CDF/DOC/COMP_UPG/PUBLIC/7817

CDF Distributed Computing Plan 2005

Ian Fisk, Igor Sfiligoi

September 7th, 2005

version 1.0

This document describes the current planning for CDF distributed computing and proposed interactions with the grid projects and opportunistic utilization of large-scale computing resources. It attempts to formulate a prioritized program of work that meets the needs of 6 identified CDF computing areas: user analysis, user Monte Carlo (MC) generation, organized group analysis, centralized experiment MC generation, real time event reconstruction, and organized data reprocessing. There are four large computing resources that were constructed by CDF that have been used to meet the experiment needs in Run2: the CDF Analysis Farm (CAF), the real time production farm, the collection of distributed analysis farms (dCAFs), and the offsite simulation production farms. CDF has recognized needs for additional computing resources and will attempt to satisfy some of these by capitalizing on deployed grid facilities, the two more prominent being the LHC Computing Grid (LCG) in Europe and the Open Science Grid (OSG) in the US. Additionally, there is a Fermilab computing initiative, FermiGrid, designed to enable more effective sharing of computing facilities between supported experiments that CDF intends to incorporate into the computing strategy of the experiment.

1 Introduction

The CDF experiment comprises 500 physicists from 60 institutions. The experiment now has nearly 1PB of data under management. After an initial large SMP system two Linux clusters were established to meet the computing needs of the experiment. The CDF Analysis Farm (CAF)[1] is a large batch farm with some limited interactive capabilities. Additionally, a production batch farm[2] was established for primary event reconstruction and organized reprocessing. Due to the speed of the reconstruction application and the achieved luminosity of the accelerator, the requirements for the production farm have not driven the overall size of the experiment computing. The capabilities of the analysis farm have grown exponentially since the beginning of Run2. The CDF computing model calls for a general flattening to grow with integrated luminosity.

The computing for simulation production and user analysis represent the largest fraction of the total computing resources. Driven in part by physical infrastructure limitations (power and cooling) at FNAL and in part by the ability to leverage external resources, CDF has moved a significant part of its computing resources offsite. Most of the remote resources are organized as distributed CDF analysis farms (dCAFs). There are now ten sites over three continents, located at universities and national labs. Together the CAFs and dCAFs represent 5 THz CPU (2.3 MSI2K) of computing capability and handle more than 10,000 processing request from approximately 100 users per day. Experiment data is delivered to the processing farms at between 200MB/s and 400MB/s. Additionally, some sites have elected to create production style farms dedicated to the production of simulated data.

CDF established a goal of 50% off-site computing resources by the end of 2005. This goal has largely been met, but there are two potential issues. The first is that not all centers are created equal. It continues to be more efficient to perform analysis tasks on the FNAL CAF. The central CAF is the only site with access to all the data. The adoption of the SAM data management system should improve the ability to utilize off-site resources. The CDF model utilizes largely static partitioning of datasets to remote clusters and concentrated analysis tasks where the data is located. Additionally, the central CAF enjoys support from a team of experiment-dedicated operations staff. The current methods to enforce experiment policy that encourages users to utilize centers in the most efficient way possible should be improved.

The second identified issue in growing the CDF computing resources is the need to make use of more common distributed computing infrastructure. The CDF computing system has been successful in delivering resources to users, but is based on the use of dedicated resources. There are several sites with shared resources that could be available to CDF if the computing system were designed to take advantage of it. There are several sites that have indicated no additional resources will be added to the CDF specific farms, but additions will be added to the common farms.

CDF has a well established user community and the migration and upgrade plans need to ensure that no functionality that is important to the users is lost and that the interface migration is smooth. Most of the plans described in the remainder of this document attempt to maintain the existing submission infrastructure and maintain the interactive functionality available in the CAF. This results in some limitations for the adoption of distributed computing systems and requires some development and creativity in the adoption of others, but is seen as necessary when dealing with an established user base.

2 Technological solutions

First we describe the technological solutions we have available or are envisioned to be available in the near future. To make things easier, we have introduced a matrix. The matrix is shown in Table 1. One axis shows the proximity to Fermilab, while the other shows the kind of resources that are used. This gets us four categories, ranging from the very desirable onsite dedicated resources, to the flexible but difficult offsite shared resources. For each categories we have several possible solutions and we are going to describe them all below.

