Current status and evolution of the LCG build system

Patricia Méndez Lorenzo
PH-SFT Group Meeting
19-Oct-2015

Outlook

- Introduction and first project
- The LCG build system
  - Technical structure
  - Build procedures
- Jenkins automation and visualization
- Future plans

Not all credits of this talk belong to me but to Pere, Benedikt, Dima, Ivan and the GENSER team

Introduction

- My start in the team: September 2014
  - Coming from the IT department:
    - *Grid experimental support*
    - *ServiceNow project*
- First project in SFT: Training in ROOT builds
  - Handling of external dependencies for ROOT builds
  - Generic and reusable approach:
    - *Structure used afterwards for the migration of GeantV and VecGeom to the Jenkins production instance*
- Next step: LCG builds and Jenkins management

ROOT builds with external dependencies

- Project goal:
  - Enable ROOT builds with all external packages
    - *Full package list identification:*
      - `/cmake/modules/SearchInstalledSoftware.cmake`
    - `"-Dfail-on-missing = ON"` enabled

- Project steps:
  1. External packages installation in AFS
     - `/afs/cern.ch/sw/lcg/app/releases/ROOT-externals`
  2. Setup of the cmake environment variables through specific scripts
  3. Procedure automation in Jenkins

Project description

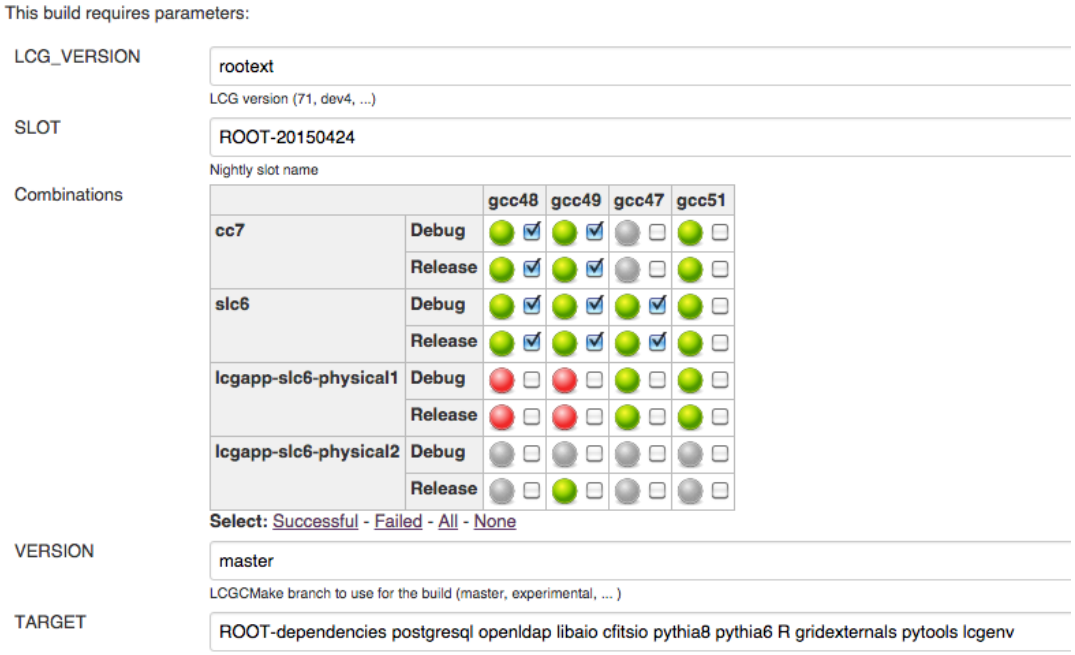
- Installation of packages in AFS
  - Reuse of the cmake structure used for the LCG external packages builds
- **`Setup.py`** script sets the cmake env variables for ROOT builds
  - `$PATH, $LD_LIBRARY_PATH` and `$CMAKE_PREFIX_PATH` including the external packages previously installed in AFS

```
CMAKE_PREFIX_PATH=/afs/cern.ch/sw/lcg/app/releases/ROOT-externals/ROOT-20150424/4suite/1.0.2p1/x86_64-slc6-gcc49-dbg:
    /afs/cern.ch/sw/lcg/app/releases/ROOT-externals/ROOT-20150424/Boost/1.55.0_python2.7/x86_64-slc6-gcc49-dbg:
    /afs/cern.ch/sw/lcg/app/releases/ROOT-externals/ROOT-20150424/Davix/0.4.0/x86_64-slc6-gcc49-dbg:
    /afs/cern.ch/sw/lcg/app/releases/ROOT-externals/ROOT-20150424/GSL/1.10/x86_64-slc6-gcc49-dbg:
    /afs/cern.ch/sw/lcg/app/releases/ROOT-externals/ROOT-20150424/Grid/dcap/2.47.7-1/x86_64-slc6-gcc49-dbg:
```

