CMS Software Installation

Peter Elmer
Princeton University
WLCG Asian Tier-2 Workshop
03 Dec, 2006

CMS Offline Software

- The CMS offline software consists of several "top-level" packages you may want to install yourself:
  - CMSSW – framework, simulation, reconstruction, etc.
  - PHEDEX – data transfer/placement tools
  - PRODAGENT/PRODMGR/PRODREQUEST – Production workflow management system
  - CRAB – User tools for grid submissions
- Plus some number of dependencies written by CMS
  - SCRAM/IGUANA/DBS/DLS/BOSS....
- And some number of external dependencies (next slide)

CMSSW external depedencies

- CMSSW uses a large number of external packages:

  boost, bz2lib, castor, clhep, cmake, coin, cppunit, curl, db4, dcap, doxygen, elementtree, expat, frontier_client, gcc, gccxml, gdbm, geant4, glimpse, gpt, graphviz, gsl, gsoap, hepmc, heppdt, hippodraw, iodbc, libjpg, libpng, libtiff, libungif, meschach, mimetic, mysql, mysqlpp, openssl, oracle, oval, p5-dbd-oracle, p5-dbi, p5-libwww-perl, p5-template-toolkit, p5-uri, p5-xml-parser, pcre, python, qmtest, qt, qutexmlrpc, sigcpp, simage, soqt, sqlite, tkonlinesw, uuid, valgrind, xdaq, xerces-c, zlib, SCRAM, CORAL, GENSER, POOL, ROOT, SEAL

- PHEDEX/PRODAGENT, etc. have further external dependencies

Installation method

- CMS builds and distributes each of these packages as standard rpm's (i.e using the “redhat/rpm package manager”)
- As the number of packages to install (and versions!) is so large, we use apt (the “advanced packaging tool”) to manage the dependencies for you such that you can specify only the top level package you want and its version (and the rest of the dependencies are installed automatically for you, with their correct versions).

Architectures/platforms

- Currently we support SL(C)3 as an operating system for CMSSW with the following requirements:
  - You should install SL(C)3 software via an SL(C)3 machine
  - You should build/compile SL(C)3 releases on SL(C)3 machines
  - Things built on SL(C)3 will run on both SL(C)3 and SL(C)4
  - RHEL and other RHEL-rebuilds should be equivalent, however using SL(C) is preferred.
- During the CMSSW_1_2_0_preX releases work has been done to do the port to SL(C)4 and gcc345. We expect that we will have SLC4/gcc345 builds by one of the early CMSSW_1_2_x releases.

Other architectures/platforms

- Regarding 64-bit machines:
  - The binaries/libraries should work on 64-bit machines if you have the necessary 32-bit compatibility libraries
  - We expect to support building 32-bit executables on 64-bit machines (perhaps during the CMSSW_1_3_x release series)
  - Preliminary work has been done to port to native 64-bit builds, but we currently do not have a schedule for this.
- Preliminary work has also been done to port CMSSW to MacOSX, but this has not yet converged. We don't have a schedule for this. (If someone is particularly interested in this and has experience porting software, help would be welcome!)

Extra OS requirements

- Although we distribute almost everything we need ourselves into the CMS SW area, we also expect a couple of non-standard things from the SL(C)3 operating system installation:
  - apt
  - rpm-build
  - (rpm is usually there by default)
- These must be installed as part of the OS install. In the future we will provide a means to avoid this, but for now they are required.

Additional information

- While we use rpm for the SW installation, the rpm database for CMS is ***separate*** from the operating system one, i.e. we use the same tool, but not the same database.
- If you are installing software yourself, you should do this as some unix user *other than* the root user (see also later slide about grid installation and CMS Tier-1/Tier-2 sites)
- Typical sizes after installation:
  - A single CMSSW release + dependencies ~ 4.7GB
  - Recent CMSSW release (no dependencies) ~ 1.8GB
  - The corresponding rpms are smaller due to compression

Initial creation of a SW area

- You must choose some directory to install the software, this will be your “SW area”, say /mnt/x/y/z
- Then you need to do the following one time (when you first create the SW area):

- setenv VO_CMS_SW_DIR /mnt/x/y/z
- wget -O $VO_CMS_SW_DIR/aptinstaller.sh http://cmsdoc.cern.ch/cms/cpt/Software/download/cms/aptinstaller.sh
- chmod +x $VO_CMS_SW_DIR/aptinstaller.sh
- $VO_CMS_SW_DIR/aptinstaller.sh setup -path $VO_CMS_SW_DIR

(Note the line wrap-around in the 2nd command)

Installing software

- The previous slide showed commands you only need to do once. The following two commands are necessary every time you want to update the software area (in the shell where you will do the update):

- eval `$VO_CMS_SW_DIR/aptinstaller.sh config -path $VO_CMS_SW_DIR -csh`
- apt-get update

- After this you can peform all of the normal apt commands, see next slide.

Installing software - 2

- See a list of all rpms available
  - apt-cache pkgnames | grep '^(cms|lcg|external)'
- Search for some keyword in the full list of rpms (e.g. CMSSW)
  - apt-cache search CMSSW
- Install a release (and any necessary dependencies)
  - apt-get install cms+cmssw+CMSSW_1_1_2
  - This will print a list of the rpms that will be installed, the size of the rpms to be downloaded and the size after installation and request confirmation. If “y”, it will install the release.

Users of the SW area

- The previous slides described what the person administrating the SW area must do to install releases, etc.
- For a user of the SW area, the following things must be in the users login scripts, for tcsh

  setenv VO_CMS_SW_DIR /mnt/x/y/z

  source $VO_CMS_SW_DIR/cmsset_default.csh

  or for bash:

  export VO_CMS_SW_DIR=/mnt/x/y/z

  source $VO_CMS_SW_DIR/cmsset_default.sh
- With this in the login, normal scramv1 commands can be issued.

Extra requirements for Tier-2's

- The previous slides described how a site admin or individual can install the software
- For CMS Tier-2 sites, the computing model is that they provide resources for MC production as well as provide analysis resources to the collaboration. Thus we have an explicit policy regarding software installation in these sites:

  https://twiki.cern.ch/twiki/bin/view/CMS/OfflineSWDistributionPolicy
- The main things to consider is that we expect Tier-2 sites to provide the current set of production releases in their SW area and prefer (if possible) that SW installation be done centrally.

Squid cache for apt repository

- The cms apt/rpm repository is at CERN, and thus your software downloads will pull rpms from CERN.
- We have set up and are testing a squid cache of the apt repository at FNAL. The first person who accesses the cache to download a release will trigger the caching in the machine hosting the cache. Subsequent downloads will happen from the cache and thus be significantly faster.
- We are still working through some problems with this, but eventually we should document the setup necessary.

Documentation source

- The Twiki page regarding software installations is:

  https://twiki.cern.ch/twiki/bin/view/CMS/CMSSW_aptinstaller

- Although I have copied some things in the documentation page to this tutorial presentation, you should look at that Twiki page to find the most recent instructions, plus more information not included here.

- If you have questions, the HyperNews forum to use is:

  "Software Distribution Tools Discussions"

  https://hypernews.cern.ch/HyperNews/CMS/get/softwareDistrib.html

  hn-cms-softwareDistrib@cern.ch