Editorial Manager(tm) for Journal of Physics: Conference Series
Manuscript Draft

Manuscript Number: CHEP245R1

Title: CDF way to Grid

Article Type: Contributed

Corresponding Author: Dr DONATELLA LUCCHESI, PhD

Corresponding Author's Institution: University of Padova

First Author: DONATELLA LUCCHESI, PhD

Order of Authors: DONATELLA LUCCHESI, PhD

CDF way to Grid

Donatella Lucchesi
(CDF Collaboration)
Department of Physics, University of Padova, via Marzolo 8,35131 Italy
E-mail: donatella.lucchesi@pd.infn.it

Abstract. The CDFII (Collider Detector at Fermilab II) experiment is taking data since 2001 using a multipurpose detector. Currently about 5 fb^{-1} of data has been written to tape and successfully analyzed by physicists using CDF computing infrastructure. The computing architecture has evolved from the initial dedicated farms to using decentralized Grid-based resources on the EGEE Grid, Open Science Grid (OSG) and Fermilab Campus Grid. In order to deliver high quality physics results in a timely manner to a running experiment, CDF had to adapt to Grid with minimum impact on the physicists analyzing the data. The use of portals to access the computing resources has allowed CDF to migrate to Grid computing without changing how the users work. The infrastructure modifications were done by small steps over several years to reach the current stable configuration. The evolution of the architecture and the performances reached by using portals are presented here.

1. Introduction

The CDFII experiment collects events from proton-antiproton collisions at the Tevatron collider at Fermi National Laboratory since 2001. Up to now the collaboration has on tape around 5 fb^{-1} corresponding to 1.5 PB of data. Simulated events using Monte Carlo (MC) technique are also necessary to calculate detector acceptance and analysis procedure efficiency. As of today CDF has produced about 0.7 PB of MC data which is almost half of the data collected. CDF needs can be separated in raw data reconstruction, n-tuples production, events simulation and user analysis. At the begin of the data taking these tasks were done in parallel on dedicated farms. The increasing of the cpu demand and the advent of the Grid era forced CDF computing to evolve toward a distributed model and to adapt to the Grid. The core of the computing model is the CAF (CDF Analysis Farm)[1] and during the time it evolved from being an analysis farm to become a portal to access Grid resources. In CDF users develop and debug their code on their desktop and when they are ready for the analysis submit the same program to the CAF. Each job is parallelized by splitting it in sections usually the order of hundreds. The job is first submitted to the CAF head node and then to the farm using Kerberos [2] for authentication and integrity check. If the submission succeeds the users receive back a job JID that can be used to monitor the job status at anytime from anywhere where CAF client is available. The job output is copied into the location specified by the users. The CAF features were retained when it moved from farm to portal and the users did not have to learn other methods to run their jobs. In order to access OSG and EGEE resources the CAF code was modified to be interfaced with both middlewares.

2. CDF computing model

CDF experiment writes data to tape via an intermediate cache disk at about 60 MB/sec. These events are later reconstructed to high level objects like electrons, muons and jets using a dedicated program. In the past reconstruction jobs were ran on a dedicated farm, now they are submitted to the generic CDF resources. The reconstructed events are then written to tape and users can access them for analysis. Data and simulated data cataloguing is done with SAM [3] which provides distributed data access as well as datasets and files history. As shown in figure 1 users can submit jobs to any site from any place provided they have access to CDF software and Kerberos client to authenticate them self . Jobs that require data are submitted to the Fermilab farm where this is available, only exception is CNAF that hosts specific datasets. These sites have also a "SAM station", a computer system that steers the data requests to the catalog, necessary to get files from the tape robot or the disk storage and to store files to tape. Monte Carlo data is instead generated everywhere with a preference in the sites where data access is not granted. MC datasets final destination is anyway Fermilab where they are stored to tape.