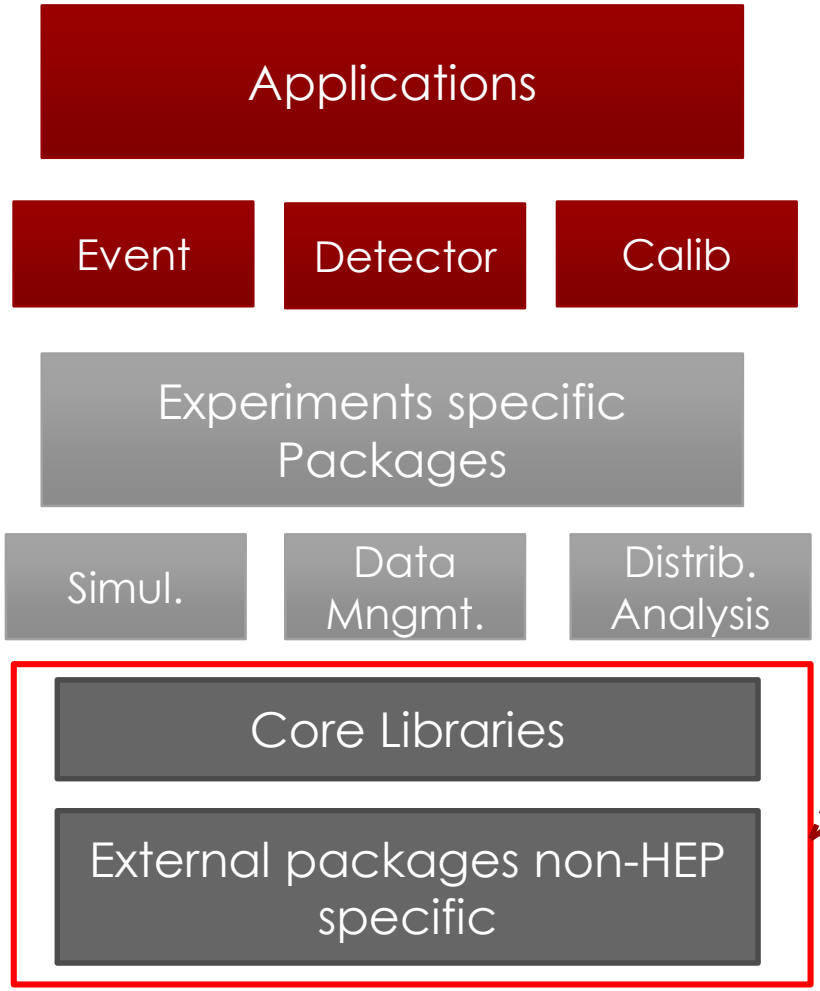
- Automation procedure via Jenkins for both ROOT builds and external packages installation
  - ROOT → env variable declared in Jenkins passing the location of the externals to setup.py
  - External PKGs build “a la” LCG_release reusing the same installation procedures

The project in Jenkins

- Concurrent installations enabled through different slot names
- Current available distributions:
  - `gcc48,gcc49,gcc51` -- `opt` and `dbg` modes -- `slc6` and `CentOS7`
- List of external packages configurable for each build
  - Latest slot contains 21 packages available in AFS
- Distribution to systems With no AFS access still pending
  - CVMFS end-system usage foreseen



Current project: LCG build system



We provide consistent releases to experimental and theoretical communities including both **external** and **LCG project** packages

Key aspects of the releases

1. Relocation, reproducibility and scalability
   - Wide and increasing range of packages, compilers, operating systems, architectures and end-systems

2. Quality of the built product
   - What we produce needs to be used by the experiments nightly builds

3. Unique and project-independent orchestration tool
   - The infrastructure needs to be automatic and useful for all Projects in SFT

