HEAD	build	tests	build	tests
Moore	MOORE_HEAD	build	tests (1)	build	tests
Gauss	GAUSS_HEAD	build (173)	tests	build (173)	tests
Panoptes	PANOPTES_HEAD	build (10)	tests	build (10)	tests
Alignment	ALIGNMENT_HEAD	build (6)	tests	build (6)	tests

Platforms not ready: slc4_amd64_gcc34_dbg, slc4_ia32_gcc34_dbg

100   tests total
6 (6%)   tests FAIL
6 (6%)   tests UNTESTED
88 (88%) tests PASS

COOL on i686-slc5-gcc34-dbg (from slot dev)

□ Other than PASSED or FAILED tests (Crashed, Unknown, etc.)
■ FAILED tests

Configuration possibilities:

- **several independent 'slots'** with different configuration (independent lists of projects in the slot)
- each project in the slot can be taken in any version available in cvs/svn repository (set by 'tags')
- each package in the project can have a version number set independently (requirements files changed on fly)
- 'head' tags can be used
- 'head of everything' option = 'head' version for each package in the project
- setup of the dependencies by project.cmt file modification
- CMTPROJECTPATH and CMTEXTRATAGS environment variables set by the system according to the configuration
- a slot can wait for another one to be finished before start (LHCb slots depend on LCG AA slots)