LCGCAF THE CDF PORTAL TO THE GLITE MIDDLEWARE

A.Fella*, INFN CNAF Italy, S.C.Hsu, Fermi National Accelerator Laboratory, S.Sarkar, D.Jeans INFN CNAF Italy, F.Delli Paoli, D.Lucchesi INFN Padova Italy, I.Sfiligoi INFN Frascati Italy, S.Belforte, INFN Trieste Italy, E.Lipeles, M.Neubauer, F.Wuerthwein, University of California, San Diego

Abstract

The increasing luminosity of the Tevatron collider will soon cause the computing requirements for data analysis and MC production to grow larger than the dedicated CPU resources that will be available. In order to meet future demands, CDF is investing in shared, GRID resources. Moreover a significant fraction of these resources is expected to be available to CDF before LHC era starts and CDF could benefit from using them. CDF is therefore reorganizing its computing model to be integrated with the new GRID model. LcgCAF builds upon the gLite Middleware in order to establish a standard CDF environment transparent to the end-users. LcgCAF is a suite of software entities that handle authentication/security, job submission and monitoring and data handling tasks. CDF authentication and security are entirely based on the Kerberos 5 system, so we needed to develop a kerberos certificate renewing service based on GSI certificates, to guarantee job output transfer to CDF disk servers. An enqueuing and status monitoring functionality was introduced to make the CAF submission latencies independent of the gLite Work Load Management System ones. The CDF batch monitoring is presently based on information from the GridIce monitoring and the LCG Logging and Bookkeeping systems. Interactive monitoring is based on the Clarens Web Services Framework. The beta version of LcgCAF is already deployed and is able to run CDF jobs on most INFN-GRID sites.

INTRODUCTION

The CDF experiment [8] in the Tevatron Run II has been collecting Physics data since 2000. Over the years, improvements in accelerator running have resulted in a higher instantaneous luminosity, while that of detector and trigger conditions have steadily increased the data taking efficiency. CDF has so far collected about 1 PB of data. Analysis of a rapidly growing volume of data and a constant need of larger samples of Monte Carlo (MC) events require consistent growth of computing resources over time. Thus far, in addition to the Central production and analysis farms at Fermilab (CAFs [1]), CDF has been successfully maintaining its own global computing environment with dedicated resources, in the form of 10 Decentralized Analysis Farms (dCAFs) and a few devoted MC production farms distributed over three continents. About 50% of the total CDF computing (2.5 MSI2K) is done at remote sites. However, expansion of the dedicated computing pools has virtually become impossible. Many sites have expressed intent to migrate towards shared resources. In fact, there are several sites in the LHC Computing GRID (LCG [2]) in Europe with shared resources that could be available to CDF immediately, if the computing framework were designed to take advantage of it.

LCGCAF ARCHITECTURE

The LCG-GRID standard based approach is to make use of the gLite [5] Workload Management System (WMS) that can act as a single point of submission to all the LCG. The concept behind the LcgCAF [6] is to re-implement the CAF infrastructure on top of the gLite middleware and build a GRID enabled dCAF preserving all the CDF specific features so that the users do not experience a steep migration curve.
The LcgCAF portal accepts user jobs and delegates to the gLite WMS. It also handles authentication/security, job monitoring and data handling tasks.

User Desktop

From the user point of view, submission to LcgCAF is identical to the standard dCAF submission. Once the user authenticates himself and selects LcgCAF as resource target, the Submission interface transfers the user application tarball, the job output location, the input dataset name and the name of the startup script to the LcgCAF portal. On successful submission the job ID and GRID Job ID (GID) are returned. The user receives a summary email with statistics and job status after the job is over. The interactive monitoring feature is available as command line interface (CLI). The CDF Monitor client interface permits to keep the available set of command, identical as the CDF standards and realizes the job run time monitoring. See section ”Monitor system” for detail.

The Submission Path

The CDF Submission Portal is a gLite User Interface (UI) on which the CDF Submitter, Mailer and Monitor daemons reside. The user submission request is received by the Submitter daemon which transforms the user kerberos credential into GSI Proxy [4] required for GSI authentication. Then, it proceeds with the creation of a Directed Acyclic Graph (DAG) used to represent a set of jobs that are related by a defined logical schema: the input, output or execution of one or more jobs are dependent on one or more other jobs.