4. Visualization and documentation
   - Ensure feedback/improvement cycle

Packages and combinations

Packages

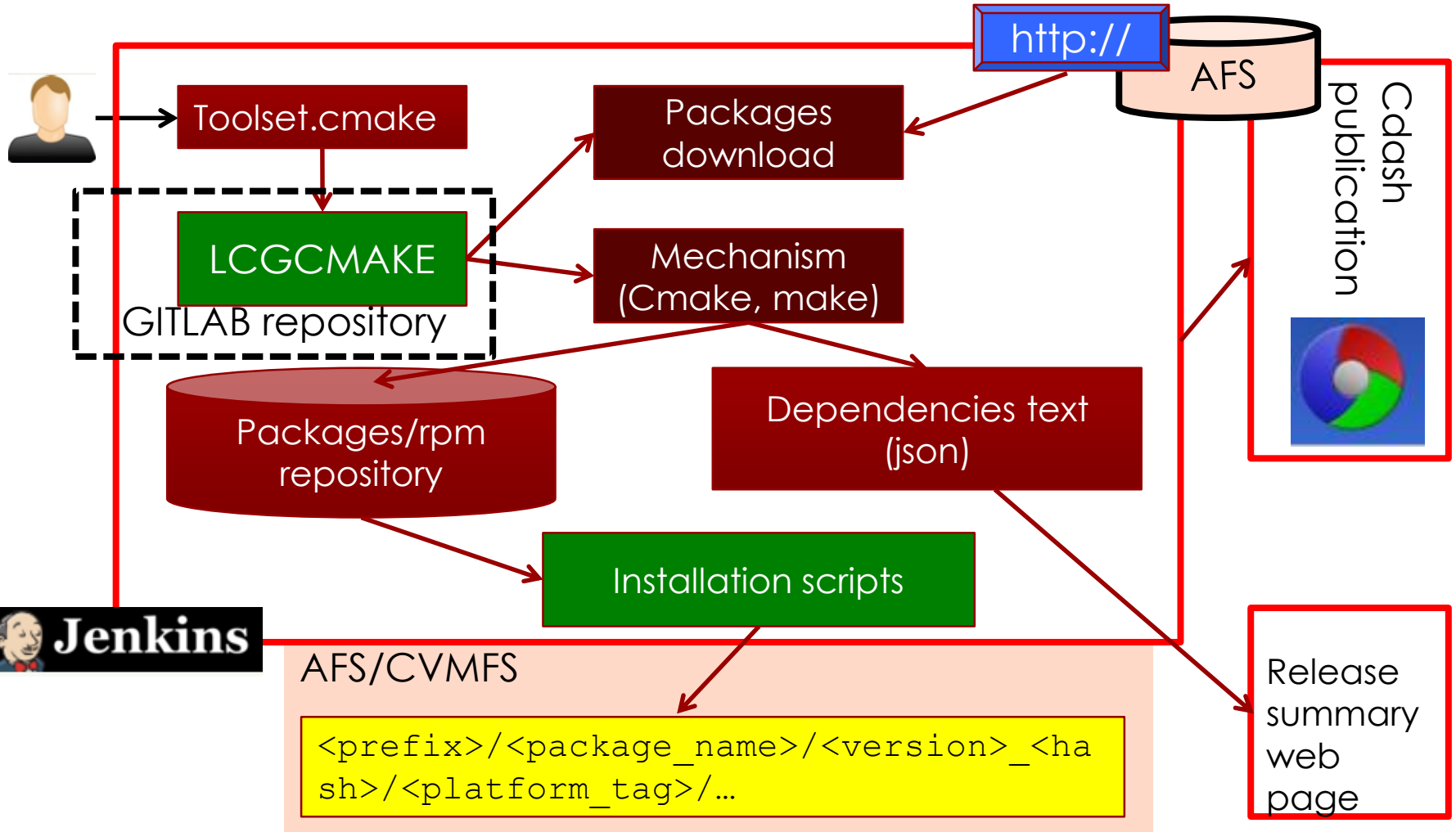
- 140 external packages including 10 Grid packages
- 50 MC Generators
- Two group projects: ROOT and Geant4

Packages and combinations agreed with experiments

Combinations

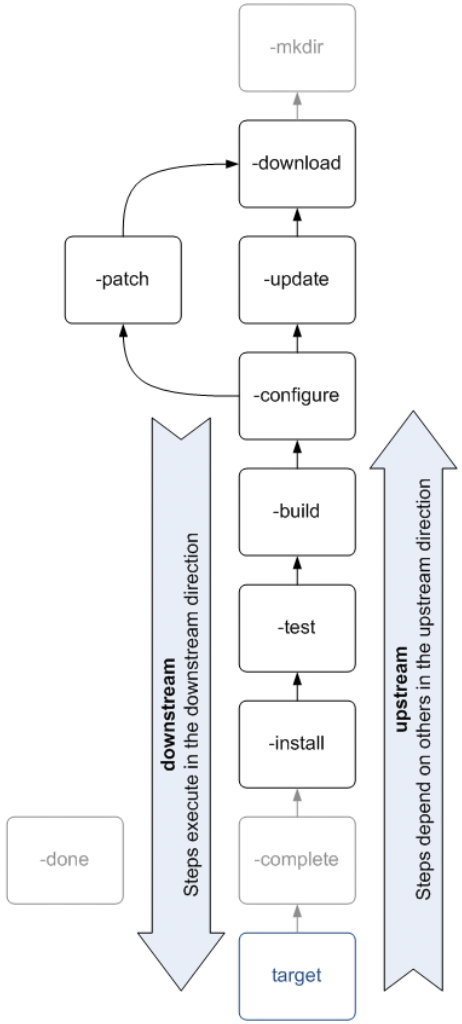
Compilers	Architectures	Operating systems	Build types
• gcc4.8.4, gcc4.9.3, gcc5.1.0, gcc5.2.0 • clang35, 37 • icc15 • native	• Intel • arm64	• SLC6 • CentOS7 • CentOS6 • Ubuntu12, 14 • Mac 10.8, 10.9, 10.10	• Release • Debug

