

Grid Computing at the DØ Experiment

T. Kurča (for the DØ collaboration)

Abstract—DØ is a pioneer in grid computing for large scale production activities involving the handling of collider data samples. A data grid (SAM) has been used since the start of Tevatron Run II as the sole means of data transport (enabling local offsite analysis). The focus of the computational grid (SAM-Grid) so far has been on production activities. Integration of SAM-Grid with other grids, like LCG and OSG are ongoing projects. All Monte Carlo data are produced off-site. In 2005 and 2007 large fractions of the Run IIa and Run IIb data sets respectively (>1 billion events) were reprocessed using native SAM-Grid, LCG and OSG resources. The value of grid computing to the DØ experiment is conservatively estimated at roughly \$4M/year. Evolution towards full grid computing and lessons learned from these activities will be discussed.

I. INTRODUCTION

The DØ detector [1] is one of the two large colliding beams detectors at Fermilab. Large computing resources are required for the processing and the analysis of Petabytes of data collected at the Tevatron collider.

The DØ computing model is based on distributed computing from its origin and was designed to handle large amounts of data. The model is built on the SAM Data Management System (Sequential Access via Metadata) [2].

Basic characteristics of the DØ computing model are:

- primary data processing is done at Fermilab [3]
- all Monte Carlo data are produced remotely
- all data are centralized at Fermilab
- there is no automatic data replication
- remote analysis centres are usually prestaging data of interest

The three most important phases in the evolution of the DØ computing model are:

1. The SAM data grid - usage of SAM data handling services optimized for data delivery for analysis jobs of users distributed around the world.
2. Creation of the native SAM-Grid [4]. Standard grid middleware was integrated with the SAM system and a real computing grid for DØ institutes was created.
3. Interoperability between the SAM-Grid and LCG [5], OSG (Open Science Grid) [6] respectively. Usage of computing resources without the need of specific DØ installations at foreign sites.

II. THE SAM DATA MANAGEMENT SYSTEM

The SAM project has started by DØ at Fermilab in 1997 with the goal to address the data handling challenge of the Run II. Later as the system grew more configurable and operationally stable, in 2001 also CDF experiment [7] opted to adopt SAM for its data handling needs. Today the SAM system manages a throughput of Petabytes of data per month throughout dozens sites in America, Europe and Asia [8].

The SAM system is based on a set of servers (stations) distributed around the world to communicate data traffic between storage resources (Enstore [9], HPSS [10], dCache [11]) deployed at participating institutes by the data transfer means SRM [12] and gridftp. The DØ experiment has or shares storage resources at Fermilab, CC-IN2P3 [13], Imperial College and WestGrid [14]. Central Oracle DB is located at Fermilab and is used by SAM for file namespace and metadata registration. A user can run his analysis jobs on remote sites with installed SAM services. He does not have to care about the data location. SAM provides data delivery and very detailed bookkeeping about the consumed or missed data.

The SAM project is designed and implemented with four principal goals in mind.

1. Provide reliable storage for data coming either directly from the detector or from the data processing facilities around the world.
2. Enable data distribution among all the collaborating institutions, today on the order of 70 per experiment.
3. Thoroughly catalogue the data for content, provenance, status, location, processing history, user defined datasets etc.
4. Manage the distributed resources in order to optimize their usage and the data throughput, while enforcing the administrative policies of experiments.

III. THE SAM-GRID

In 2001 a new project of enhancing SAM functionalities to the real computational grid has started. The goal of having common run-time environment, submission interface, and monitoring tools was achieved by integration of standard grid middleware, such as Condor-G and the Globus Toolkit, with software developed at Fermilab. This move toward a more grid-like architecture had to be achieved without jeopardizing production quality service for the ongoing experimental physics program at Fermilab.

Manuscript received November 26, 2007.