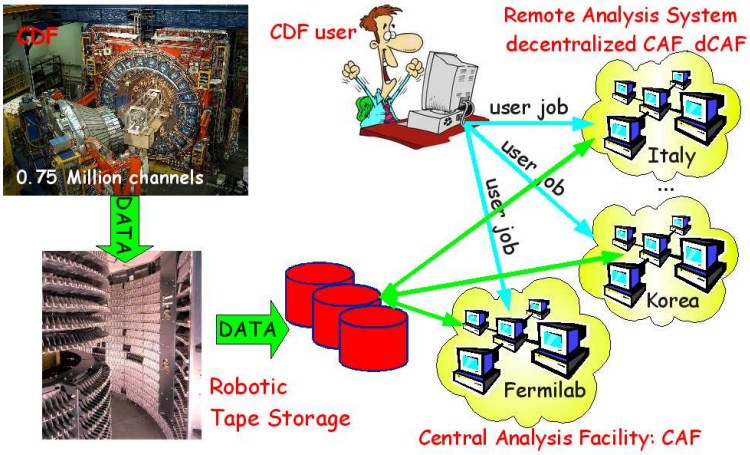

Figure 1. CDF Computing Model

3. The CAF

The core of the CDF computing model is the CAF. Only CAF client tools are used in CDF to submit and monitor any kind of job. The users can submit from anywhere and do not need to stay connected. Jobs can be monitored from everywhere at any time provided the users have Kerberos credential and access to CDF software. The job end together with a job summary are notified to the users via email. The CAF was developed at the begin of CDFII as an head node for a farm and it was intended to be an intermediate layer between users and batch system. Given its flexibility it was easy to modify it to be adapted later to the Grid and become a portal.

3.1. The CAF implementation

The CAF consists in a set of daemons that accept requests from the users via kerberized socket connections and coverts them into commands to the underlying batch system. The first batch system used was FBSNG [4] but since 2001 Condor [5] replaced it. As shown in figure 2 the daemons can be grouped in three main classes:

Submitter daemon

It accepts the user requests together with the user tarball. It creates the batch-system-specific

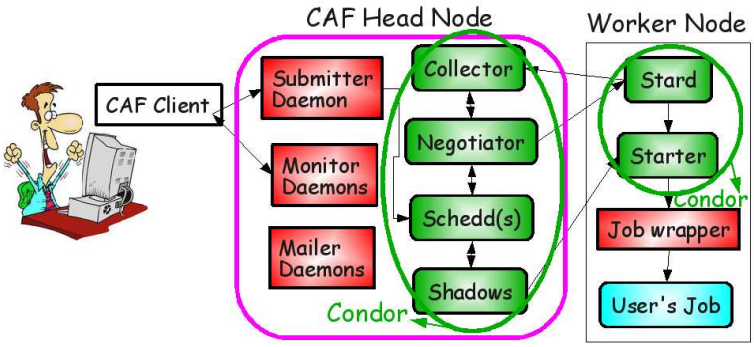

Figure 2. CAF layout with the most important daemons.

submission files for each job sections. It actually submits jobs to the batch queue and when this is done it returns a job ID to the user.

Monitor daemons

They follow the job sections during their life and allow the users to interact with the job. Requests by the users are accepted and converted into batch-system-specific queries that are translated using a CDF-specific jargon back to the users.

Mailer daemon

It monitors the status of the job and when all the sections end it sends a notification to the users including a summary with the exit status code of each section.

The **job wrapper** is an other part of the CAF code. It sets up the security envelope and prepares the environment for the actual user executable. When the user job ends it copies anything it finds on the worker node directory where the job is executed to the final destination. The job wrapper also monitors the job during its execution logging information not captured by the batch system. This allows the users to interact with a specific section in a interactive way on the dedicated farm and in a almost interactive way on the Grid.

3.2. The CAF monitor

The monitor is a very important part of the CAF architecture. Users access the monitor information to verify the jobs status, to kill them if something is wrong and to submit new sections when the previous ones are almost at the end. The monitor daemons were designed when the CAF was a local farm and the process to retrieve information have been updated to cope with Grid environment. OSG Condor commands are used in the OSG Grid while in EGEE custom software has been developed. In both cases the previous functionalities were preserved.

The web based monitor is implemented in the same way in OSG and EGEE. The monitoring process sends periodically information about the job status directly to the head node. A daemon running on the head node called xml monitor is in charged of exporting these information in XML format and a web server publishes them in a user-friendly way, as it can be seen from a screen shot in figure 3. Job status for all users as like as specific information about a single submitted job can be easily retrieved using this interface, giving a good and quick overview of the status of the CAF and the submitted jobs. An history of managed jobs is also available for analysis purposes.