The core elements



1. Reproducibility, relocation, scalability

- Building external packages from sources based on CMAKE
  - Open-source system dedicated to build, test and install software
    - *Build specifications included in platform-independent list files*
    - *Updated versions of the tool available in both AFS and CVMFS*
- SFT implementation via LCGCMAKE
  - Based on "*ExternalProject*" CMAKE module
    - *Enable the build of external software sources*
    - *Easy way to control package dependencies*
  - Repository recently migrated to GITLAB



Structure of LCGCMAKE

1. Declaration of the external packages and versions through `heptools` text files

```
# Application Area Projects
LCG_AA_project(COOL  3_0_4)
LCG_AA_project(CORAL 3_0_4)
LCG_AA_project(ROOT  6.04.02)
# Externals
LCG_external_package(4suite   1.0.2p1    )
LCG_external_package(AIDA     3.2.1      )
```

2. Modular `CMakeLists` build specification files

```
LCGPackage_Add(
  sqlite
  URL http://service-spi.web.cern.ch/service-spi/external/tarFiles/sqlite-autoconf-
sqlite_native_version}.tar.gz
  CONFIGURE_COMMAND ./configure --prefix=<INSTALL_DIR>
  BUILD_COMMAND  make
  DEPENDS Python
)
```

3. Set of scripts for automation purposes, end-systems installation, build maintenance and results publication

LCGCMAKE toolkit evolution

- Packages and builds
  - Introduction of 20 new packages most of them pytools related in 2015
  - Geant4 included as an additional project (latest LCG_80 release)
  - Split of the nightly builds in three different slots
    - *Including three supported ROOT6 versions: 6.02, 6.04, master*
  - In total 10 LCG releases have been provided in 2015

- Scripts
  - Automation of releases procedures included in Jenkins
    1. *Local build of external packages*
    2. *Copy and installation procedures in AFS and CVMFS*
    3. *Releases publication and documentation*

LCGCMAKE toolkit evolution (2)

- Build nodes maintenance: cleanup procedures
  - About 300 cores are shared for several jobs including nightly builds and releases, Geant4, GeantV and VecGeom
  - Control file created by each job at start time (parent dir level)

```
[sftnight@lcgapp-slc6-x86-64-21 ~]$ cd /build/jenkins/workspace/root-incremental-master/BUILDTYPE/Debug/COM
L/slc6
[sftnight@lcgapp-slc6-x86-64-21 slc6]$ ls
build  controlfile  root  rootspi  roottest
```

- *A lcgcmake job deletes the associated working directory (recursively) after 48h*
- *Procedure automatized via Jenkins and puppetized via a cron job.*

LCGCMAKE toolkit evolution (3)

- Procedures
  - Homogeneous Jenkins setups for both releases and nightly builds
    - *Same workspace area in Jenkins:*
      `<job_name>/BUILDTYPE/<type>/COMPILER/<compiler>/LABEL/<jenkins_label>`
      Simple creation of `install` and `build` directories with no platform references
- Publication procedures and visualization
  - Cdash visualization
  - After every LCG release a lcgcmake job triggers the creation of a json based PKGs dependencies file
    - *For PKGs documentation purposes*

Build results visualization in CDash

dev2

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
lcgapp-slc6-x86-64-7.cern.ch	dev2-x86_64-slc6-gcc48-dbg	2	0	1	0	0	0	1 +1	5 -1	13 hours ago
lcgapp-slc6-x86-64-17.cern.ch	dev2-x86_64-slc6-gcc48-opt	5	0	1	0	0	0	1 +1	5 -1	13 hours ago
lcgapp-slc6-x86-64-21.cern.ch	dev2-x86_64-slc6-gcc49-dbg	5	0	1	0	0	0	1 +1	5 -1	13 hours ago
lcgapp-slc6-x86-64-8.cern.ch	dev2-x86_64-slc6-gcc49-opt	5	0	1	0	0	0	1 +1	5 -1	13 hours ago

dev3

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
lcgapp-slc6-x86-64-22.cern.ch	dev3-x86_64-slc6-gcc48-dbg	5	0	1	0	0	0	1	5	13 hours ago

Testing started on 2015-10-16 02:55:34

Site Name: lcgapp-slc6-x86-64-7.cern.ch
Build Name: dev2-x86_64-slc6-gcc48-dbg
Total time: 10m 13s 310ms
OS Name: Linux
OS Platform: x86_64
OS Release: 2.6.32-573.3.1.el6.x86_64
OS Version: #1 SMP Fri Aug 14 10:45:09 CEST 2015
Compiler Version: unknown

1 tests failed.