	Dedicated	Shared
Onsite	Production farm	CAF on FermiGrid via glide-ins [FermiGlideCAF]
	CAF	OSG submission to FermiGrid
		Condor-G submission to FermiGrid
Offsite	CAF offsite [dCAF]	CAF over Condor-G (any Grid) via glide-ins [GlideCAF]
	CAF over Condor-C (over dCAFs) [ccCAF]	CAF over LCG Resource Broker [LcgCAF]
		CAF over Condor-C (any Grid) [ccCAF]
		SamGrid direct submission
		Condor-G direct submission (to any Grid)
		OSG direct submission
		LCG direct submission

Table 1: Matrix of location vs. ownership

2.1 Production farm

The production farm[2] is the most reliable of all the CDF computing facilities. It runs on dedicated hardware and has excellent networking. It accepts only very well understood applications to prevent different instances running on the same node to interfere with each other.

From the implementation point of view, it is essentially a CAF. See next section for more details.

2.2 The CAF

The CAF at FNAL[1] is a cluster of dedicated nodes located at Fermilab, managed by the Condor[3] workload management system, with excellent network connectivity toward the CDF data handling system. Each worker node is configured to look exactly like a node in the CDF interactive pool, so any program that a user tests and runs on the interactive pool should run unchanged on any CAF node. No restrictions are made on the type of jobs it can run.

A set of CAF specific client utilities is used by the users to submit and monitor their jobs. Both classical batch-type monitoring as well as limited interactive monitoring is available. A Web server with most of the information is also deployed.

The CDF specific part is made of a set of dedicated processes, running on a reliable node near the cluster of worker nodes; we call it the head node. One of the processes accepts requests from the users, using a proprietary, kerberos based protocol and transforms them into submission commands for the workload management system. The other processes are needed for monitoring.

User jobs come complete with a tar ball, possibly created on the fly by the CAF client code, a list of parallel sections to run and the desired output location. The CAF portal then submits one job per section. Each section is also assigned its own wrapper, that extracts the tar ball on the worker node, serves as a monitoring portal, and tars and sends the user output to its final destination.

2.3 The dCAFs

The dCAFs are essentially CAFs located outside Fermilab. They are still dedicated pools of machines, with the look and feel of an interactive node, but the access to the CDF data handling system is different. Fermilab is the only place where all the CDF data are located, so each dCAF has at most a subset of these data. Data can be staged in on demand, but limits in the network bandwidth does encourage the use of pre-staged data for data intensive applications.

The submission interface is the same as for the CAF, the user has just to select the appropriate site. The user is thus in full control where his job will run, and can ask for the data to be pre-staged as needed.

2.4 The GlideCAF

As explained in the introduction, CDF needs to look at resources outside the dedicated pools. We have done a first step and developed a CAF based solution that can exploit resources of a shared pool, as long as it has a Globus Gatekeeper[4] installed, and a head node can be set-up near by, network wise. Availability of the CDF software distribution on the worker nodes is also highly recommended, although not needed by the CAF infrastructure per se.

The GlideCAF[5] is very similar to a dCAF, with just a slight modification of the Condor configuration, and one more CAF process. This new process is a glide-in factory; it keeps submitting condor glide-ins to the Gatekeeper. From the user point of view, it is just a regular dCAF.

The fundamental principle on which the GlideCAF works is based on the Condor glide-in mechanism. Once a glide-in starts on a worker node, it talks back to the head node and joins the CDF specific Condor pool. At this point, a regular CDF job can be started on that worker node as if it was a dedicated resource.

The advantage of using a glide-in is given by the fact that the glide-in can make a set of sanity checks before reporting back to the head node. Broken nodes can thus be eliminated by the glide-ins themselves and no CDF job is lost on them. Policymaking and monitoring is also greatly improved, since we are in complete control over our pool.

There are however some limitations to this approach. First of all, the network between the head node and the worker nodes needs to be very good. Moreover, right now a firewall is not allowed between the head and worker nodes, although we are working toward the removal of this limit.

2.5 FermiGrid

FermiGrid[6] is a Computing Division initiative to facilitate more efficient sharing of resources at FNAL between supported organizations, with very good networking between the participating clusters. FermiGrid is essentially an Open Science Grid (OSG)[7] site, with the advantage of being located at Fermilab. Currently, the CMS collaboration and the GP farms are allowing their resources to be used by FermiGrid jobs. CDF is also seriously considering putting part of its dedicated resources into the FermiGrid and use them by having higher priority on those resources.

We have three ways to access the FermiGrid resources; using OSG provided tools, using Condor-G[3] and using the GlideCAF infrastructure. The third solution is obviously the most easy to use for the CDF users, so we plan to use just that one. For more details on the GlideCAF, see the previous section.

2.6 The ccCAF

Right now, CDF has more than ten dCAFs in production, and the user needs to select the one he wants to run on. For lots of application, most if not all the dCAFs would do, so it would be useful to have a single submission point for all of them.

The ccCAF project is exactly this. As have been seen by now, most of our dCAFs are Condor based. We thus plan to use a feature, called Condor-C[3], that allows the delegation of a job from a Condor pool to the other.

The idea is to set up a CAF portal, with no worker nodes attached, and let the Condor-C system move the jobs to the least loaded pool. From the user point of view, nothing changes.

Moreover, the same mechanism can be used to exploit Grid resources, too. Following the GlideCAF recipe it is easy to create glide-in pools on the fly; you just need the glide-in factory. Once a glide-in based pool is set up, Condor-C can use this pool too. CMS is committed to develop the same idea, too.

2.7 The LcgCAF

Having to install lots of Condor pools is potentially heavy and difficult to maintain, so we are also looking at other, more Grid tools. The most powerful tool out there is the LCG[8] Resource Broker (RB); it can act as a single point of submission to all the LCG resources.

We are thus reimplementing the CAF infrastructure to make use of the RB while still maintaining the same user interfaces. From the user point of view nothing changes, but the CAF portal now talks with the RB and the Lcg Monitoring tools[9] to get the same result.

2.8 SamGrid

The SamGrid[10] is a product of the Computing Division developed for DZero. It uses its own submission interface, but is still a very much alive project that is adapting to the new realities, and is promising good integration with both LCG and OSG.

SamGrid uses SAM for both data handling and job scheduling. This allows it to do intelligent collocation of both data and CPU during resource matchmaking. Since CDF is using SAM for data handling anyhow, this could help us a lot.

The only problematic point is its user interface. Tying together data handling and job scheduling requires a lot of advance knowledge of the job run, limiting its use to only a very limited set of applications.

2.9 The other grids

The main grids currently deployed are LCG[8] and OSG[7]. They both have huge number of resources deployed, and a significant fraction could be used by CDF in an opportunistic way. They are however very much work in progress, and the available user interfaces are still very primitive. One solution is the use of Condor-G, which is pretty much Grid neutral and gives the user most of the current functionalities.

Another problem are the job failure rates, which are still quite high. However, given the amount of resources we could gain, it is certainly worth a good look.

As described in the previous sections, we are trying to hide most of the details of the Grids behind CAF portals, but direct submission will always be an option.

3 CDF usage patterns

The next thing to look at are the CDF usage patterns. To formulate the plans for adoption of the available technological solutions we introduce another matrix. This second matrix is shown in Table 2. One axis shows the usage pattern while the other shows a group of similar technological solutions. The numbers in the matrix show the relative priority of the technological solution for the referenced usage pattern, both in terms of ease of use and product readiness. The requirement and priorities of each end every usage pattern are described below.

	CAF FermiGrid GlideCAF	DCAF GlideCAF* CcCAF on DCAFs	LcgCAF CcCAF GlideCAF	SamGrid	Condor-G LCG OSG	Production Farm
User Analysis	1	2#				
User MC	X	1	2*			
Organized Analysis	1	3#	4#	4	4#	2!
Centralized MC	X	1	1	2	2	X
Real Time Production						1
Reprocessing	2&	2#	4#	3		1

* - Software distribution available
\# - Data pre-staged at sites
! - Only spare resources
& - If needed
X - Technically feasible, but not a good use of the resource

Table 2: Matrix of usage patterns vs. technological solutions

3.1 User Analysis

3.1.1 Requirements

User analysis benefits from significant processing resources, a flexible user environment, and broad access to data at a high rate. The user analysis requirements are essentially the same as the requirements used when developing the CAF.

- Users need to be able to develop code on local desktop and laptop resources. The environment is packaged and submitted for analysis. The user must be able to submit a customized arbitrary application. CDF software distribution is often required.
- Analysis jobs need to run asynchronously on remote batch resources. Currently the CAF users utilize 30 batch slots per person per day.
- Data is served to the application from local caches. Currently the data serving requirements as calculated from CAF experience is 2Mb/s per process.
- The users can interact with the job through a defined interface. Users have the ability to check the progress and debug processes.
- Results are returned to storage available to the users.

3.1.2 Solutions

Working through the technological solutions from the matrix in Table 2 there are some solutions that meet the requirements and a number that do not.

- The CAF is a dedicated user analysis resource that was built to meet the requirements. The CAF resources meet all the requirements for user analysis.
- The FermiGrid GlideCAF also meets all the requirements for user analysis. From the user point of view it is completely equivalent to the CAF.
- The group of distributed CAFs were also built to meet the requirements of user analysis. For all purposes, dedicated dCAFs and GlideCAFs with CDF software distribution installed should be considered equivalent. The dCAFs host only a selection of the CDF data, so analysis requests need to be matched to dCAF resources either manually or by using ccCAF over dCAFs.
- The LcgCAF and the ccCAF at large are not suitable for user analysis due to the lack of data and possibly lack of CDF software distribution. The second problem could also affect some of the more opportunistic GlideCAFs.
- There are two analysis requirements that SamGrid does not appear to be able to meet without additional development. The automatic packaging, deployment and execution of a arbitrary application code would need additional development. Additionally the deployment of the user analysis interactive functionality that the CDF user community has come to expect would require additional effort.
- Direct submission to the Grids has similar problems as LcgCAF and ccCAF and is not considered suitable for the same reasons.
- On the production farm, the services and resources meet the requirements for user analysis, but are rejected by the farm own requirements. The production farm requires very stable operations, which do not mesh well with also serving arbitrary user analysis applications.