The interactive monitor for the EGEE portal accesses the same information via a command-line tool which queries directly another daemon running on the head node, namely monitor, and implements unix-like commands like `ls, ps, tail, ..` in order to give access to the analogous information about the jobs with the exception of `kill` which is mapped to the corresponding job

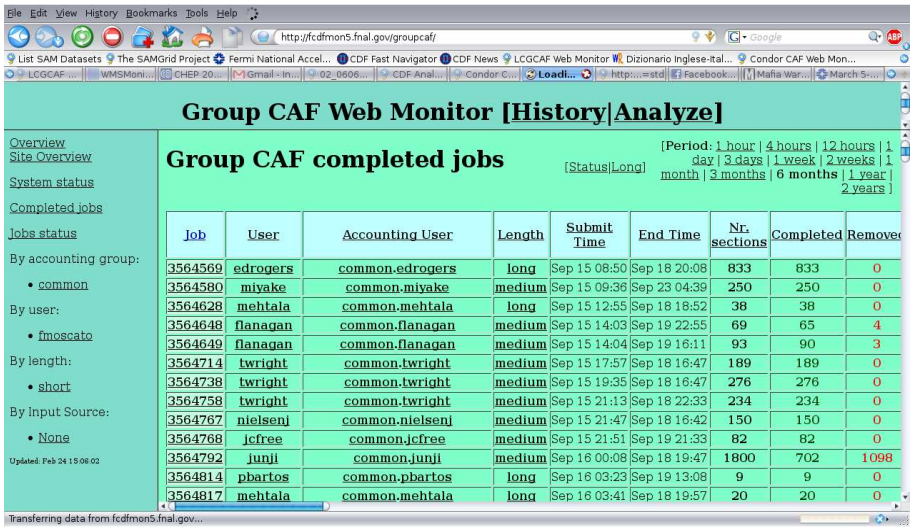

Figure 3. Screen shot of the CAF Web monitor.

killing commands. On OSG portal the intended commands are obtained via Condor Computing on Demand (COD)(with the exception of GlideinWMS [9] that has its own method). The output of the interactive monitor commands is always the same regardless the used Grid. An example is shown in figure 4 that displays the output of `CafMon dir` command.
These queries are, unlike the web interface, authenticated by a Kerberos certificate, so that only the owner of the job can ask for specific job information and take actions on the job itself. The interactive monitor was unique to CDF for long time and it is one of the most successful part of the CAF.

```
/cdf/home/lucchesi>CafMon dir 4672 4

Analysis Farm: lcgcaf    Host: pcdf11.pd.infn.it

WARNING: Python C API version mismatch for module krb5:
  This Python has API version 1010, module krb5 has version 1007.
total 12
-rw-------  1 cdf009 cdf   0 Nov 28 23:22 job_4.err
-rw-------  1 cdf009 cdf 144 Nov 28 23:22 job_4.out
-rw-r--r--  1 cdf009 cdf   5 Oct  5  2006 quit.tcl
-rwxr-xr-x  1 cdf009 cdf 138 Sep  6 14:38 run.sh
```

Figure 4. Screen shot of the output command `dir` of the CAF interactive monitor.

4. CDF transition to Grid

The CAF as local farm was hosted at Fermilab. As the resources demand increased due to the higher and higher delivered luminosity of the Tevatron, the model evolved into the dCAFs (distributed CAFs) located at several CDF institutions around the World. Instead of keeping this model and growing the number of dedicated farms to cope with the increased cpu request, CDF decided to adapt the computing architecture to the Grid structure. This had the advantage of exploiting the LHC resources almost for free before LHC starts to take data. In addition to that CDF moved from a self-supported farms model to the Grid-supported model where CDF has to maintain only its specific code and the cpus are part of the LHC Tier centers.

The transition to Grid was performed following these simple requirements:

- Adopt the CAF model and in particular keep the CAF client code to facilitate the users.
- Keep usage of Kerberos for authentication; in this way users do not need to apply for Grid certificate, everything is done by the CAF middleware.
- Data analysis happens at Fermilab and for small datasets at CNAF, Monte Carlo data production occurs in any sites. This assumption avoids CDF to move data around and to find an adequate method to do it using the SAM catalogue.

In order to implement this program the CAF head node of the local farm has become GridCAF, a "portal" to interface OSG and EGEE gLite [6] middleware.