T. Kurča is with the IPNL, Universite Lyon 1, CNRS/IN2P3, Villeurbanne, France and Universite de Lyon, Lyon, France; 43 Bd du 11 Novembre 1918 F-696222 Villeurbanne, Cedex (telephone :+33-4-72-44-84-43, e-mail :kurca@in2p3.fr)

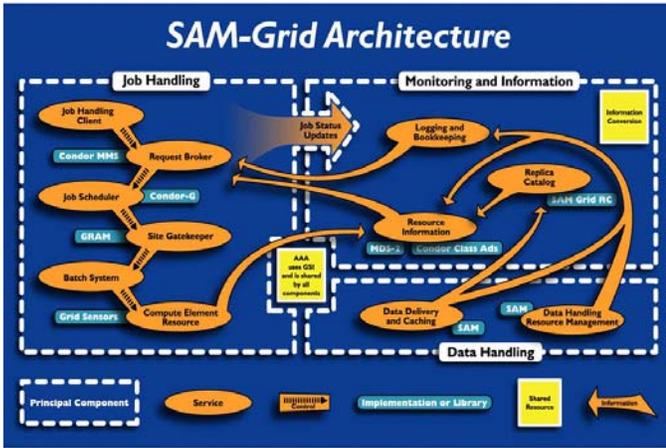


Fig. 1. Three principal components of the SAM-Grid architecture: Data and job handling, monitoring –information system.

The three major components of the new system are: data handling (SAM), job, and the information management (JIM) systems. The SAM-Grid does require DØ specific installations on the gateway node at remote sites (SAM station, job manager), but it does not require any preinstalled software or running daemons at the worker nodes of the cluster. The SAM-Grid grid level services include the resource selection service, the global data handling service, such as metadata and replica catalogue, and the submission services, which are responsible for maintaining the queue of grid jobs and for interaction with the remote resources. This global grid layer interacts via the GRAM interface with the local, fabric layer. The SAM-Grid has developed its own job-managers adhering to the GRAM protocol. This interface adapts the generic directives of the grid services to the peculiarities of the fabric configuration at different sites. The fabric services include the local data handling and storage services, the local monitoring and the local job scheduler.

Because the software infrastructure at each site is now uniform and is adapted to the local fabric configuration, the maintenance necessary to run production consists of a single grid administrator with contact persons at each site. In some cases the privileged access is needed. This is a significant improvement on the pre-grid model, where every site needed a dedicated person responsible for maintaining the local submission scripts and for submitting the jobs locally. In the SAM-Grid model a single user can submit from his client node to any collaborating site.

IV. SAM-GRID - LCG/OSG INTEROPERABILITY

At the end of 2004 the SAM-Grid/LCG integration project was started. The goal of the project was to make the LCG resources available to DØ through the SAM-Grid system with minimal requirements on new developments.

About two years later the project of interfacing SAM-Grid with OSG had been started. The SAM-Grid interoperability with the both grids LCG and OSG is based on the same principles and uses the flexibility of an additional layer between local batch system and the grid batch adapter.

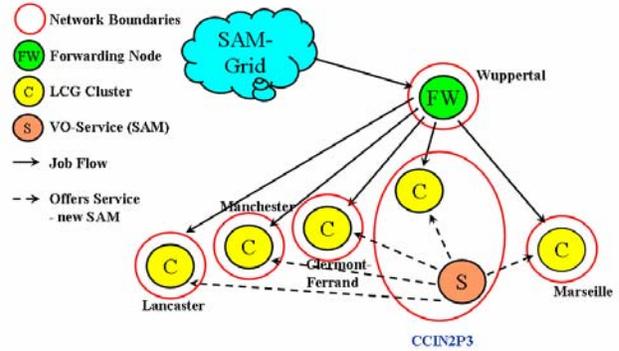


Fig. 2. Architecture of the working installation of the SAM-Grid/LCG interoperability project. The SAM station at CC-IN2P3 Lyon ensures VO-specific services for jobs running at different LCG sites.

The proposed architecture provides a job forwarding mechanism from the SAM-Grid to LCG/OSG. The two main new system components are:

1. The forwarding node provides a gateway to the whole LCG/OSG world. Multiplicity of this node can help to overcome potential scalability limits.
2. Remote SAM-Grid VO-specific services (data handling, monitoring, sandboxing) are provided by a SAM gateway node (SAM station). A single node can serve multiple LCG/OSG sites at a time. This is different in the native SAM-Grid model, where SAM services have to be instantiated at each execution site.