3.1.3 Priorities

The CDF priorities for deploying user analysis computing are calculated using a number of input parameters. The effort required to implement or to continue to support the solution, the transparency to the user community, and the potential access to additional resources.

The top two CDF User Analysis Priorities are given below

1. The most efficient center for large-scale user analysis will continue to be the CAF. CDF should expect to spend operational effort supporting the CAF infrastructure and continue to increase the computing capacity within budgetary and facility constraints. Moving part of the dedicated nodes to a FermiGrid accessible pool and gain them back via the GlideCAF should also be seriously considered, both to reduce the wasted CPU cycles and to reduce our operational load in the long term. This will also allow us to better exploit any idle resources on the other Fermi clusters.
2. In addition to the CAF resources, the efficient use of dCAFs for user analysis should continue to be a priority, both in terms of dedicated and glide-in based sites. The broader adoption of SAM by analysis users should improve the ability to utilize remote centers. The current model is that data is allocated to dCAF sites in a fairly static fashion and the available data is published. The current CAF submitter requires that the user identifies the submission site. It would be desirable to use a SAM-based site selector as a component of the ccCAF.

3.2 User Submitted Simulation

3.2.1 Requirements

User submitted simulation benefits from significant processing resources and access to user controlled storage.

- User Simulation applications need access to conditions databases.
- The diversity of simulation applications is lower than analysis applications, but users must be able to customize the simulation application by developing on a local desktop and laptop. The environment needs to be packaged and submitted to simulation resources. CDF software distribution is often required.
- Simulation jobs need to run asynchronously on remote batch resources.
- Results are returned to user controlled storage.

Though not a technical requirement of the simulation, the experiment has defined a requirement that simulation applications should not consume resources that are more efficiently used for analysis applications. This influences the utilization of experiment owned farms and the configuration of the distributed database.

3.2.2 Solutions

Working through the technological solutions from the matrix in Table 2 there are some solutions that meet the requirements and a number that do not.

- The CAF can meet the technical requirements for user analysis as well as user simulation, but it is more efficient to use the CAF resources solely for analysis due to the availability of CDF data. The same stands true for the FermiGrid GlideCAF. User simulation is not seen as an appropriate use of the resources at Fermilab.

- The group of distributed CAFs were built to meet the requirements of both user analysis and user simulation. The dCAFs are the best match for user simulation. This is true for both dedicated and glide-in based dCAFs. The use of ccCAF would make their use even easier.
- The LcgCAF and the more generic ccCAF do also meet the requirements of the User Simulation, as long as the CDF software distribution is available to the worker nodes. Sites deploying AFS can be immediately used, while for other sites we are still looking at alternative methods.
- There are two simulation requirements that SamGrid does not appear to be able to meet without additional development. The automatic packaging, deployment and execution of a arbitrary application code would need additional development. Additionally the deployment of the user analysis interactive functionality that the CDF user community has come to expect would require additional effort.
- Direct submission to the grid has the same problems the SAMGrid has. CDF users are used to the CAF interface and a change would be difficult to impose.
- The production farm requirements for high availability and reliability as a dedicated resource prevent it from being considered for user simulation applications.

3.2.3 Priorities

The top two CDF User Simulation Priorities are given below

- The most efficient centers for large-scale user simulation should be the dCAF resources, both dedicated and glide-in based. The predictive environment of these sites should maximize the job success rate. The use of ccCAF would make their use even easier.
- The LcgCAF, the generic ccCAF and the most opportunistic GlideCAFs are also important for the CDF collaboration. The amount of available resources out there is so huge that it must be exploited even if the failure rate would result to be considerably higher than we are used to.

3.3 Organized Analysis

3.3.1 Requirements

Organized Analysis is similar to User Analysis in requirements for access to data, processing resources, and conditions information. The requirements on organized analysis in terms of total resources are generally higher, but in terms of functionality there are fewer requirements for organized analysis. Organized analysis applications are not typically undergoing active development, so there is not a need to develop and submit arbitrary applications, and the existing applications can be packaged better. There is a much smaller number of people executing organized analysis jobs with a greater expertise, so there is less of a need to maintain an interface consistent with the CAF.

