

Table of Contents

WLCG Operations and Tools TEG - WG3 Software Mangement, Software Configuration and Deployment Management.....	1
Application Software Management.....	1
Deployment Management.....	2
Configuration Management.....	2
Yaim.....	3
Puppet.....	3
Quattor.....	3
Additional Info.....	3
Contributors.....	3

WLCG Operations and Tools TEG - WG3 Software Management, Software Configuration and Deployment Management

Application Software Management

The software stack for LHC experiments is layered into several levels. The bottom being the operating system (e.g. Scientific Linux), on top of a set of common libraries which are used by the LHC experiments (e.g. ROOT, COOL, CORAL, Boost, mysql, Python, etc) and the top most layer being the experiment specific applications for data reconstruction, analysis and simulation. Essential needs for the experiments are

- possibility to have a fast turnaround of package versions and their grid wide deployment
- package version decoupling from operating system to allow later versions to be used than natively provided
- provide the same package version in addition to grid deployment on different OSes, e.g. on development platforms (Mac)
- possibility to fix the package versions of the whole stack which is essential for reproducibility

The common software layer for LHC experiments is being steered by PH/SFT through the "LCG Applications Area" where the set of common libraries and their versions is being discussed and defined. The topmost layer, the experiment specific applications, are being developed individually within experiments. The software management and configuration of these layers for experiments is being driven by several tools, some being used in common by multiple experiments.

... missing paragraph on grid middleware ...

Atlas and LHCb are using Cmt (<http://www.cmtsite.org>) as the main tool for software management. Cmt implements its own language through which it allows package versioning, package building and the setup of runtime environments. Usually 2 to 3 times a year the PH/SFT group is providing a so called "LCGCMT configuration" which denotes a baseline of ~ 120 common packages provided to experiments. The packages within such a configuration include the "self developed" projects (ROOT, COOL, CORAL), "external packages" - recompiled from source (e.g. gcc, Boost, Python, mysql, frontier) and grid middleware clients provided so far by gLite. Experiments are taking these LCGCMT configurations to build their specific applications on top. To allow an even faster turnaround of packages, LHCb has invented an experiment specific "LHCbGrid" configuration, which can overwrite the package versions and instructions provided by LCGCMT for grid middleware packages. One concern in this scenario is the future of grid middleware packages which were so far provided via gLite to PH/SFT and subsequently picked up by the experiments.

CMS uses rpm spec files which are derived from templates are augmented with version numbers and are being used for package building. A set of spec files containing a global tag denote a self contained set of compatible versions for the CMS software stack. A "cmsbuild" script will subsequently use one of these global tags to build the binary rpms which are used for later deployment. The advantage of this solution is the usage of a standard tool which is available and known in the development community for long time. The dependency management is also handled by rpm in a standard way. In addition to source rpms CMS is using scram for building their self developed projects (CMSSW)

Experiments that are decoupling the grid middleware software layers from their own distribution are experiencing problems, while experiments deploying grid middleware together with their software stack have not reported issues of that kind.

... missing paragraphs on gnu make, cmake - used by PH/SFT, LHCb, Alice ...

Deployment Management

A rather recent development in the area of software deployment is cvmfs, a centrally deployed filesystem that is distributed to remote users via levels of caches. The local cvmfs client will only download the files which are needed for performing its work, usually a fraction of the deployed software area. Software deployment is done by the librarians of the different VOs deploying their software to a "release node" from which the files are being replicated to a "Stratum 0" node. This is the root node for the software deployment. Several "Stratum 1" nodes are being attached to this root node (currently operational at Cern, Ral, BNL, coming soon at Fermilab, Taiwan) which provide additional copies of the software tree. The final replication to the sites is being done via squid caches to which the cvmfs client will talk to retrieve the necessary files and download them persistent into a locally mounted cache. Currently cvmfs is deployed on ~ 80 sites and hosts 15 volumes (~ 2TB) for LHC experiments and other VOs. Cvmfs provides server side monitoring and a nagios probe for sites for their local testing. The advantage of this system is that it facilitates the software deployment in a single place, therefore reduces the workload for software librarians and deployment people. In addition any changes made on the root Stratum 0 node are visible with very little delay on all the connected clients. This solution is currently being used in production by Atlas and LHCb and ~ 80 sites (T0/T1/T2) in production. Experiments and sites that are using CMVFS are happy with its performance as it reduced / removed issues with software installation thus reducing man power intensive work. It is generally seen as a lightweight, scalable service solving problems with software installation in individual site shared areas, thus reducing man power

LHCb in addition for non cvmfs sites uses tarball distribution, providing individual tarballs for the common software layers and experiment projects (Reconstruction, Analysis, MC, ...). The deployment of the software is being done via special SAM jobs with privileges to write into the sites' shared software area. This system provides an easy way of software deployment both on grid sites and individual user machines. It is quite old but has been proven to be effective enough for software deployment on grid sites.

CMS is using a modified version of rpm / apt, to allow execution in user space, which is being used for software deployment. This special apt provides its own rpm database which is disconnected from the operating system one. The deployment on grid sites is done via special jobs with lcgadmin role. CMS through the years has gained a lot of experience with this system and in general is happy with its performance. Some issues are being observed with rpm locks and changes of the grid middleware packages.

... missing torrent, packman

For grid services the processes of bug fixes, make them available, service deployment, including hardware request, service installation, is perceived as too slow. Also the fact that no baseline version of services across the grid is being enforced is seen as problem. The staged rollout of grid middleware packages as a community effort is seen as a scalable solution. Another issue with grid services are inconsistencies between different information services, that are needed by the experiments, e.g. GOCDB, BDII, CMS siteDB.

Configuration Management

6. Configuration management

what works well

- overall everything is working well.
- Recovery procedures handled by IT operators save operational efforts.
- The use of yaim allows to automatize processes in quattor, useful in some cases, for example for CRAB servers deployment.

what are the top three problems

- The main issues are related to the maintenance of the vobox budget e.g. the re- distribution of resources to new/existing services, dealing with Service Managers in order to get back machines and/or not increase the established budget, and the occasional delays when waiting for new machines in the hardware request process.

Yaim

Puppet

Quattor

Additional Info

Contributors

Jakob Blomer, Marco Clemencic, Andreas Pfeiffer, Marco Cattaneo,

-- StefanRoiser - 09-Nov-2011

This topic: LCG > WLCGTegOperationsWG3

Topic revision: r6 - 14-Nov-2011 - 09:53:21 - StefanRoiser



Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback