Fermilab

FERMILAB-Pub-03/032-E August 2003

Distributing Applications in Distributed Environment

Natalia Ratnikova[a*], Andrea Sciaba[b], Stephan Wynhoff[c†]

[a]Fermi National Accelerator Laboratory, P.O.Box 500, MS 234, Batavia, IL 60510, USA

[b]Istituto Nazionale di Fisica Nucleare, CNAF, Viale Berti Pichat 6/2, I-40127 Bologna, Italy

[c]Princeton University, Princeton, New Jersey 08544 USA

Software distribution is a process of delivering software products to the users. It is an essential part of the software process. The complexity of this task increases in the highly geographically dispersed collaborations, such as modern HEP experiments.

This paper focuses on general requirements to the software distribution system, main problems and various solutions. New requirements specific to the successful software operation in the GRID environment are discussed. We describe the current organization of the CMS software distribution and automated tools developed and used for the software distribution within the Collaboration.

For efficient and flexible software distribution it is important that software is relocatable (i.e. could be installed in arbitrary location), it has a minimal number of external dependences and does not require "root" privileges. Distribution software should provide friendly interfaces and a high level of automation. Though using different approaches, most distribution models include the same basic steps: packaging, transfer, unpacking, installation and setup.

CMS is using SCRAM (Software Configuration Release And Management [1]) tool to set up, build, and install software releases. The SCRAM built-in software distribution mechanism allows to download remotely the project sources and the list the of the required external products from the cvs repository. The procedure is very simple (single "click" on the Web page). SCRAM does not imply any restrictions on the software locations or any special privileges. However a high number of external products complicates the configuration issues. External packages need to be installed individually. Initial installation of the complete set of CMS software projects and its further support are non-trivial and require certain expertise.

Very large Regional Centers, such as CERN and FNAL, provide software installations on the distributed file system AFS, that enables users to access and use software via network, as if all files were stored on their local machines. AFS maintenance and support require considerable hardware resources and professional staff. It is usually part of the services, provided by the Computer Center to the experiments. This approach also relies on fast and stable network connection.

For the software installation on remote stations without high-performance network or staff experienced in the CMS software installations, we use the efficient method of distributing software on the hard disk. The user gets a replication of the master disk, containing well verified reference installation of all necessary products, including Linux operating system, general software, CMS software, sample data, documentation and instructions. Although such distribution may quickly become out-of-date, this truly plug-and-play distribution is extremely useful for newcomers, wishing to start using CMS software.

Another form of hard copy is distribution of RPM files on CD-ROM. These distributions use the benefits of RPM tool [2], such as the ability to perform automatically configuration tasks, the

*Tel: + 1-630-8406755, fax: + 1-630-8402783.
E-mail address: natasha@fnal.gov (N. Ratnikova)
†Present address: CERN-EP/CMS, 1211 Geneva 23, Switzerland. *E-mail address:* Stephan.Wynhoff@cern.ch

knowledge of the dependencies among different packages and the simplicity of use.

For the CMS software, RPM "spec" files are generated with a set of scripts having as argument the version number of the program to be packaged; the source files contain also the binaries compiled on a reference Linux machine in form of tarballs. Pre- and post-install scripts in the spec file are used to perform any checks and operations necessary to configure the software.

The RPM build is performed manually whenever a new version of a package is released and a tarball is prepared. The whole compilation and build procedure is planned to be automated in the near future, though. The RPM installation can be done both manually, with a single shell script, or with some cluster installation and managing system (e.g. LCFG in DataGrid [3]). The user can freely choose where to install the software; root access is not needed, as it is possible to use a private RPM database. The RPM distribution of the CMS software has been already used successfully in a number of sites. The CMS RPM files could be used also on machines where part of the software was installed in a different way (e.g. from tarballs), at the cost of disabling the dependency checking.

A common feature of the above approaches is that they distribute complete versions of products including the source code. Another original approach was developed for application-based distributions. Under application here we imply an executable file or set of files, accompanied by all required shared libraries and data files, that can be executed in a particular environment to accomplish a particular physics task.

To ensure that software is relocatable, it is required that locations of all application components are defined via corresponding environment variables, specified in the product configuration. SCRAM takes full care of the configuration in the CMS environment. The Distribution After Release tool, DAR, (concept first described in [4]) takes advantage of this. It assumes that the application is correctly installed on a particular master site. Then DAR analyzes all environment set with SCRAM and selects parts containing location specifications. If the variable specifying location points to some file, then this file is added to the distribution. If it is a directory, then all directory content is added. The rest of variables is treated as application flags. The resulting DAR-file includes all files referred by SCRAM, directly or indirectly, and the complete runtime environment. Once the DAR-file is transferred to the destination node, the application can be installed at any location that becomes a master location for the application. Every file is then unwind and put into directories below the master location in accordance with the original runtime environment, and a local runtime environment setup file is created.

This approach allows to replicate the complete application setup on remote site in highly automated fashion, assuming that application is installed on master site. The command *dar* -c ⟨*release_top*⟩ ⟨*export_dir*⟩ on the master site grabs all relevant files and runtime environment for the release installed in ⟨*release_top*⟩ and creates a DAR-file in the export directory. The command *dar* -i ⟨DAR-file⟩ ⟨*installation_dir*⟩ installs the application on a remote site. The same approach based on runtime environment can be used for distribution of non-SCRAM managed products.

DAR is most heavily used in cases when fast deployment of well defined applications is of primary importance, or where applications may or must be used "as is" without further modifications. DAR and DAR-like distributions are used in the official CMS mass production, on the USCMS GRID Testbed and for demonstrations on the Super Computing 2002 exhibition.

REFERENCES

1. C. Williams, SCRAM: Software Configuration, Release and Management, Dr. Dobb's Journal, Algorithms, April, 2001.
2. The RPM Package Manager (www.rpm.org).
3. The DataGrid Project (www.edg.org).
4. N. Ratnikova, G. Graham, C. Williams, H-P. Wellisch, CMS Software Distribution and Installation Systems: Concepts, Practical Solutions and Experience at Fermilab as a CMS Tier 1 Center, Proceedings of CHEP 2001