3.3.2 Solutions

Working through the technological solutions from the matrix in Table 2 we see that all them meet the requirements for organized analysis. We are discussing each of them, anyhow.

- Both the CAF and the FermiGrid GlideCAF can meet the technical requirements for organized analysis. The organized analysis use-case is a simplification of the design use-case.
- The dCAFs, both dedicated and glide-in based, can be used for coordinated organized analysis. The limited availability of data on dCAF resources makes them less attractive than the CAF, but the dCAF clusters meet the requirements. The ccCAF meets the requirements but is not that important in this scenario.
- The LcgCAF and the ccCAF at large are problematic because of the problems for large scale data management for analysis. Other VOs that have successfully performed analysis applications on LCG have done so against statically resident datasets. The use of SAM could alleviate the problem, but without deployment of multiple SAM stations around the globe, and pre-staging of data, the efficiency of such a solution is not likely to be high.
- The issues that had been anticipated for SAMGrid for user analysis and simulation are no longer requirements of organized analysis. It should be possible to package the application for remote execution, but the existence of multiple SAM stations around the globe is still a requirement.
- Direct submission to the Grids has similar problems as LcgCAF and ccCAF, but it is technically feasible. If we solve the data handling problems before we develop the CAF portals, this should be considered.
- The production farm requirements for high availability and reliability as a dedicated resource prevent it from being considered for user analysis applications, but might permit it to be used for organized analysis. Organized analysis applications are typically well established and understood, so the chance of a rogue application causing problems with the production worker nodes is reduced. The people running organized analysis applications are usually expert, so the chances of a user mistake negatively influencing production jobs is also reduced. The production farm ability to access experiment data makes it an attractive resource for organized analysis. However, during running periods the production farm can be can be heavily utilized by event reconstruction and not many resources are left for the organized analysis.

3.3.3 Priorities

The Priorities of Organized Analysis are given below

1. Like for the large-scale user analysis, CAF and the FermiGrid GlideCAF will continue to be the most efficient places to run the organized analysis.
2. The second best place for the organized analysis is the production farm, using the spare CPU cycles not used by the real time production and/or reprocessing. The production farm is very similar to the CAF, so it can grow as needed by reconfiguring the worker nodes that are now part of the CAF.
3. Outside Fermilab, the dCAFs, both dedicated and glide-in based, represent the best place to run the organized analysis. Pre-staging of data is essential to make an efficient use of the resources, though. The ccCAF could be useful, but is not essential, due to the limited number of users involved.
4. Grid resources, accessed both in terms of CAF portals and direct submissions, are another resource we want to try to exploit, due to their large amount. The data handling is however a real problem here, and until we solve it, either by installing and maintaining several SAM stations around the globe, or by using the available Storage Elements, both CDF owned and opportunistically assigned, we are not likely

to achieve an acceptable efficiency for our jobs, and we are better leaving these resources for simulation jobs.

3.4 Centralized Simulation

3.4.1 Requirements

Centralized simulation production is a larger scale, but somewhat simplified use-case of user simulation. The people performing centralized simulation are expert and the application is well tested and stable. The number of events expected from centralized simulation is large.

- Large quantities of batch processing resources are needed by centralized simulation.
- Access to the conditions database information is required.
- Access to data storage resources is needed in two areas: resources are needed at the processing site to stage the results before migration to archival storage at a dedicated resource and they are needed at a dedicated CDF resources (typically the FNAL mass storage system).

3.4.2 Solutions

Working through the technological solutions from the matrix in Table 2 we can see that all of them meet the requirements for centralized simulation. Not all of them can be used though, and we are explaining why.

- The CAF and the FermiGrid GlideCAF can meet the technical requirements for centralized simulation, but it is more efficient to use the Fermilab resources solely for analysis due to the availability of CDF data. Centralized simulation is not seen as an appropriate use of the Fermilab resources.
- The group of distributed CAFs, both dedicated and glide-in based, are the best resource for centralized simulation. The use of ccCAF could make it marginally simpler, but is not seen as essential.
- The LcgCAF and the more generic ccCAF are also a good match for the centralized simulation. The delivery of the output files could potentially be troublesome, but a higher failure rate could be compensated by the amount of resources we can gain.
- The issues that had been anticipated for SAMGrid for user analysis and simulation are no longer requirements of centralized simulation. It is possible to package the application for remote execution and the existence of a local SAM station should also provide tools for managing data.
- The direct submission to the Grid has the same limits as LcgCAF and ccCAF and should be considered until we have the two CAF portals up and running.
- The production farm can meet the technical requirements for centralized simulation, and centralized simulation is predictable enough to meet the requirements of the production farm, but it is more efficient to use the Fermilab resources solely for analysis due to the availability of CDF data. Centralized simulation is not seen as an appropriate use of the production farm.