5. The Condor based GridCAF: GlideCAF

The first GridCAF is an evolution of the local farm with Condor as batch system and it is based on the Condor glide-in concept: a tool for submitting and executing the Condor daemons on a Globus resource and to make it as part of Condor pool. The dedicated farm has Condor running on each node forming a dedicated pool. Grid resources can become part of the pool if allowed and the Condor daemons can be started on them. The GlideCAF [7] starting from a CDF dedicated farm adds to it additional resources in an opportunistic way and creates a private Condor pool out of Grid resources. In order to do that a new daemon, the glidekeeper, is added to the CAF code (see figure 5). The glidekeeper submits glide-in job to all the sites that are in

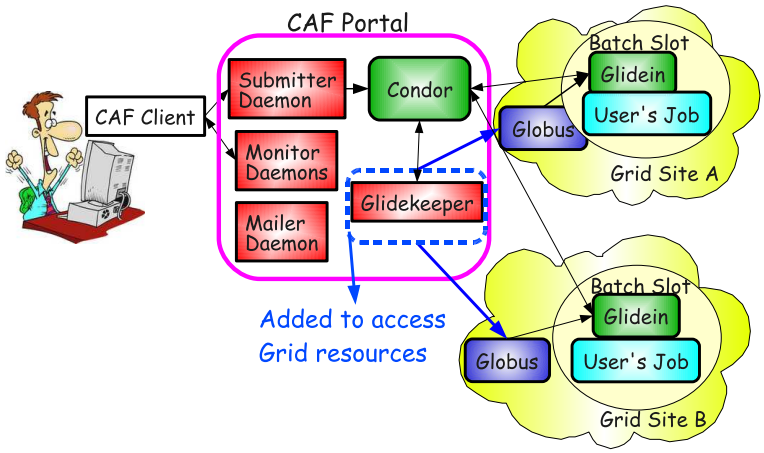

Figure 5. The Condor based GlideCAF layout.

the CDF wish list. When this arrives to a worker node Condor processes start, join the CDF pool and start asking for users jobs. The glide-in job exits when the first job finishes but the resource can stay in the pool longer. The glide-in method is very powerful: it guarantees the job execution in a working site in a short time. But it introduces a small inefficiency in the system due to glide-in jobs them self and to the fact that several glide-in jobs run on different sites but only one gets the user jobs. Up to now the resources have been higher than the demand making this inefficiency negligible.

5.1. GlideCAF performances

The first GlideCAF was deployed in Italy at CNAF Tier1 and later at Fermilab. It served successfully CDF for many years. Figure 6 shows the number of running jobs at CNAF as function of the time during the last three years. It has been possible to run up to 2000

jobs concurrently with no issues. In figure 7 the analogous plot for the Fermilab GlideCAF is displayed. Here the number of running jobs at the same time is higher respect to CNAF because data is available only locally.

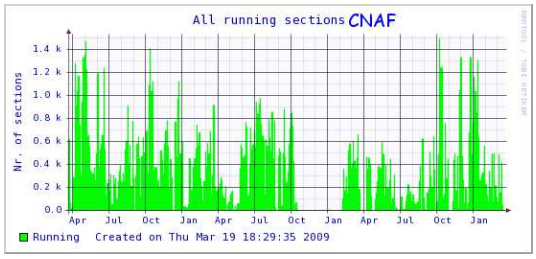

Figure 6. Running jobs as function of time during the last three year at CNAF-Italy.

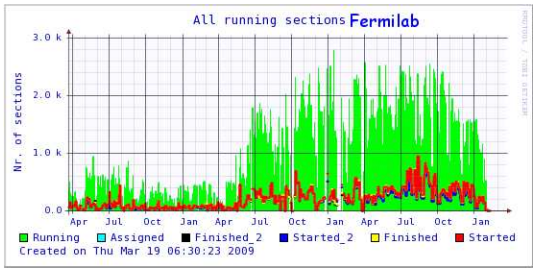

Figure 7. Running jobs as function of time during the last three years at Fermilab.

The GlideCAF at Fermilab has shown scalability issues as the number of the managed resources increased. Some of them were mitigated with the usage of more powerful hardware and with ad hoc Condor configuration. With this set up CDF is able to run up to 5,100 jobs concurrently as shown in figure 8 but no more than that. Figure 9 shows the actual CDF request in number of waiting jobs. The system need to improve to go up by about an order of magnitude.

