

CMS Software Deployment on OSG

Bockjoo Kim¹, Michael Thomas², Paul Avery¹, Frank Wuerthwein³

1. University of Florida, Gainesville, FL 32611, USA
2. California Institute of Technology, Pasadena, CA 91125, USA
3. University of California at San Diego, La Jolla, CA 92093, USA

bockjoo@phys.ufl.edu, thomas@hep.caltech.edu, avery@phys.ufl.edu, fkw@fnal.gov

Abstract. A set of software deployment tools has been developed for the installation, verification, and removal of a CMS software release. The tools that are mainly targeted for the deployment on the OSG have the features of instant release deployment, corrective resubmission of the initial installation job, and an independent web-based deployment portal with Grid security infrastructure login mechanism. We have been deploying over 500 installations and found the tools are reliable and adaptable to cope with problems with changes in the Grid computing environment and the software releases. We present the design of the tools, statistics that we gathered during the operation of the tools, and our experience with the CMS software deployment on the OSG Grid computing environment.

1. INTRODUCTION

The CMS (Compact Muon Solenoid) experiment [1] is a high energy physics experiment located in an underground cavern at the Large Hadron Collider (LHC) at CERN, Geneva, Switzerland [2]. The experiment is an international collaboration to run the complex detector and take data with about 2000 scientists and engineers from 160 institutions in 37 countries.

The CMS experiment has been releasing its software with new framework to facilitate its use by collaborators since the end of 2005. The CMS experiment employs a tiered distributed computing model based on the Grids, the LHC Computing Grid (LCG) [3] and the Open Science Grid (OSG) [4]. There are approximately 37-tiered CMS centers around the world.

Because of fast software development cycle and the need for the immediate deployment to the CMS centers, a set of software deployment tools has been developed for the installation, verification, and removal of a release. The tools are mainly targeted for deployment and removal of the CMS softwares on the OSG site with the capability of the instant release deployment, corrective resubmission of the installation jobs. We also use an independent web-based deployment portal with the Grid security infrastructure login mechanism.

We present the design of the tools. Also we discuss statistics that we gathered during the operation of the tools, and our experience with the CMS software deployment on the OSG Grid computing environment.

2. CMS SOFTWARE LIFE CYCLE

The CMS software life cycle goes through a typical software life cycle. There are two types of CMS software installation. One is the installation by individual collaborators and the other type is the

installation that is deployed through the Grid. Since the latter type of installation requires more careful installation management due to the fact that many different collaborators can access the installation, the removal procedure requires a scheduled procedure to remove a release if the release is deprecated. The cycle can be described as follows:

- **Development and Build:** CMS software developers develop each component of the CMS software and tests building libraries.
- **Packaging:** Once CMS software is developed and proved to be built successfully, the CMS software packaging tool team packages the software using the RPMs and the APT packaging tools and repository.
- **Tagging a Release:** Then, the package is tagged to allow the installers to discover the new releases using XML.
- **Release Announcement:** The release is actually announced by a release announcement email.
- **Deployment the Release at Sites:** Now, the release is officially ready to be deployed and the CMS software is deployed at various sites
- **Tagging for a Deprecation Release:** When a release becomes too buggy or can be replaced with a newer release(s), the release is tagged as the deprecated release again using XML.
- **Release Deprecation Announcement:** The deprecated release is announced for the collaborators to inform the deprecated release will be removed within a certain time.
- **Removal of Deprecated Release:** If there is no objection from the collaborators, the release can be removed from the installation. Sometimes, if there are works that use the deprecated release, the collaborator(s) can request the extension of the life of the deprecated release.

3. CMS SOFTWARE DEPLOYMENT AND REMOVAL TOOL

The Grid can provide computing resource and storage with great flexibility. Because of this fact, application software can be installed without publication and notification. This can cause multiple copies of identical application software to be installed by different grid users.

If software installation could be organized and managed in a consistent and maintainable manner, one could share the installation responsibility within the collaborators who require the same application software and efficiently use the grid resources.

3.1. CMS Software Packaging Tool

CMS employs APT packaging for software distribution and provides packaging tools for initial setup and update. RPM packages are provided as the part of the APT packaging. The tool sets up a non-root area where RPMs are installed. The packaging team provides the release publication in an XML form so that available releases can be checked periodically for the automated installations.

3.2. Design of the CMS Software Deployment and Removal Tool

Grid jobs can be submitted through either the LCG/EGEE user interface (UI) or Condor-G[5] to sites that are involved with the CMS experiment. For OSG sites, jobs are typically submitted through Condor-G.

