TOOLS FOR GRID DEPLOYMENT OF CDF OFFLINE AND SAM DATA HANDLING SYSTEMS FOR SUMMER 2004 COMPUTING.

A. Kreymer, A. Baranovski, G. Garzoglio,R. Herber, R. Illingworth, R. Kennedy, L. Loebel-Carpenter, A. Lyon, W. Merritt, S. Stonjek,I. Terekhov, J. Trumbo, S. Veseli, S. White,
Fermi National Accelerator Laboratory
V. Bartsch, M. Leslie, University of Oxford
S. Belforte, Istituto Nazionale di Fisica Nucleare
M. Burgon-Lyon, R. St. Denis, University of Glasgow
U. Kerzel, Universität Karlsruhe
F. Ratnikov, Rutgers University
A. Sill, Texas Tech University

Abstract

The Fermilab CDF Run-II experiment is now providing official support for remote computing, which has provided approximately 35% of the total CDF computing capacity during the summer of 2004.

We face the challenge of unreliable networks, time differences, and remote managers having little experience with this particular software.

The approach we have taken has been to separate the data handling components from the main CDF offline code releases by means of shared libraries, permitting live upgrades to otherwise frozen code. We now use a special 'development lite' release to ensure that all sites have the latest tools available. We have put substantial effort into revision control, so that essentially all active CDF sites are running exactly the same SAM code.
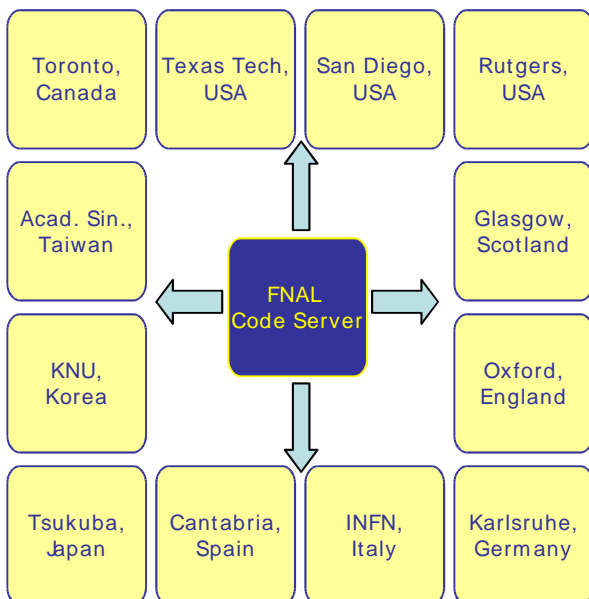
MONITORING TOOLS

We have created lightweight testing and monitoring tools to assure that these sites are in fact functional when installed. These are used in conjunction with externally developed tools such as Ganglia. SAM TV allows a user or SAM Shifter to see the details of any registered SAM Station, the projects presently running, the file transfers taking place, and the status of each process. DCAF monitoring allows all the batch queues for each DCAF site to be viewed, giving usage plots.
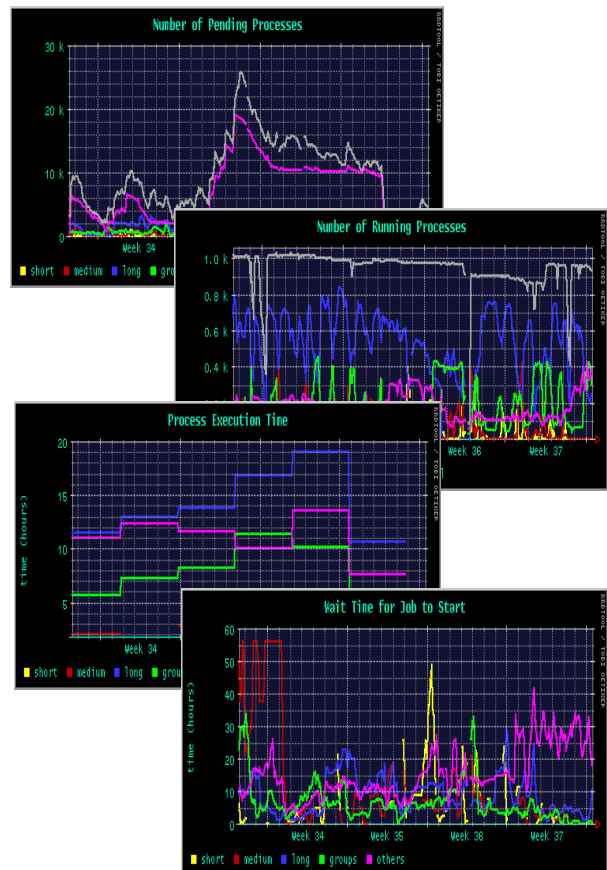
DCAF USING FBSNG AND CONDOR