Since this is not possible with the current GlideCAF, CDF is moving toward the glideinWMS [8]. A CAF portal with glideinWMS is production at CDF to access OSG sites. A detailed description of the glideinWMS implementation at CDF and its performances tests have been discussed at this conference [9].

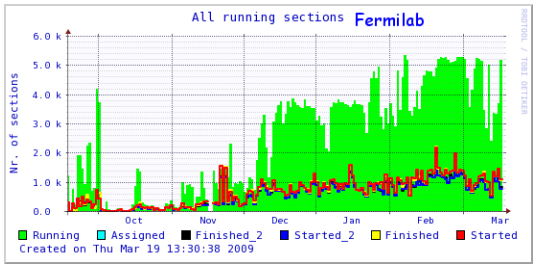

Figure 8. Fermilab GlideCAF running jobs as function of time during the last six months using the most performing configuration.

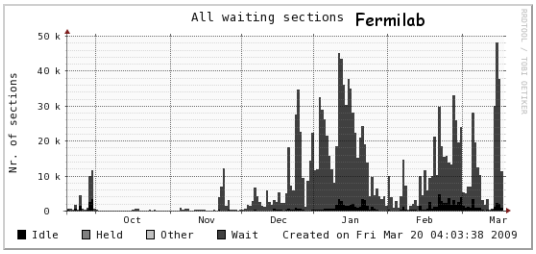

Figure 9. Fermilab GlideCAF waiting sections as function of time during the last six months. Peaks of 40,000 waiting sections were reached.

The CDF usage of Grid resources in OSG can be seen in figure 10. The relative cpu time used by the selected VOs from April 2008 to March 2009 is displayed in the pie chart. CDF is the second best user and this is manly due to the fact that CDF has a large amount of resources in the Fermilab Tier1. Only with the introduction of the glideinWMS CDF is starting to exploit OSG resources outside Fermilab.

6. The gLite based GridCAF: LcgCAF

The second GridCAF is a rewrite of the CAF code into the LcgCAF [10], a portal based on Workload Management System (WMS), the gLite facility to submit jobs to the EGEE resources matching the users requirements. The general LcgCAF architecture, shown in figure 11 is based

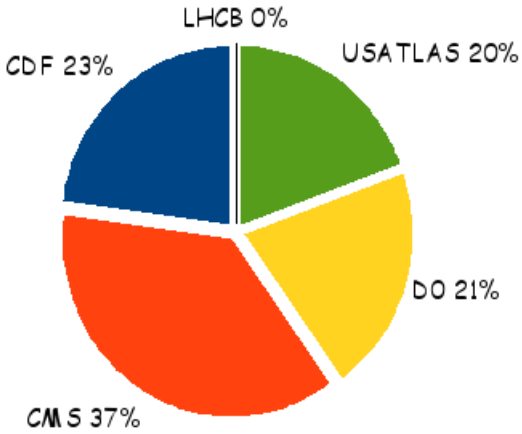

Figure 10. Usage of OSG resources (cpu time) by the selected VO's.

on a submission point which is also a Grid User Interface (UI) where the major part of the services responsible for accepting, submitting and monitoring the users job are running. After the users are authenticated, the CAF clients connect to the submitter daemon which then delegates the job to the WMS hosted at CNAF Tier1. A this point the Resource Broker, one of the elements of the WMS, dispatches jobs to the Grid Computing Element (CE), the interface to the local resource manger. Once the job arrives to worker node the situation is the same as

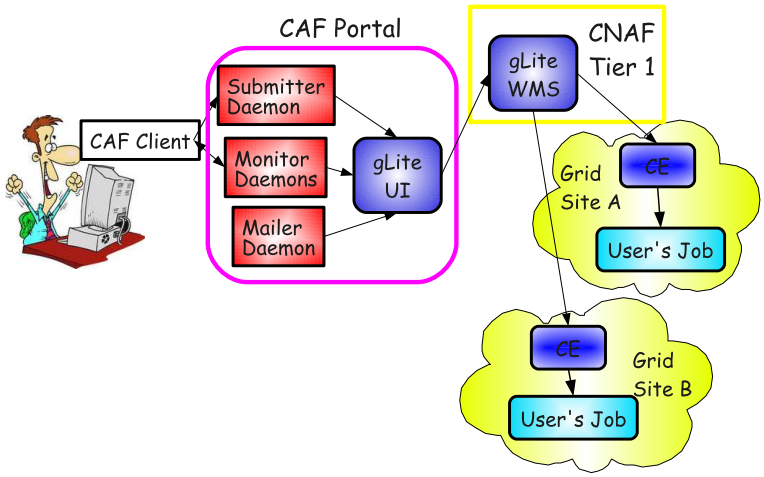

Figure 11. The WMS based LcgCAF layout.