Name	Status	Time	Details
permissions-test	Failed	1m 14s 440ms	Completed (Failed)

LCG Releases publication

LCG Software Elements

This page lists the various configurations of the LCG external and generator software stack provided by CERN's PH-SFT group to the LHC experiments. Details on the software distribution can be found on the websites of the LCG SPI and the Generator Services (GENSER) projects.

LCG Configurations

release ▲/ ▼	release date ▲/ ▼	description
79	Jul 15, 2015	root6 sufix dessapears from this release since this becomes the default setup
78root6	Jul 02, 2015	Version upgrade
77root6	Jun 10, 2015	Version upgrade
76root6	May 19, 2015	Version upgrade
75root6	Apr 18, 2015	Version upgrade
74root6	Feb 25, 2015	Version upgrade
73root6	Feb 25, 2015	Version upgrade
72root6	Feb 25, 2015	Version upgrade
72a	Feb 25, 2015	Version upgrade
71root6	Dec 11, 2014	Setting the contains of
71	Feb 25, 2015	Version upgrade

LCG Software Elements

Main | LCG Configuration 79

Compare with another configuration: ---

Description: root6 sufix dessapears from this release since this becomes the default setup
Release date: Jul 15, 2015
Platforms: x86_64-slc6-gcc49-dbg, x86_64-slc6-gcc49-opt, x86_64-slc6-gcc48-opt, x86_64-slc6-gcc48-dbg, x86_64-cc7-gcc48-dbg, x86_64-cc7-gcc48-opt, x86_64-cc7-gcc49-opt, x86_64-cc7-gcc49-dbg

Packages

Projects:

COOL	3_0_4
CORAL	3_0_4
LCGCMT	LCGCMT_79
RELAX	RELAX_1_4_2
ROOT	6.04.02

Databases:

Using the information provided at the end of the build configuration

	LCG Configuration 79		LCG Configuration 70
Databases	cx_oracle	5.1.1	
	genshi	0.6	
	mysql_python	1.2.3	
Graphics	pydot	1.0.28	
	pyqt	4.9.5	
Grid	cgsigsoap		1.3.5-2
IO	COOL	3_0_4	COOL_2_9_3
	CORAL	3_0_4	CORAL_2_4_3
	Frontier_Client	2.8.12	2.8.10
Math	Fastjet	3.1.1	3.0.6
	sympy	0.7.6	
XML	4suite	1.0.2p1	
	pyxml	0.8.4p1	