A forwarding node acts as an interface between SAM-Grid and LCG/OSG. For the SAM-Grid a forwarding node is an execution site, or in other words, a gateway to LCG/OSG resources. Jobs submitted to the forwarding node are adapted and submitted to LCG/OSG through its user interface. LCG/OSG clusters are seen by SAM-Grid as other batch systems. Jobs running at remote worker nodes interact via the dedicated SAM station with the DØ data management system. Through this station they retrieve input data and store the resulting files in SAM (see Fig. 2).

V. PRODUCTION ACTIVITIES

The two principal production activities of each high energy physics experiment are the simulation, i.e. production of Monte Carlo data, and the reconstruction of real data registered by the detector. With the better understanding of the detector performance, improvements in the detector

calibration and the software algorithms, the need for a secondary reconstruction or a reprocessing arises periodically.

Usually the primary data processing continues when the reprocessing of older data is required. At the same time also Monte Carlo production cannot be stopped. Accumulation of those requests leads to the very large demands on the computing resources, not available at DØ dedicated clusters. Interoperability of SAM-Grid with the LCG and OSG grids makes those resources, on the opportunistic basis, available for the DØ.

A. Large Scale Challenges

Over the past several years the DØ experiment has used distributed computing for large scale projects in both the simulation and the data reconstruction. The SAM data handling system allows the DØ experiment a very efficient method of storing and accessing data. Over 450 billion events have been processed by SAM since the start of Run II, when the DØ has begun to use SAM.

- In order to analyze the real data, large Monte Carlo samples are required. Monte Carlo simulation is a continuous, permanent process in the life of each high energy physics experiment. Using about 2.5 THz of distributed computing resources allows the DØ experiment to simulate currently up to 15 million events per week.
- First major reprocessing activity dates back to 2003, when the DST (Data Summary Tape) data were processed without the need to access the calibration DB (data base). Advantages of distributed computing on the SAM data “grid” were fully used. More than 100 million events were processed remotely using about 2 THz of DØ CPU resources.
- In 2005 a large reprocessing of over 1.5 billion collider events was done, which required 3.4 THz of CPU resources for 6 months. This processing was done on the raw data, so the access to calibration DB was needed. Installations of proxy DB servers on the participating sites helped to reduce the load on the Fermilab central DB. A second fixing of the data was able to process 1.4 billion of events in only 6 weeks.
- After the new detector upgrades and improved detector calibration, a second reprocessing of the Run IIb data was required. In 2007 for the first time, in parallel with native SAM-Grid sites at CC-IN2P3 and at WestGrid, DØ has used OSG and LCG resources at a significant level for this reprocessing.

VI. EXPERIENCE, LESSONS LEARNED

Experience and lessons learned mainly from the last, most complex project, the 2007 reprocessing, are summarized in the following paragraphs. But they are valid and can be generalized also for the most of our previous data challenges.

A. System Commissioning

Particularly challenging for those activities is the large variety of computing resources, their difference in size and in the network connectivity. The problem is to select the optimal resources for the DØ requirements on availability, reliability and accessibility. Our general approach to this problem is iterative, i.e. testing and adding sites to our resources one at a time. We have selected only sites with good network connectivity to storage resources and a good stability in time.

The sites were classified according to their bandwidth and latency and appropriate data transfer queues in the SAM were created. This way we prevented sites with good connectivity from waiting for the data because of “slow sites”.

Network traffic was further optimized for applications with different input patterns – like reprocessing and merging jobs. Sites with NFS installations proved to be not suitable for the jobs running I/O intensive applications.

B. Site Certification

The large diversity of used resources makes it necessary to verify that the physics results from the different clusters are identical. For this purpose the so called certification procedure was developed. The same data sets were processed on each participating cluster and the basic physics distribution histograms were compared to the reference ones. Because of the need of manual intervention, site certification was a relatively long process, lasting, in average one week. For the future the tools for automatic comparison with reference histograms would be desirable.

C. Data Accessibility

Data reprocessing jobs require two types of input data. The application, about 800 MB in size, and the raw data file, in average 300 MB.