in any other CAF: the LcgCAF wrapper is executed. It adapts the environment to the CDF needs, gets the support software and the real user job. Then it executes the user scripts and forks some monitoring processes, and finally it retrieves the output when the job is completed.

6.1. LcgCAF usage

LcgCAF can accept Monte Carlo production and other simulation jobs. Actually LcgCAF accesses 11 different European Grid sites, seven of them are Italian; the preference for Italian sites is due to the presence of numerous and active CDF community in several Italian institutions.

The major part of the jobs run at CNAF Tier1 where CDF has dedicated resources in terms of cpu and disk space. This explains the relative high usage of resources in Italy as shown in figure 12 where the CDF VO is at the level of the major HEP VOs. The European resources are poorly exploited with only few percent of the pie chart in figure 13. One of the major problems that retain users from submitting large amount of jobs using LcgCAF is the high rate of failures due to the transfer of the job output back to Fermilab. This is improving and it is briefly described in the next section.

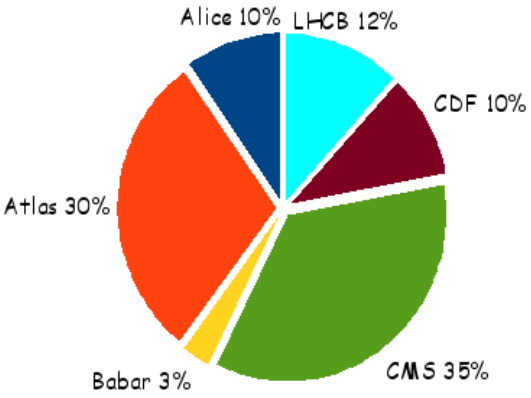

Figure 12. Usage of the Italian resources (normalized cpu time) by the selected VO's.

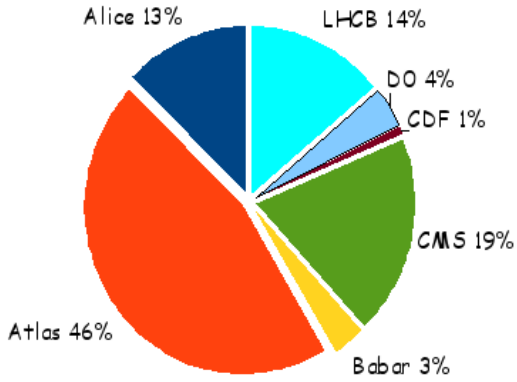

Figure 13. Usage of the European resources (normalized cpu time) by the selected VO's.

7. Code distribution and job retrieval

With the Grid transition CDF had to solve two additional problems: the CDF code distribution and the data movement.

7.1. CDF code distribution

The CDF code distribution was approached with a different philosophy in OSG and EGEE. There is no CDF code in OSG sites except Fermilab where the software is distributed via NFS. If the job has to run outside Fermilab it must be submitted with everything it needs. CDF has created standard Monte Carlo tarball that are self-contained to allow Monte Carlo data production in OSG sites outside Fermilab.

Differently, in EGEE the CDF code is distributed using Parrot [11] which is able to virtually mount a remote file system using different protocols. This software, able to run at user level, traps the user program's system calls in order to establish if the requested file is local or not and, if it is not, it is transferred and stored in a local cache in a completely transparent way for the user program. Details on how Parrot is implemented in LcgCAF and its performances were presented at this conference [12].

7.2. Job output retrieval

The *rcp - remote copy protocol* is the default method used in the early CAF to copy the job output to the final destination. When the resources are displaced in several sites around the World this method shows its limits and causes inefficiencies. There is a waste of cpu because of output loss and worker nodes idle waiting for output transfer. To overcome this problems a new mechanism has been designed and tested. The new method, described in detail here [13], adds an intermediate layer constituted by a local Storage Element (SE), which is physically near the

site where the job runs in order to reduce the transfer time. The output is then transferred to FNAL optimizing the available resources in an asynchronous way.