The original CAF (Cental Analysis Farm) was controlled using FBSNG (Fermilab Batch System Next Generation) based software. A new version that uses Condor as the underlying batch system is now the standard for new DCAF (Distributed CAF) installations.

DEPLOYMENT OF SAM

We have made the first deployment of the SAM data handling system outside its original home in the D0 experiment. We have deployed to about 20 remote CDF sites. We are distributing and configuring both client code within CDF code releases, and the SAM servers to which the clients connect. Complete station installations, which once took days, are now performed in minutes. Simplification of this process was a crucial step for deployment.

SAM installation on a new cluster requires only a local SAM account. Root access is not required, removing headaches for both the remote system administrators and the CDF users.

REUSE POTENTIAL

The tools developed for CDF can be used to install SAM servers for D0 and other experiments. As new experiments, such as MINOS, expand their use of SAM the development and deployment of common tools become more important.

FURTHER INFORMATION

- Overview of which stations are running http://cdfdb-prd.fnal.gov/sam_local/SamAtAGlance/
- DCAF queues monitoring links and instructions for setting up a remote site:http://cdfkits.fnal.gov/DIST/doc/DCAF/
- CDF SAM Home Page: http://projects.fnal.gov/samgrid/cdf/cdf.html

COMMENTARY/FAQ

This paper, up to this point, is a transcription of the poster pretty much as presented at CHEP 2004.

But posters are very limited in what they can display. Their proper role is to attract attention and initiate discussion.

Therefore the rest of this document is cast as a set of questions and answers, as they might have occurred at CHEP. They are somewhat theoretical, as many of us were unable to attend due to travel restrictions. Because the answers faithfully reflect what I might have said, they are necessarily incomplete, or even incorrect. But you may find them useful.

Q - That's pretty impessive to have moved 35% of your computing offsite. It it true ?

A - Well, sort of. The capacity of the remote sites is indeed 35% of our total capacity. And we verify regularly that all sites are up and available. But the actual average usage is lower at the moment, probably under 25%, due to lack of user demand. Individual sites (particularly INFN and Karlsruhe) are heavily used, so we know things scale and that the full system will work well when pushed harder. Some sites have delivered 50 Tbytes in one month to end users doing normal physics analysis.

Q - So what sort of tools do you actually use to install the various DCAF components ?

A - The three main components mentioned here are the DCAF system, the CDF offline code, and the SAM data handling system. Each has unique requirements, and is handled somewhat differently.

The DCAF involves a batch system, and as such requires special privileges. Its installation is descibed at http://cdfkits.fnal.gov/grid/doc/dcaf/dcaf.html

DCAF installation uses tools such as rpm and/or yum to install security components, Fermilab's UPS/UPD to install the batch tools, and CVS to install various scripts and miscellaneous files. The security parts of this obviously require root access. In order to get this system deployed and stable while we are still working on getting future Grid tools in place, we have to take over one DCAF 'Head Node' at each site, which is placed inside the Fermilab kerberos security domain.

The CDF offline software is installed by a few scripts descibed in detail under http://cdfkits.fnal.gov . These make heavy use of the UPS/UPD system for revision control. CDF offline code is installed under the non-privileged 'cdfsoft' account at each site. This area is then NFS exported to the other nodes at that site. Surprisingly, this has scaled well for us. At Fermilab, a single server handles hundreds of desktop clients, and a second single server handles the entire CAF load of over 3000 client processes. In the event of hardware failure, either of these dual 1.3 GHz Linux servers can easily handle both loads.

CDF code installation is quite mature, having been installed on well over 100 systems. Normally the user just logs into a fresh cdfsoft account, runs the 'init_cdf' script and is done in a few minutes. We have to intervene manually perhaps 5% of the time, usually due to local conditions (odd shells, full disks, hardware failures, unique local security firewalls, etc.)