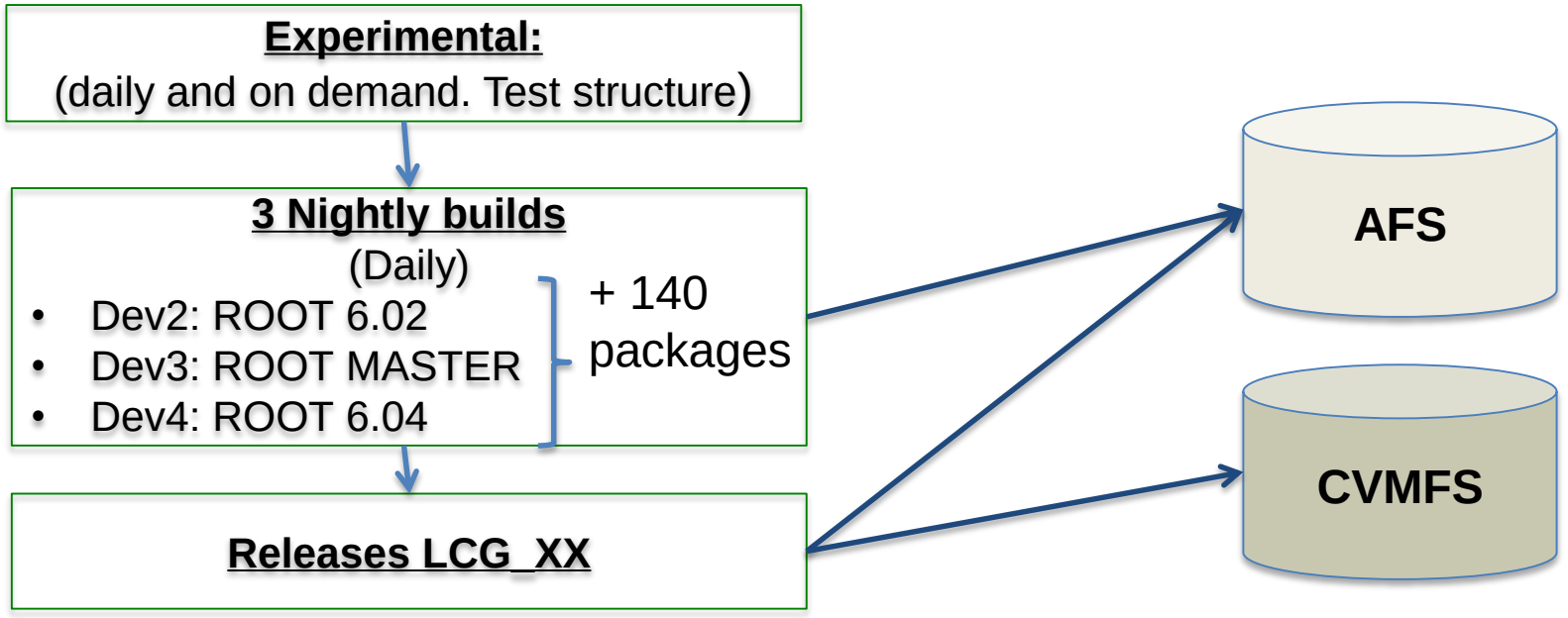
LCGCMAKE migration to GITLAB

- Migration completed in summer 2015
  - Transition period of 2 weeks included regular SVN-GIT synchronizations
- GIT has enabled a more modular and light lcgcmake setup:
  - SVN setup
    - *Increasing number of heptool files included in the system*
    - *Assignment of packages to each job at each CMakeList file*

```
---Addtional deopendencies to matplotlib----------
if((heptools_version MATCHES experimental) OR (heptools_version MATCHES dev4) OR (heptools_version MATCHES 76root6) )
set(pytools_extra_pkg pyzmq subprocess32 Jinja2 jsonschema terminado ptyprocess pandas pygments mistune prettytable)
endif()
```

  - GIT setup
    - *Experimental and testing branches enabled*
    - *Single heptool and specific CMakeList files per branch*
- Builds based on selectable branches implemented in Jenkins

2. Release Procedures



- Quality driven setup towards reliable LCG releases
  - Nightly build results copied to AFS for experimental usage/feedback
    - *Results used by the experiments*
    - *Follow an incremental approach*
  - From summer 2015 deployment of the LCG releases in CVMFS

AFS/CVMFS deployment

- Common installation tree in all builds and end-systems for binaries

`<prefix>/<package_name>/<version>_<hash>/<platform_tag>/`

  - hash → Calculated from the list of package dependencies and the version
  - Platform_tag → combination of architecture, OS, compiler version and build type: x86_64-slc6-gcc49-opt…

- Nightly build approach
  - Jobs automatically scheduled by Jenkins (daily basis)
  - Single copy to AFS at the end of the build
  - Builds executed “against” the release area
    - *Soft-links setup for already existing packages*
- LCG Releases approach
  - Jobs manually triggered on demand
  - RPMs creation and installation in both AFS and CVMFS areas
  - Maintenance of all installed RPMs in AFS
  - Management of installed RPMs through a specific DB

Summary of the builds

	Compilers						OS					Release type	
BUILD CHAINS	gcc48	gcc49	gcc51	clang35	icc15	native	CentOS7	SLC6	mac1010	mac109	ubuntu14	Release	Debug
Experimental	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
dev2	✓	✓		✓		✓		✓			✓	✓	✓
dev3	✓	✓				✓	✓	✓			✓	✓	✓
dev4		✓	✓			✓	✓	✓			✓	✓	✓
Releases	✓	✓					✓	✓				✓	✓

3. Orchestration

- **Jenkins** toolkit used for project-independent builds
  - *Master (web interfaced) - slave (build nodes) based architecture*
  - *Electric commander replacement*

- We manage our own PH-SFT Jenkins instance:
  - Master: SLC6 Physical node (puppet managed)
    https://phsft-jenkins.cern.ch
  - At the end of 2014 we asked for a Jenkins service provided by IT for all experiments including us
    - *No agreement achieved, each community has set their own instance*

Summary of the slaves

Machines	Number of cores
SLC6	215
SLC5	4
Centos7	40
CentOS6	2
Ubuntu12	>8
Ubuntu14	>16
Windows7	8
Mac10.8	8
Mac10.9	24
Mac10.10	24
Mac10.11	8
arm64	16
Fedora20/21/22	4/2/2
Cuda7	16
Xeon Phi	16