* armando.fella@cnaf.infn.it

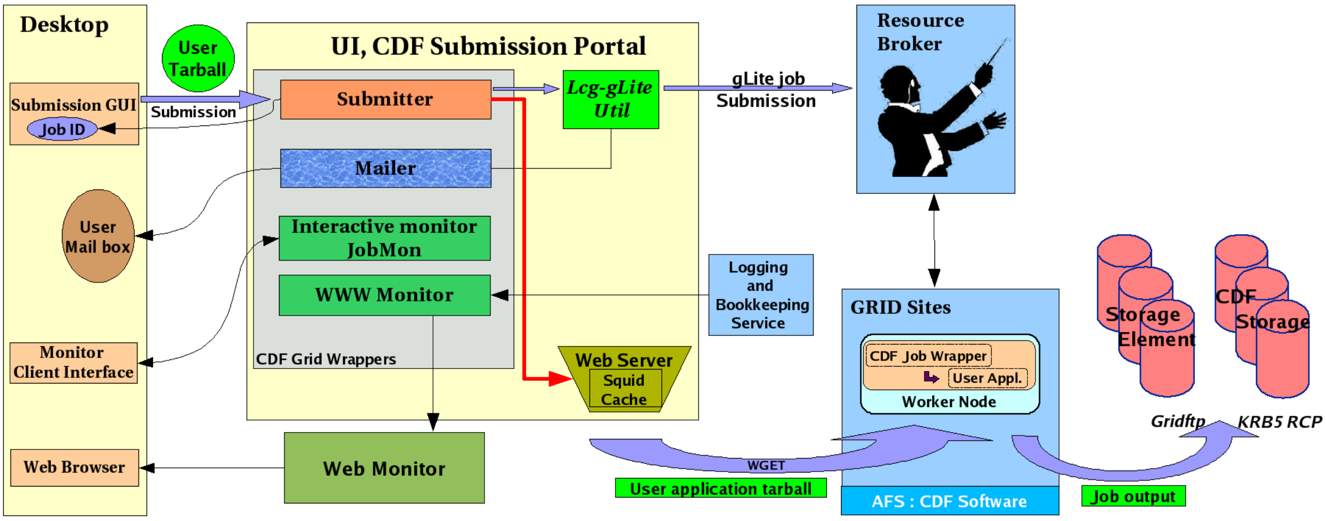

Figure 1: LcgCAF Architecture: submission and monitor processes.

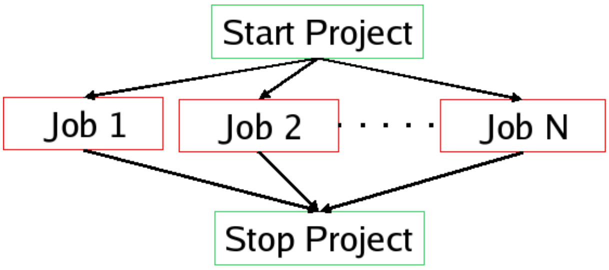

Figure 2: CDF Directed Acyclic Graph (DAG)

Referring to Fig. 2 the CDF DAG schema is formed by a single job node which initializes environment and data handling procedure; the layer below contains the proper user job nodes, each one depending on the first node. The third layer contains a single job node in dependence with all the above layer nodes, its task is to close the data handling procedure.

The gLite submission model includes specific methods to manage the transfer of job related input and output files called Input Sandbox and Output Sandbox. Since the CDF user application tarball tipically reaches size of hundreds MByte, the Input Sandbox is not the proper way to manage its displacement. On the other hand the CDF job Wrapper and the JobMon daemon are included in the Input Sandbox. The CDF job needs to access locally the user application tarball during execution, one of the Submitter daemon tasks is the caching of the user application tarball on the Web Server so that the job Wrapper, running on the WN, can pull it. To guarantee scalability of tarball retrieval a Squid cache/proxy system is integrated.

The Job Description Language (JDL) file in which GRID requirements and job fundamentals are defined, is compiled, and the proper gLite job submission command is launched. With this step the submission flux passes to the gLite WMS which submits the DAG to the site which matches with the highest rank. (see Fig. 1)

Job Execution

Once a job reached the Worker Node the CDF Job Wrapper is executed. It takes care of all the job life phases from the user tarball retrieval to the output transfer. In chronological order it generates the following action sequence:

- pull the user application tarball from the dedicated proxy server by WGET tool, untarring it and preparing the job work space.
- refresh the kerberos authentication which is needed to be able to transfer the output tarball to CDF specific storage. It was necessary to develop a mechanism called KDispenser that sends the renewed kerberos ticket on request from CDF job Wrapper
- fork the job process executing the start script provided by user. The job accesses the CDF Software from AFS.
- fork the execution of the JobMon daemon that provides interactive monitor service
- at job termination create summary logs in job workspace
- pack and transfer the job workspace to output location

The job output transfer at present time is targeted to CDF specific storage location and the possibility to store to a GRID Storage Element (SE) is in progress. In the two scenarios the transfer tools and the authentication methods are RCP via Kerberos authentication and Gridftp via GSI Proxy respectively. Once the job is over, the Mailer daemon retrieves the Job Output Sandbox using the gLite utility, collects information from CDF job wrapper logs, queries the GRID Logging and Book-keeping server (LB), and finally compiles and sends the job summary email to the user. (see Fig. 1)

MONITOR SYSTEM

The web based monitoring [7] for the LcgCAF relies on the LCG Logging and Book-keeping information and on CDF Job Wrapper logs collected in GRID Job Output Sandbox. That informations are translated in XML format and then transfered to the Web Server. Interactive monitoring, on the other hand, is based on JobMon distributed system. JobMon [3] is the monitor project officially supported by Open Science Grid (OSG [11]) , inserted in the Clarens server framework, it permits user to interact with system where job resides during running time. The JobMon interface reside on the CDF Submission Portal; when the CDF Monitor daemon submits a query to the JobMon, the JobMonD daemon on the Worker Node is contacted to make it execute that query. The set of available queries comprises: ps, head, cat, top, tail, kill, dir (ls), jobs (job status and information), log (checking progress of a job), node (show node name that job is running on).