In many cases, the application could be cached at storages local to the computing sites. However, for sites where local storages were not available, the application had to be transported over the WAN from the “closest” grid storage. In OSG, 8 sites out of 12 had local storages; 4 of them were available for grid access and 4 for local access only (3 via SRM interfaces, 1 via NFS). On the other hand, input data were always transported from Fermilab, where a Mass Storage System, Enstore, holds the entire data set for the experiment. It is not desirable caching such data because it is processed once, unless application failures occur. The output from all the jobs was transported to temporary storages at Fermilab, where it was merged.

D. Monitoring and Troubleshooting Operations

Monitoring and troubleshooting the system was one of the challenges of the operations. Job failures were caused by three major factors:

- Site/Grid (OSG/LCG) problems: these were typically site gateway and worker nodes configuration

problems. The site administrators and the OSG troubleshooting team were responsible for addressing these.

- Data delivery problems: these were either data handling services problems or storage element problems. The SAM system group was responsible for the former, the site administrators and the OSG troubleshooting team for the latter. In many occasions, only careful inspections of log files could distinguish between the two cases.
- Application failures: these were caused by software bugs or corrupted input data files. The DØ data reprocessing operation team was responsible for addressing these in collaboration with the DØ offline software group.

In order to investigate and properly triaging the problems, a monitoring system was developed to attempt an automatic categorization of the failures. The system plotted the histogram of the output size of all jobs submitted in 5 day intervals. Because the data reprocessing application is the same for every job, the length of its output strongly correlates with the type of failure for the job. Fig. 3 shows an example of such histograms.

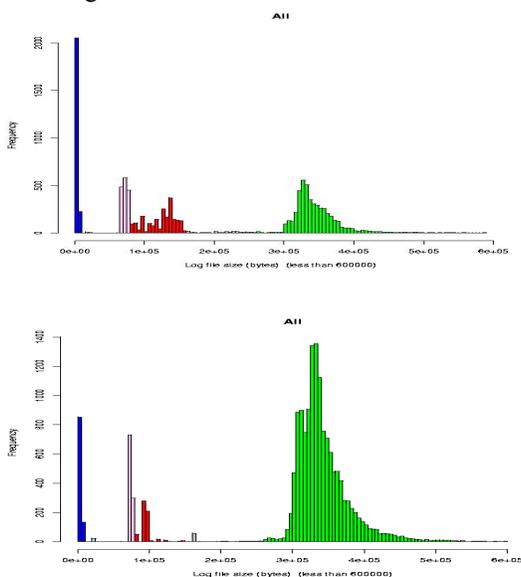


Fig. 3. Failure source analysis based on the log-file size before (above) and after (below) troubleshooting exercise. Different colours correspond to different sources of errors and different log-files sizes.
 Blue 0-8 kB: OSG CE or Worker Node configuration problem, lost standard output.
 Aqua 8-25 kB: service/hardware failure; could not start bootstrap executable
 Pink 25-80 kB: SAM problem: could not get application, possibly raw files
 Red 80-160 kB: SAM problem: could not get raw files, possibly application
 Gray 160-250 kB Possible DØ application crash
 Green >250 kB Success

A process for triaging problems to the right group had to be developed. In particular for problems on the OSG, we could take advantage of the ticketing service offered by the OSG Grid Operation Center (GOC) [15]. Problems submitted to the system are tracked and followed up for resolutions. Despite the help from GOC, the sheer number of resources made it impossible for the single person that we could dedicate from

our group to keep up with site problems. The OSG Troubleshooting team was therefore involved to interact with system administrators, investigate failures, propose solutions, follow up with the resolution, etc. The team was instrumental to the success of the data reprocessing activity. Fig. 3 shows the reduction of OSG system problems after the OSG troubleshooting team was involved.

D.1 Number of running jobs, efficiency

When looking at Fig. 4, showing the number of reconstruction jobs entering different grids, we realize rather different efficiency rates. Explanation for this difference lies in the complexity and in the maturity of those systems. Native SAM-Grid is the most mature and without the additional layers of complexity which are typical for interfacing SAM-Grid with LCG and OSG respectively.

LCG was also more advanced compared to OSG. Utilisation of LCG resources was already tuned in the past. We have been using LCG sites for data fixing and for Monte Carlo production.