8. Conclusions

The CDF experiment keeps taking data with very high efficiency and the users need resources to analyze it and to generate a large amount of simulated data. The previous computing model satisfied CDF needs but it is not suitable in the actual Grid era where all the resources are in Tier centers. Moreover CDF does not own enough resources to satisfy the peak needs and must use opportunistically other cpus. To accomplish that the CAF moved from being an head node to a portal with two different flavor: one based on glidein mechanism for OSG Grid and one that exploits the WMS for EGEE Grid. This was done in a completely transparent way for the physicists. Improvements on the submission side are being deployed in OSG with the adoption of glideinWMS while data transfer is at the prototype level almost everywhere. With this configuration CDF did access OSG and EGEE resources successfully up to now and it will be able to analyze data and produce physics results concurrently to the LHC VOs born on the Grid.

Acknowledgments

CDF owes a lot of gratitude to many people: the former CAF-team for the CAF design, the Condor team for the help in setting up the CAFs, the Fermilab center and the CNAF center for their continuous support. Up to now CDF has been able to keep up with the Grid evolution thanks to Igor Sfiligoi, Doug Benjamin, Marian Zvada, Simone Pagan Griso and Gabriele Compostella. To them my thanks for having been the best colleagues I could hope.

References

[1] M. Casarsa, S. C. Hsu, E. Lipeles, M. Neubauer, S. Sarkar, I. Sfiligoi, F. Wuerthwein, "The Cdf Analysis Farm" 2005 *AIP Conf. Proc.* **794** 275.

[2] "Kerberos Web site" *http://web.mit.edu/Kerberos/.*

[3] I. Terekhov *et al.*, "Distributed data access and resource management in the D0 SAM system," *FERMILAB-CONF-01-101*
SAM web site *http://d0db.fnal.gov/sam/*

[4] "FBSNG Web site" *Next Generation of FBS http://www-isd.fnal.gov/fbsng/.*

[5] D. Thain, T. Tannenbaum, M. Livny, "Distributed computing in practice: the Condor experience.", *Concurrency - Practice and Experience* **17, 2-4**, 323 (2005).

[6] E. Laure *et. al.*, "Programming the Grid with gLite", *EGEE-TR-2006-001* (2006).

[7] S. Sarkar, I. Sfiligoi *et al.*, "GlideCAF - A late binding approach to the Grid" *presented at Computing in High Energy and Nuclear Physics, Mumbay, India, Feb. 13-17, 2006* **147** 2006.

[8] I. Sfiligoi " glideinWMS - A generic pilot-based Workload Management System" *presented at Computing in High Energy and Nuclear Physics, 2-7 September 2007, Victoria, British Columbia, Canada* published on *Journal of Physics: Conference Series 119 (2008) 062044 http://www.iop.org/EJ/volume/1742-6596/119*
GlideinWMS Web site *http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/doc.html.*

[9] M. Zvada "CDF GlideinWMS usage in Grid computing of High Energy Physics" *presented at Computing in High Energy and Nuclear Physics, Prague, Czech Republic, 21-27 March 2009* **407** 2009.

[10] Compostella, G. Delli Paoli, F. Jeans, D. Lucchesi, D. Sarkar, S. Sfiligoi, I. "LcgCAF: a CDF Submission Portal to Access Grid Resources" *presented at 2006 Nuclear Science Symposium, Medical Imaging Conference published Nuclear Science Symposium Conference Record, 2006. IEEE Oct. 29 2006-Nov. 1 2006 Volume: 2, On page(s): 873-878*

[11] Moretti C, Sligoi I, Thain D, Transparently Distributing CDF Software with Parrot, Feb 13-17 2006. Presented at CHEP06, Mumbay, India, **26**.

[12] G. Compostella "CDF software distribution on Grid using Parrot" *presented at Computing in High Energy and Nuclear Physics, Prague, Czech Republic, 21-27 March 2009* **402** 2009.

[13] M. K. Jha "A new CDF model for data movement based on SRM" *presented at Computing in High Energy and Nuclear Physics, Prague, Czech Republic, 21-27 March 2009* **413** 2009.