3.4.3 Priorities

The two CDF Organized Analysis Priorities are given below. The centralized simulation use case lends itself well to several technical solution and the prioritization of resource utilization and development effort should be performed with the knowledge of how many additional resources are available.

1. a) The dCAFs, both dedicated and glide-in based, are expected to be a significant resource for centralized simulation. These resources are already performing user simulation and central simulation is easier to support.
   b) Together with the dCAFs, LcgCAF and ccCAF are expected to be major players once deployed. Although they are not yet ready for real use, work is in progress on both of them, and the amount of resources they are expected to bring is huge.
2. While we wait for the grid CAF portals to be deployed, direct submission to the grid has high value. SamGrid should be easier to use than the other grids, but at present its size is just a fraction of the other ones.

3.5 Real Time Production

3.5.1 Requirements

The requirements of real time event production are to keep up with the incoming data from the detector and validate the data in time to identify and respond to problems.

- Provide sufficient computing resources and access to data to reconstruct events at a rate approximately equivalent to the data taking rate.
- Provide a sufficiently reliable environment to ensure that all events are processed with a low rate of resubmission and a reasonable operational load. A job failure rate of 1 in 1000 is seen as borderline, the target failure rate is 1 in 100,000.

3.5.2 Solutions and Priorities

There is only one solution identified for real time event production, and that is the continued use of a dedicated production farm. The need for reliable data access and a stable processing environment that is unperturbed by other applications dictates a somewhat insulated facility. High energy physics experiments have traditionally used dedicated farms for primary event reconstruction, though shared facilities are used for many other applications. The LHC experiments plan to continue to use dedicated farms for the initial reconstruction.

The decision facing the usage of the current production farm is whether any other applications should be allowed to opportunistically use the resources and if so what. It is possible that some predictable applications could be allowed to use the production farm resources, because there is structure in the farm usage.

3.6 Data Reprocessing

3.6.1 Requirements

The requirements of data reprocessing are to provide up sufficient data access and processing resources to successfully reprocess a desired dataset and perform the calculation in time to be useful for analysis applications.

- Provide sufficient computing resources and access to data to reconstruct events at a rate sufficient to meet the expectations of the experiment community. Currently 2-3 months of computing are required to reprocess the datasets requested. These numbers tend to grow with increasing luminosity, though time estimates of more than 6 months for reprocessing will frequently result in reprocessed data that may be obsolete before it is finished.
- Provide a sufficiently reliable environment to ensure that all events are processed with a low rate of resubmission and a reasonable operational load. To maintain a consistent dataset it must be possible to ensure that all events have been reprocessed.

3.6.2 Solutions

Once the real time nature has been removed as a requirement there are a larger number of solutions available for data reprocessing, that are not available for real time event reconstruction. Unlike event simulation activities, it is important that all jobs are processed and that every event is computed to avoid systematic effects in analysis due to reproducible job failures.

- Both the CAF and the FermiGrid GlideCAF can meet the technical requirements for reprocessing. The reprocessing use-case is a simplification of the design use-case. The CAF resources are the primary analysis resources of the experiment and would not be the first choice for reprocessing resources, even though they are technically capable of performing the task.
- The dCAFs, both dedicated and glide-in based, can be used for reprocessing. The limited availability of data on dCAF resources makes them less attractive than the CAF. Event reprocessing requires significant access to data, but the access is largely sequential, completely predictable, and lends itself well to pre-staging. The ccCAF also meets the requirements but is not that important in this scenario.
- As with the organized analysis, the LcgCAF and the ccCAF at large are problematic because of the problems for large scale data management. However, if the data handling problem is solved, they do meet the requirements.
- The success of the SamGrid reprocessing for DZero is a good indication that SamGrid could be a useful resource for CDF event reprocessing. The integration of the data management and workflow management should allow for a reasonably quick adoption. An assessment of the true number of processing resources that could be made available to CDF through the SamGrid interface should be made.
- Direct submission to the Grids has similar problems as LcgCAF and ccCAF, plus it has a potentially higher failure rate and much worse monitoring. It thus not considered reliable enough to satisfy the requirements.
- The production farm was designed for real time event production so by definition it is capable of performing event reprocessing. Traditionally, the production farm was the primary CDF resource for reprocessing. There are two potential limitations. The first is that the production farm is usually well utilized during data taking and reprocessing is generally performed during down-time periods. The second issue is that as the integrated luminosity increases the total resources needed for reprocessing continues to grow. If CDF intends to reprocess the complete dataset additional resources will be required.

3.6.3 Priorities

The Priorities of the CDF Reprocessing are given below. The reprocessing use case lends itself well to several technical solution and the prioritization of resource utilization and development effort should be performed with the knowledge of how many additional resources are available.