Within the past year or so we have come to rely on a special 'development lite' release of the CDF offline code. This is updated nightly from CVS at all sites. It provides the latest installation and management scripts, and provides the latest shared libraries for data handling.

Integrating and distributing the SAM software has been particuilary challenging. SAM was originally designed and used just by the D0 experiment. Fortunately the Fermilab Run-II experiments have worked to keep similar infrastructures and use similar tools. We can install SAM stations, which manage the remote projects and file caches, in a matter of minutes using a fairly simple init_sam script similar to (and derived from) the old init_cdf script used for CDF's offline code. Exactly the same script can be used to install stations for either CDF or D0. Of course, it was not at all simple to create the simple init_sam script. Fortunately, the end users do not know or care about this, now that things are working.

One of the larger challenges was to improve revision control. When only one experiment was involved, a single 'current' version of software usually sufficed. With two experiments using SAM, with different processing models and different schedules, it was necessary to define an overall formal SAM release based on specific versions of each component product.

Because most of SAM is implemented with Python scripts, we do not have many additional complication due to dependencies on operating systems and compilers.

Q - What is this UPS/UPD you keep mentioning. Never heard of it ! Why don't you just use Pacman, rpm's, yum, or some other more familiar package manager ?

A - UPS (Unix Product Support) and UPD (Unix Product Distribution) are widely used at Fermilab, and I'm not sure why they have not caught on elsewhere.   See http://www.fnal.gov/docs/products/ups/index.html
UPS allows you to install versions of products such that each process can select which version of each product to use at runtime, with a simple 'setup' command. Setup usually involves setting the execution and/or loader path, and setting a few environment variables. And possibly setting up particular versions of other required products. The actions are specified in a simple text file called a 'table' file. We test new versions of products without making any changes at all to our application programs or scripts, simply by setting up the newer versions. This is important when our applications are as large and complex as they are.

The UPD product is a set of perl scripts which package and distribute the product tarfiles, via web protocols. (You can ftp the tarfiles and extract the file manually if you like, There is nothing complicated about the packaging.) UPD allows product dependencies, and makes it easy to specify rules for placement of the extracted files on the target system, such that your UPS commands will find these products when you set them up.

UPS/UPD do not require root access, and in fact we strongly discourage installing products as root, for security reasons. Individual users can easiy and temporarily install, test or develop new product versions without affecting other users.

Q - Sounds complicated, how hard really is it to install this stuff ?

A – Well, installing a DCAF batch system is not exactly trivial, not too surprising. But at least 9 sites have succeeded, using the standard instructions.

The CDF code and SAM station and client code are a snap. People do just run the init_cdf or init_sam scripts and succeed most of the time. Installation usually takes just a few minutes.

Q - How do you handle the wide variety of operating systems in such a large scale deployment.

A - It's really not that bad. DCAF is installed primarily on the head nodes, which we insist on controlling completely at the root level.

CDF offline code is distributed only for Linux.

CDF, being a SAM newcomer, chose to stick with Linux for our SAM deployment. The D0 experiment does still have IRIX, SunOS and OSF1 SAM clients running at Fermilab. SAM code is mostly Python scripts, and is therefore quite portable.

The next major migration issue will be moving from our RedHat 7.3 based support to require Scientific Linux 3 (based on RedHat Enteprise Linux).

Q – Your monitoring tools seem a bit hodge-podge. Why not tie it all together with a framework like MonaLisa ?

A – Good idea. We're setting that up now.

Q – But how do you get all those stubborn people to cooperate, install the latest software, etc.

A – It is no accident. It takes a modest but serious level of effort to provide the necessary support so that people do not feel the need to roll their own solutions to various problems. And to integrate any private solutions for use by everyone. For us this is around an FTE of effort.
We have held three workshops at which most of the site managers have gotten together to work through major transitions (Initial installation, GridFTP deployment and standardization, and SAM v6 deployment, to vastly oversimply the thrust of the meetings so far.)
We have short (5 to 10 minute) daily meetings via H.323 videoconference to coordinate CDF's Data Handling and Grid activities.
The effort is assisted by few mailing lists, and by the CDF GRID web page presently at http://cdfkits.fnal.gov/DIST/doc/DCAF/