Table of Contents

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

Application Software Management

The software stack for LHC experiments is based on several layers. The bottom being the operating system (e.g. Scientific Linux), on top of it is 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
- decoupling of common library versions from operating systems 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 for LHC experiments is being steered by the "LCG Applications Area (LCG/AA)" where the set of 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.

Atlas and LHCb are using CMT 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 LCG/AA 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 "HEP self developed" projects (ROOT, COOL, CORAL), "external packages" - recompiled from source (e.g. gcc, Boost, Python, mysql, frontier) and grid middleware clients. 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. LCG/AA, LHCb and Atlas are currently investigating a possible move to CMake as a build tool. 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, augmented with version numbers and are being used for package building.` 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)

Alice is using cmake for their software building (AliRoot) and otherwise the natively provided build tools for other packages. In total ~ 150 packages are used for the Alice software stack. Dependency management is handled by the major packages (AliRoot). Binaries are made available for SLC5, Ubuntu and MacOSX.

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 (need to find reference!!!)

Works well:

- LCG/AA software repository, LCGCMT software configuration

Works not so well

- Removal of obsolete software (might get easier when cvmfs is everywhere, but I think we need a system to measure software popularity)

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" cvmfs servers replicate this root node (currently operational at Cern, Ral, BNL, coming soon at Fermilab, Taiwan), these in turn are the servers which sites connect to. The final replication to the sites is being done via squid caches through which the cvmfs client will retrieve the necessary file catalogs and files and download them 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 a nagios probe as well as a number of diagnostic tools for sites for their local testing. The advantages of this system include that it facilitates the software deployment in a single place, therefore reduces the workload for software librarians and deployment people, and further reduces internal network traffic between worker nodes and software servers and in many cases improves efficiency of jobs, particularly during the setup stage. 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. 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.

While CVMFS is working very well at sites that use it it is not currently suitable for the few sites that have diskless worker nodes and problematic at some sites that have small HDs (<~80GB) on their WNs. This may be addressed in future developments but means that CVMFS is not currently viable as the sole method of distributing VO software.

It is worth noting that before CVMFS was available, many (large) sites saw severe performance and scalability issues for NFS based shared software areas. Particularly Atlas jobs place heavy demands on the software area throughout the job and with many thousands of jobs running software servers could easily become overloaded. This could adversely affect both job efficiency and site availablity.

Atlas uses in addition to CVMFS the possibility to install in the shared software areas on sites. An installer program (LJSFi) recognizes the shared area type and performs the necessary actions, i.e. resource discovery, job handling, installation validation and result collection.

LHCb has a slightly different mechanism to distribute software to shared software areas at sites with out CVMFS using 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.

Alice was using the sites shared software areas and is moving now to BitTorrent based software distribution. A pilot will fetch the lastest AliEn build and unpack it on the worker node, then execute as many payloads as possible within the pilot lifetime. At the same time the files are seeded for other jobs on the same site. The decentralized nature of this distribution is seen as an advantage as there is no single point of failure. The system also guarantees a consistent set of packages to be deployed and easy upgrades to later patches. Apart from some teething troubles this torrent based distribution works well and means sites do not have to worry about it.

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. --> to be moved to WG4/5?

Works well:

- cvmfs for software shared area

Works not so well

- cvmfs not universally adopted, so have to maintain two software delivery systems

Phase2: pkgsrc

Additional Info

Contributors

Jakob Blomer, Marco Clemencic, Andreas Pfeiffer, Marco Cattaneo, Costin Grigoras, Steve Traylen, Ian Collier, Alexandre Lossent, Alessandra Forti, Alessandro de Salvo

20111212 meeting

Improvement Areas (10=Blocker , 5= Medium, 1=Low)	
Impact	**Areas**
8	"Hammering" of the common software area by experiment tools
5	Lot of manpower invested into maintaining software deployment operationally and development
5	CVMFS is not installed on all sites, making it necessary to support two deployment systems
5	Package removal from sites shared software areas is not easy
5	Many duplications in the common part of the experiments software stack what concerns packages
5	Decoupling of experiment software from underlying operating system packages, not absolutely necessary
5	Integration of grid middleware clients into the experiments software stack needed for speedup of package deployment
3	Only very few areas where common tools are used between experiments for sw mgmt and deployment

-- StefanRoiser - 09-Nov-2011

This topic: LCG > WLCGTegOperationsWG3
Topic revision: r18 - 2011-12-16 - StefanRoiser