2014-2015 activities

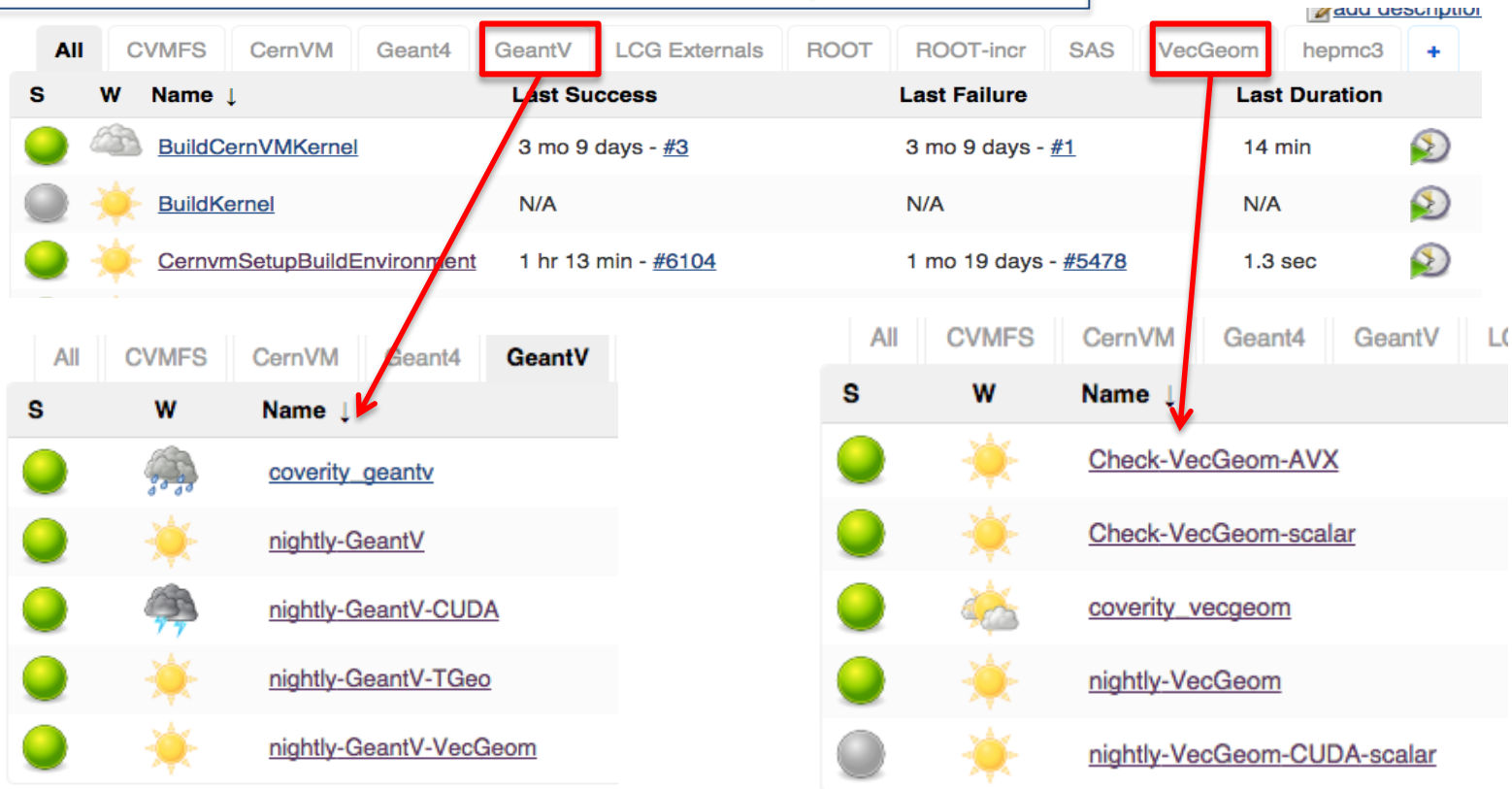
- Migrate Jenkins from a test to a production instance:
  1. Provide SSO protection and open it outside CERN
     - *Any CERN user has read access (NICE account)*
     - *Restricted set of Jenkins users (24 users)*
     - *https access outside CERN*
  2. Set up of a Jenkins test instance
     - *New plug-ins and big versions tests*
  3. TSM backup for the server production instance
  4. Interface to cdash for visualization purposes
     - *Summer student task in 2015*
  5. Implementation of new “exotic” build flavors (arm64, power8)
  6. Email notification improvement (ongoing)
  7. Version 1.626 updated in August 2015 “almost” transparent
- Deprecate Electric Commander:
  - Old nodes activity cleanup