1. The use of the production farm is the most promising solution for CDF event reprocessing. The production farm has served as the primary resource for Run2. The similarities between the production farm and the CAF would also allow for a fluid transition of processing resources between the two.
2. The next obvious place to run are the CAF, the FermiGrid GlideCAF and the dCAF, both dedicated and glide-in based. However, the CAF resources are the primary analysis resources of the experiment and should be used only if really needed.
3. The third most promising resource for reprocessing is the use of SAMGrid. DZero has clearly demonstrated success with SamGrid for reprocessing. The use of SAM for data handling in CDF make SamGrid a reasonable choice for distributed event reprocessing, provided a sufficient amount of resources can be gathered this way.
4. The final set of resources that could be used for reprocessing is by means of LcgCAF and ccCAF. The data handling challenges make it a difficult task, but the amount of resources we could gain is huge.

4 Overall priorities

Having described the technical solutions available and the usage cases involved, including the requirements, possible solutions and the relative priorities of each one, we now need to make an experiment wise priority list. Working through the technological solutions from the matrix in Table 2 we can see that some of the solutions are top priorities for one or more usage cases, while others play just a low priority role on most of them. To make the decision process simpler, we introduce another table, where the relative priorities are converted into scores, using exponential values and giving 1st priority a score of 10000 and 4th priority a score of 10. The sum of the scores can then be used as a rough estimate of the importance of a class of solutions. The table can be found in Table 3.

The CDF-wise priorities are given below.

1. The top priority for CDF remains the production farm. Real time production is essential for the experiment and the production farm is the only resource that meets the requirements. It is also very important for both reprocessing and organized analysis, both of which serve the CDF community as a whole.
2. Next by priority are the local and remote CAFs. The CAF infrastructure hides well the implementation details, so both (d)CAFs on dedicated and shared resources are equally valuable for the collaboration. The dedicated pools are however loosing appeal to the funding agencies, so CDF needs to exploit as much shared resources as possible. The GlideCAF approach has demonstrated its potential at the Italian LCG Tier-1 in Bologna and other sites are preparing to follow the same route. Last but not least, the FermiGrid incarnation is promising to bring together all the resources at Fermilab, vastly increasing the utilization of the resources most valuable for the CDF analysis.
3. Another important resource for CDF are the Grids, especially LCG and OSG, due to the huge amount of resources they hold. When paired together with the CAF

infrastructure, they also become easier to use and more reliable. So it is very important for CDF that the LcgCAF, the ccCAF and a multi-site GlideCAF become a reality in the shortest possible time.
Even with the CAF infrastructure, two problems continue to limit the kind of applications the Grid resources can support:

- Data handling is still a serious problem, limiting the use of the Grid resources to simulation jobs. A close integration with the CDF data handling group is needed to lift this restriction.
- Resources that lack the CDF software distribution are instead limited only to well packaged jobs. Although most of the organized activities can so this, most of the user jobs will not be able to use these resources. Luckily, several Grid sites use AFS and the CDF software distribution can be accessed from there, but for the others a different approach must be found.

4. While SamGrid holds great promise for some organized tasks, its very production oriented interface prevents its use for more user oriented task. The amount of resources available to CDF only via SamGrid is also not that clear and is estimated to be just a small fraction of the other grids. For that reason, SamGrid is not considered a high priority for CDF.
5. Direct submission to the other grids is also not considered a high priority for CDF. Projects for hiding the complexities and differences of the grids behind CAF portals are well underway, so the direct submission is only appealing for the short term and as a backup solution.

	CAF FermiGrid GlideCAF	**DCAF GlideCAF CcCAF on DCAFs**	**LcgCAF CcCAF GlideCAF**	**SamGrid**	**Condor-G LCG OSG**	**Production Farm**
User Analysis	10000	1000				
User MC		10000	1000/2			
Organized Analysis	10000	100/2	10/2	10	10/2	1000
Centralized MC		10000	10000	1000	1000	
Real Time Production						10000*2
Reprocessing	1000/2	1000/2	10/2	100		10000
	20500	**21550**	**11010**	**1110**	**1005**	**31000**

Table 3: Technological solutions scores

5 Schedule and Timeline

The top priority resources, i.e. the production farm, the main CAF and the currently deployed dCAFs, are all in good shape and are delivering the necessary resources needed at present. However, the requirements of the collaboration are growing fast and the scaling potential of these facilities is limited. We really need to start to exploit shared resources, both CDF funded and opportunistically available ones.

The first steps in this direction were moved at the Italian LCG Tier-1 center, where a GlideCAF is in production. The dCAF is using both CDF provided resources, as well as getting spare cycles from the resources contributed by the LHC experiments, increasing by a factor of 3 the amount of resources available to the CDF users in Italy.