The CMS software deployment tool consists of a set of scripts that submit the installation and the verification jobs through Condor-G, and are customized according to the CMS packaging tool. Figure 1 shows a diagram of the CMS software deployment tool design for OSG sites.

All installations and verifications are recorded in a database kept at a central location from which the installation and the verification jobs are submitted. Information about each OSG sites is maintained in a separate database called the GridCat. GridCat pulls information from a list of registered OSG sites and maintains the information in its backend database.

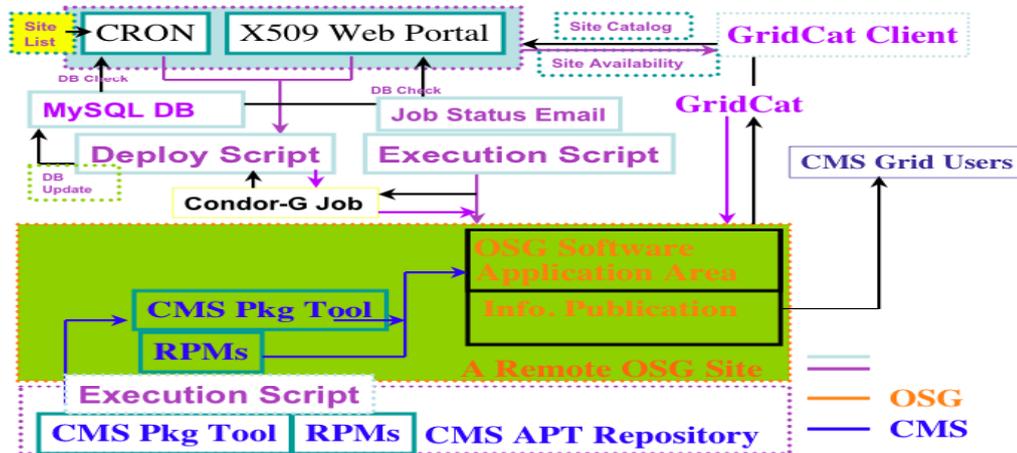


Figure 1 A diagram of the CMS software deployment tool design for OSG sites.

There are two possible ways to install the CMS software using the tool: the installation and the verification using the grid proxy based web portal or the scheduled execution of the deployment script. The tool checks if there is any deployment that is already in progress to prevent any race condition among different functionality. This is a useful feature in the tool design because sometimes there can be a different person who are installing the software at the same time. This also helps track who submitted the installation and the verification jobs.

The deployment using the web portal takes a grid proxy for an authentication, to log on to the web portal, and to execute the installation and the verification job submission. The web portal is based on the CGI with the c-wrapper and the python scripts behind the c-wrapper. Since the python scripts are the dynamic part of the CGI, it is very easy to add new functionalities through python scripts and execution shell scripts. At the moment, there are mainly the installation, the verification and DB status update functionalities. Deployment operation using the web interface is found to be very convenient in debugging new emerging OSG sites to validate the initial software deployment.

The second mode of the deployment operation on OSG is the scheduled script execution using the CRON utilities in the OS. In this mode, a list of well-established OSG CMS sites is kept in a file. When there is a new production CMS software release which is published in an XML format through a web page, the scheduled script checks the new release and executes the deployment tasks until the tasks are completed at all the listed sites. If the task fails at a certain stage, the expected correction is applied and the task is executed again. For example, if a site loses a connection briefly, the task will fail, but at the next scheduled execution of the task, the database is updated to a proper state of operation and the task is executed again.

The software that is released has a finite lifetime. Some releases have to be removed due to available disk space or when the releases become outdated or not usable because of bugs in the release. When removing such releases, great care must be provided due to dependencies that may impact other usable releases. Therefore, the release removal script has been carefully tested for a long period of time and is still being tested. Currently, the removal script checks for the deprecated release in the apt-cache, calculates the dependency, checks if any of the dependencies do not belong to any survived releases, removes the removable dependent packages, removes the release, and updates the database. For the removal script, we have also automated the scheduled execution of the removal script, but we do not run scheduled scripts yet.

All the job execution logs are kept at a place where one can access the log files from the web browser. Whether deployment tasks are successful or now, the status of the deployment tasks are notified to the deployment operator by email. Notification is sent to the operator in one of three cases: successful task completion, task held in a batch queue, or the task failure for some reason. The email body contains information on whether the task was failed or succeeded, where to find the log, what is the name of the site, and at what stage the task failed.

4. CMS SOFTWARE DEPLOYMENT AND REMOVAL ON OSG

During 2006 and 2007, we have deployed more than 500 installations that involve 33 OSG sites and 64 releases. We also remove approximately 170 installations. Table 1 shows a summary of the CMS software deployment and removal statistics during 2006 and 2007.