In general a very stable and scalable service

Latest projects in Jenkins

- GeantV and VecGeom have been migrated to our Jenkins production instance

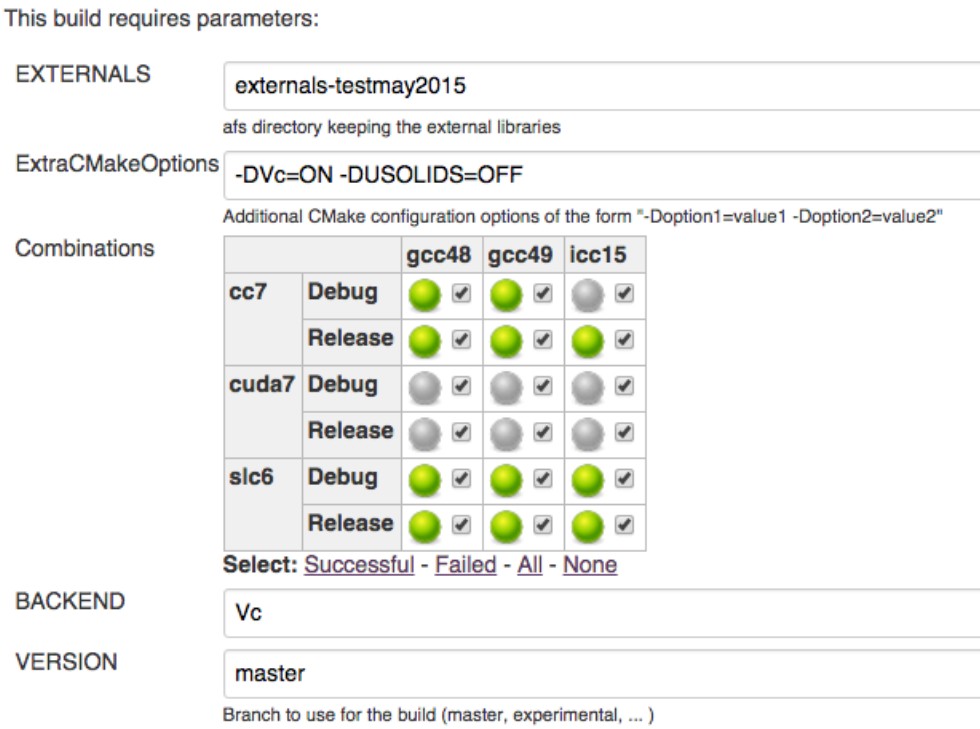
Thanks a lot to Oksana for her help here



GeantV and VecGeom

Identical migration approach for both projects following the ROOT external PKGs builds method

- Implementation of the externals packages in lcgcmake and installation in AFS via a Jenkins job
  - *The same installation procedure has been also enabled for Geant4 eternal packages*
  - Both projects have been migrated to GITLAB
  - *Jenkins structure included in both projects*
  - *setup.py scripts created to set the cmake environment variables pointing to the AFS area*
  - Builds provided in gcc48, gcc49, icc15 in slc6 and CentOS7



Future plans

- CVMFS fostering
  - Enable ROOT external packages builds in Ubuntu or MAC
  - Provide nightly builds and releases in these systems
- Improve the email notifications in Jenkins
  - Ongoing for some ROOT jobs
- Provide a more modular release infrastructure for experiments
  - Experimental views approach
- Platform independent Grid packages builds
- Consolidate GeantV and VecGeom in the system

Backup slides

Nightly builds approach

Following experimental requests we have relaxed the copy to AFS policy declared for each nightly build

- Partial copy of successfully packages builds to AFS enabled

```
#!/bin/bash -x
source lcgcmake/cdash/jk-setup.sh $BUILDTYPE $COMPILER $SLOT
lcgcmake/cdash/isDone.sh 0
lcgcmake/cdash/jk-runbuild.sh
error_value=$?
lcgcmake/cdash/cdashPublication.sh
lcgcmake/cdash/copy_to_afs.py $SLOT $WORKSPACE    #
####lcgcmake/cdash/isDone.sh 1
if [ $error_value -eq 0 ]
then
  echo "Successfully build"
  lcgcmake/cdash/isDone.sh 1
  exit 0
else
  echo "THE BUILD HAS FAILED **********"
  lcgcmake/cdash/isDone.sh 2
  exit 1
fi
rm -rf $WORKDIR/build
rm -rf $WORKDIR/install
```

Cdash-Jenkins interfaced created by Vaggos this summer

We have relaxed the copy policy to AFS in case of individual PKGs build errors

Release builds approach

Four manual Jenkins jobs automatize the release procedure

1. Packages builds as in the nightly builds

```
export TEST_LABELS=Release
export LCG_INSTALL_PREFIX=$LCG_INSTALL_PREFIX
export PYTHONUNBUFFERED=true
export LCG_IGNORE=$LCG_IGNORE
. lcgcmake/cdash/jk-setup.sh $BUILDTYPE $COMPILER $LCG_VERSION
lcgcmake/cdash/jk-runbuild.sh
cd $WORKDIR/install
$WORKSPACE/lcgcmake/cdash/package_release.py $WORKSPACE $WORKDIR $LCG_VERSION $PLATFORM $TARGET $BUILD_POLICY
```

   - *Creation of the summary files for the release and creation of the RPMs for both externals and MCgenerator packages → Local operation at the build machine*

1. Installation in AFS
   - *Copy of new RPMs to AFS and RPMs installation procedures*
2. Installation in CVMFS included in Jenkins
   - *cvmfs-sft machine has became a slave in Jenkins*
   - *Sftnight (Jenkins user) has now access to this machine for publication purposes*
3. Population in lcgsoft using the dependencies files created during the build process