A similar GlideCAF has been put in place on the FermiGrid and is using spare resources of the CMS and GP Farms pools. The added amount of resources is not huge, but it is a good prototype that gives us confidence that CDF could contribute resources to the FermiGrid and get them back using this mechanism. If the GlideCAF continues to behave the way it is behaving now, we will definitely go this path as soon as the new worker nodes arrive.

However, the real potential for growth is in the Grids. It is not unreasonable to think that we could double the amount of resources we current use, just by using the spare cycles of LCG and OSG. Moreover, several funding agencies have expressed their intention to put most of the computing funding in Grid accessible resources, so we will need to access them that way, if we want to grow. As stated in the previous section, the stability and monitoring capabilities of the CAF portal are highly desirable, so we here list the CAF projects already deployed or currently underway.

- The first step in this direction was the GlideCAF, and we can continue to use this solution for any larger site that has a strong CDF community behind. However, in the present form, it is not suitable to exploit generic Grid sites.
  Nevertheless, sites like Barcelona, Lion, Kahrlsruhe and Wisconsin has expressed their interest in installing a GlideCAF. Other sites, like CNAF and UCSD, are considering switch off their dedicated dCAFs and move to a pure GlideCAF approach.
- To get more flexibility, the LcgCAF is being developed. It is using the LCG Resource Broker (RB) to run the jobs on all the LCG resources. The Italian team in charge of the development is well integrated with the INFNGrid developers, so the project has high chances to deliver a usable and scalable product. The alpha version of LcgCAF is already deployed and is able to run CDF jobs on most INFN grid sites, with limited monitoring. Several more months of testing and fine tuning are still needed to make it production quality, but the target date for a production release is end of 2005.
- We still like the robustness and flexibility of the Condor workload management system, so we are going use it, too, to exploit the Grid resources. The easier approach seem to be an extension of the GlideCAF, i.e. the possibility to us a single GlideCAF portal to access resources on multiple Grid sites, bridging the firewalls as needed. The Condor project has a service called the Generic Connection Brokering (GCB), that holds great promise and we are committed to give it a try. If it works as described, we could get a working system in a few months, with the target of releasing a production multi-site GlideCAF before the end of 2005.

- The other Condor service is Condor-C. It allows for a job being moved seamlessly between different Condor pools, and would allow us to have a single submission point for all the Condor pools we control. The first and most obvious use would be to link all the dCAFs, being dedicated or glide-in based, and create a single submission point for all of them. However, the even more promising solution would involve the creation of dynamic, lightweight glide-in Condor pools at the Grid sites, without the need of a local CAF portal, and integrate them together with the others we own. We plan to work together with CMS to implement this second scenario and we have set a target date for the first working prototype for November 2005.
  However, making intelligent routing of jobs between different Condor pools is not a trivial task, and although we expect to have a first usable ccCAF deployed before the end of 2005, it will probably take quite a bit longer to make it really usable for generic applications.

Waiting for the CAF portals to materialize, we invite the production groups to look at the user interfaces of the various grids, if they need more resources than currently deployed via the dCAFs. No development effort will be put into this by the CDF computing group, since the grid user interfaces are ready to be used, and each group has different needs.

The last development project we want to pursue is understand better how to access the CDF software distribution from the Grid resources. As said before, some grid resources will have AFS installed, and will not be problematic. But the vast majority will not, so an alternative solution must be found. Currently, the most promising product out there is Parrot[11], that allows access from an arbitrary application to remote data as if they were local. This is achieved without any change to the application or support of the operation system. If it works as described, it would allow us to run arbitrary CDF jobs on the Grid resources.

6 References

[1] E. Lipeles, M. Neubauer, I. Sfiligoi, F. Wuerthwein, “CAF based on Condor”, CDF Note 7088 (2004)

[2] J. Antos et.al., “Data production of a large Linux PC Farm for the CDF experiment”, CDF Note 7640 (2005)

[3] “Condor Project Homepage”, http://www.cs.wisc.edu/condor

[4] “The Globus Toolkit”, http://www.globus.org/toolkit/

[5] S. Sarkar and I. Sfiligoi, “GlideCNAF – A purely Condor Glide-in based CAF”, CDF Note 7630 (2005)

[6] “The FermiGrid Home page”, http://fermigrid.fnal.gov/

[7] “The OSG Home page”, http://www.opensciencegrid.org/

[8] “The gLite Home page”, http://glite.web.cern.ch/glite/

[9] “The GridICE Home page”, http://infnforge.cnaf.infn.it/gridice/

[10] “SAM-GRID Home page”, http://www-d0.fnal.gov/computing/grid/

[11] “The Parrot Home page”, http://www.cse.nd.edu/~ccl/software/parrot/