System Intregration Testing

Gunter Folger
CERN/PH/SFT
Geant4 workshop 2009/ Catania

Overview

▶ Move of Geant4 integration testing from TINDERBOX to SPI nightly build system

  ◦ Pop versus push

  ◦ Bonsai untouched

▶ Recent changes in nightly build system

  ◦ Server approach with goal to run several "jobs" in parallel

  ◦ Geant4 contributed new display of results

▶ Future planned changes

  ◦ More involvement of developers

    • More frequent tests with direct feedback

  ◦ Shifts of running testing?

What has changed for Geant4 developer?

- Did you notice a change

  ◦ Compilation warnings no longer tolerated

  ◦ More examples included in testing

  ◦ MAC fully integrated

  ◦ Windows still requires work

Move to SPI nightlies (1)

- Supported platform include Linux, Mac, and Windows
  - Previous setup worked for Linux, but Mac and Windows not intergrated
- Regular build on set of machines once per night
  - Machine pulls information on what to build
- Build is completely on local disk
  - Source is checked out by each machine, possibly more than once
  - Performance, and less affected by afs problems
    - Network timeout, quota,...
- Support for testing different configurations on one machine
  - Currently two:
    - Current selected tags
    - Reference tag, patch, release, as soon as preliminay tag exists
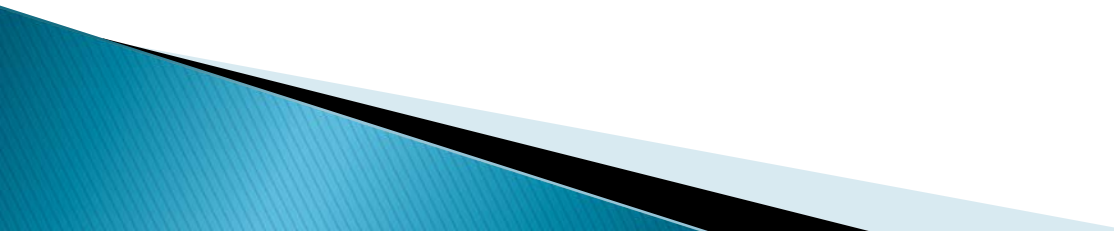   - Started to test multiple compilers on SLC5

Move to SPI nightlies (2)

- Build configured via CMT
  - was setup scripts
  - Single configuration file for all systems, yet allows for specialisation.
- Standard Geant4 make based build system used to build libraries

Move to SPI nightlies (3)

- Test tool is QMTest
  - Tinderbox no longer in use (~March 09)
  - Setup in cvs under tests/tools/qmtest
  - Based on python, easily extended
  - Allows to easily run tests, examples, but also benchmarks, without using cryptic symlinks from test to examples.
  - A test consists of 2+ tests
    - Build the executable
      - compilation warnings flagged as FAILURE
    - Run the executable with possible different input file(s) → multiple tests for same executable
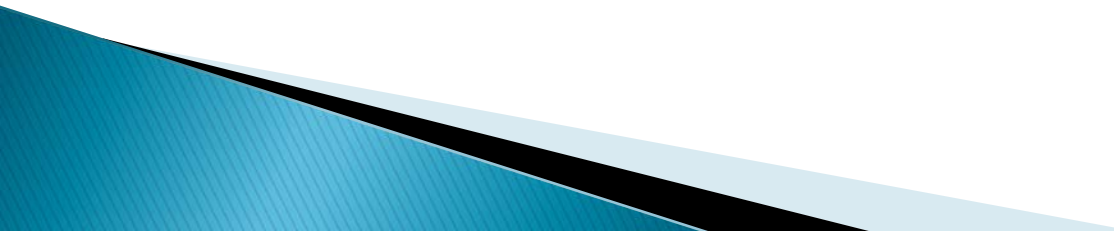
Move to SPI nightlies (4)

- Log files copied to web server

- parsed and available from
  http://lcgapp.cern.ch/spi/cgi-bin/nightlies

- Or

- http://lcgapp.cern.ch/spi/cgi-bin/nightly.py

Inventory of examples tested

▸ See also

http://geant4.cern.ch/collaboration/working_groups/testing/tests.shtml

Improvements of SPI nightlies(1)

- Original SPI display of results difficult to analyse
  - Failing tests difficult to correlate across platforms

- Addressed by storing results in database
  - Offering lots of possibilities to view results
  - Web page allows to analyse failures across platforms, days, …
  - Work of technical student Victor Diez Gonzales

Improvement of SPI nightlies(2)

▸ Distributions of builds to machines completely revised

- A server distributes build to Client machines

  · Capabilities of client in central configuration file

  · Server may run several build in parallel, if resources allow

    · Number of CPUs available and allowed to use

Summary

▸ Migration to SPI nightlies successful

▸ Effort to run testing significantly reduced

◦ Select tag in bonsai

◦ Check results

◦ Accept/reject tags, eventually debug on build machine

▸ What is next? Ideas:

◦ Add yet more tests, e.g. benchmark directory

◦ Automatic generation of stack trace for crash

◦ Create tests database