Table 1 A summary of the CMS software deployment and removal statistics during 2006 and 2007.

Installed Releases	0_4_1 TO 1_6_0 64 Releases	
Sites	OSG T2, T3, and Opportune 33 sites	
Installations (ReleasesxSites)	2006	2007
	126	380
	506	
Removals (ReleasesxSites)	2006	2007
	15	158
	173	

The CMS software has been deployed with the scheduled mode of operations at 12 of 33 OSG sites, as these are mainly CMS dedicated facilities. The other sites are either CMS Tier3 sites or non-CMS opportunistic sites to explore the possibility of running the CMS jobs. Table 2 shows a summary of the CMS software deployment and removal statistics at OSG sites where scheduled deployment was executed during 2006 and 2007.

Including external packages, a single CMS software release contains between one and two million files. The size ranges between 30 and 60 GBytes. The time it takes to install each release depends on the network status, disk response, and mostly loads on the system. Typically, the installation time ranges between 30 minutes and one hour.

Although there are not many failures of the installation jobs, the failures are intermittent or persistent. Those that are intermittent are fixed by resubmitting jobs and are due to the heavy load in the gatekeeper, the NFS timeout, or temporary network outage. The installation jobs that are failing persistently are due to more fundamental reasons such as missing tools on the machine, e.g., apt-get, rpm, or rpmbuild, file system problem, or firewall problem. The APT packaging tool uses various locks and the file system can behave differently with the lock file. There are four different file systems on the sites where the scheduled installation is being performed. If the failure is due to the file system, lock files have to be relocated at directories in the local disk area.

Table 2 A summary of the CMS software deployment and removal statistics at selected OSG sites during 2006 and 2007.

Production Sites	Installations	Removals
Caltech	44	19
Florida	109	48
MIT	42	20
Nebraska	40	19
Purdue	39	20
UCSD	42	20
Wisconsin	42	23
SPRACE(Sao Paolo)	38	14
UERJ(RioDeJaneiro)	29	4
TexasTech	28	0
Total	442	187

The removal of a release has similar problems to the installation jobs. The most time consuming part of the release removal job is to determine dependencies. Also, various checks and backups have to be made to prevent any RPM database corruption, which can disable the calculation of dependencies. Fortunately, dependencies are identical for a given release and we use the dependency information created at a site to avoid the calculation step at other sites.

The installed release information is published using the OSG information system. This information is again used by the CMS site availability monitoring (SAM) [6]. The CMS SAM tests indicate whether or not a site is missing a required CMS software release.

5. SUMMARY

We have designed and developed a set of tools to deploy and remove the CMS software releases on the OSG. Using the tools with an appropriate scheduling, we have deployed (removed) more than 500 (170) CMS software installations across the OSG during 2006 and 2007.

Most installation problems are due to file system, slow disk response, network issues, or missing fundamental packaging tools. The process of removing a release is more difficult, but we managed to implement a procedure by recycling the dependency calculation and adding checks and backups. Once the problem is identified, the tools were used to resubmit jobs to deploy or remove the CMS software releases. In particular the capability of the scheduled resubmission was found to be very convenient because we could focus on fixing a particular problem at the site without worrying about an overlapping resubmission.

The implemented CMS software deployment tools for the OSG sites are shown to work very effectively. With these tools, we plan to extend our scheduled installation to further sites thereby conveniently providing CMS collaborators with more sites where they can utilize the installed CMS softwares.

References

- [1] D. Acosta et al., eds., "CMS Physics Technical Design Report", volume I. CERN, Geneva, Switzerland, 2006, <http://cmsdoc.cern.ch/cms/cpt/tdr>.
"CMS Outreach", <http://cmsinfo.cern.ch/outreach>.
- [2] "LHC – THE LARGE HADRON COLLIDER", <http://lhc.web.cern.ch/lhc>
- [3] L. Bird et al., eds., "LHC Computing Grid – Technical Design Report", CERN, Geneva, Switzerland, 2005, <http://lcg.web.cern.ch/LCG/tdr>, "Worldwide LHC Computing Grid", <http://lcg.web.cern.ch/LCG>

- [4] F. Wuerthwein and R. Pordes. “The Open Science Grid”, in *Computing in High Energy and Nuclear Physics*, CHEP06, T.I.F.R., Mumbai, India, 2006, “Open Science Grid”, <http://www.opensciencegrid.org>.
- [5] Condor Project, <http://www.cs.wisc.edu/condor/>
- [6] SAM, <https://twiki.cern.ch/twiki/bin/view/LCG/SAMOverview>