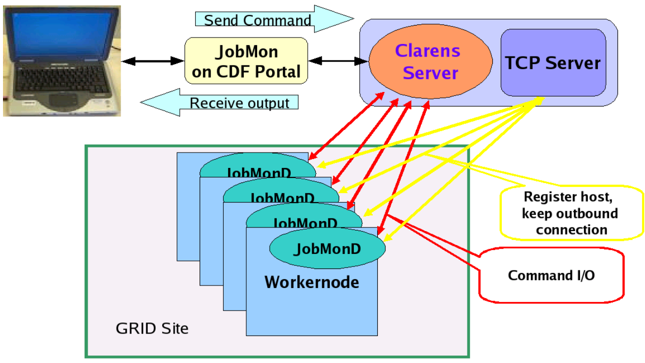

Figure 3: JobMon structure diagram.

USE OF LCGCAF FOR MC PRODUCTION

The Fig. 4 shows I/O data transfer in MC production job. The job running on a INFN-GRID site WN accesses CDF MC related software from AFS. The CDF MC package manages to retrieve from Data Base the necessary physics information. The central DB resides at Fermilab, Chicago and its remote access is managed by the FroNtier system [10]. Since the CDF experiment has a widely distributed environment for data processing and analysis, the access to their centralized database repository is critical. For delivering read-only data, such as calibrations, trigger information, and run conditions data, the client database interface has been abstracted to significantly reduce the load on the database and provide a scalable deployment model. Such a mechanism is implemented by a Squid Proxy/caching service present on each site. The produced MC events are temporary stored on disk and finally on Tape.

The LcgCAF, in beta test at present, has been successfully producing CDF MC events over INFN-GRID sites [9]. A total of two millions Herwig MC events have already been produced on CNAF Tier1 and Padova Tier2 sites. DAG jobs have been submitted on gLite WMS 1.4.1 with CDF specific patches and the output has been stored on CDF Storage Element.

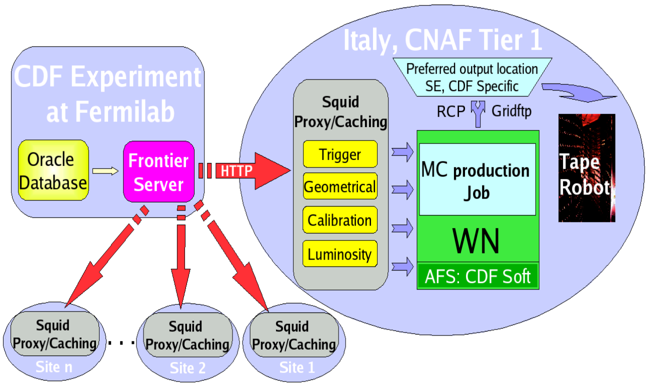

Figure 4: CDF Monte Carlo production diagram.

The DAG job submission is the best fit to the CDF job submission requirement. Such a feature is provided by the gLite WMS and represent the most important reason for LcgCaf to choose this way to access GRID resources. On the other hand the gLite WMS component is still not in production and affected heavily the LcgCAF submission success rate. During the first test phase it has been possible to tune WMS parameter respect with the CDF needs and helps to find out WMS bugs. The efficiency in this scenario was $\sim 45\%$.

Most of the problems affecting the jobs were fixed in the next WMS releases which permit, in a second test submission phase to report an efficency of $\sim 94\%$. The $\sim 6\%$ of failures were basically due to problems in accessing specific site resources. With the incoming production gLite WMS release, we are confident that it is possible to improve the LcgCaf efficency also by investigating and solving site problem.

CONCLUSION

LcgCAF is a software architecture that permits CDF to use LCG-GRID resources, is the major step in direction of complete integration in a LCG-GRID standard model. The MC Production using LcgCAF architecture is well underway on INFN-GRID Italy and it's in progress the accessing to European Sites

REFERENCES

[1] I. Sfiligoi et.al., The Condor based CDF CAF, Proceedings of the CHEP2004 Conference

[2] The LCG Home Page, http://lcg.web.cern.ch/LCG/

[3] The JobMon Project Home Page, http://jobmon.sourceforge.net/

[4] The kx509 Project Home Page, http://www.kx509.org/

[5] The gLite Home Page, http://glite.web.cern.ch/glite

[6] LcgCaf Official Site: http://www.pi.infn.it/cdf-italia/public/offline/lcgcaf.html

[7] LcgCaf Web Monitor: http://wn-04-04-26-a.cr.cnaf.infn.it:8081/lcgcaf/

[8] CDF Official Site: http://www-cdf.fnal.gov/

[9] INFN GRID Official Site: http://grid-it.cnaf.infn.it/

[10] S. Kosyakov et.al., Frontier: high performance database access using standard web components in a scalable multi-tier architecture, Proceedings of the CHEP2004 Conference

[11] The OSG Home Page, http://www.opensciencegrid.org