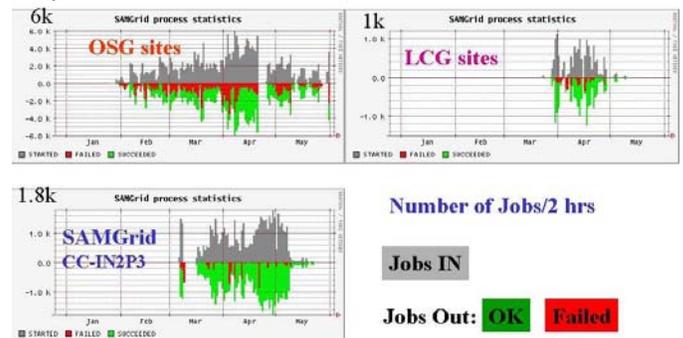


Fig. 4. Number of reconstruction jobs running on different grids used for reprocessing and their success rate. LCG sites were used just during 1 months with the goal to accelerate the reprocessing.

E. More Efficient Usage of CPU Resources

Standard grid-fabric interfaces, job-managers from Globus Toolkit suite proved to be insufficient to run production-quality jobs on the SAM-Grid. To overcome problems of flexibility, scalability, comprehensiveness and robustness, the SAM-Grid interaction with the local batch system is mediated via the batch adapter. The batch adapter is configured during its installation. Using this additional layer of abstraction, called batch “idealizer”, the SAM-Grid jobs are able to profit from peculiarities of local batch systems.

E.g. special option of the BQS [16] batch system at CC-IN2P3 is used to inform the scheduler that job plans to use HPSS, the local mass storage system. In case of HPSS downtime, the batch system will hold those jobs, avoiding the crashes due to unavailability of requested data.

BQS offers optimal usage of CPU resources by defining the required computational resources for each job. The most important ones are memory usage, time duration of the job and scratch space requirements. Different BQS job classes correspond to different resources requirements. DØ reconstruction and merging applications differ in the memory,

scratch space and CPU time requirements (Fig. 5). In reprocessing 2007 we lost this advantage of different characteristics of reconstruction and merge jobs when DØ decided to add another application, reconstruction certification, to the merging job flow. The consequence was a long duration of this workflow, jobs were sometimes running a few days and were killed because of the wall time limit. This had another consequence of slowing down the whole reprocessing. The durable location (disk space) filled with unmerged thumbnails was not liberated quickly enough and the submission of reprocessing jobs had to be limited.

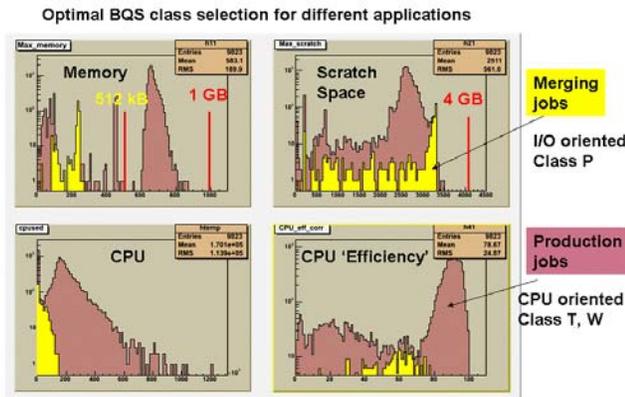


Fig. 5. Production and merging jobs, different applications, were submitted to different BQS batch system classes. Merging jobs are I/O intensive operations and require less than 512 kB of worker node memory. With this low memory requirement merging jobs enter faster into the running state.

VII. CONCLUSIONS

The DØ data reprocessing effort faced different types of challenges in operating the infrastructure in the phases before and after the completion of resource commissioning. Both phases of operations lasted several months.

During the phase of commissioning, about 50% of job failures were due to configuration problems at OSG sites and about 50% to data delivery problems. For the data reprocessing application, a job is considered successful if the produced data are successfully stored to the durable or permanent storage.

Typical site configuration problems included computing element access authentication and authorization failures, scratch areas permission or size problems, system library incompatibilities, and wrong reports of the job status (middleware failures). In addition, despite the OSG process to report cluster downtimes to the Grid Operation Centre, unscheduled downtimes affected operations for the duration of the activity. Most configuration problems were addressed with the help of the OSG troubleshooting team. The lesson learned is that one should not undergo computing activities of such a magnitude without the support of a troubleshooting team acting as a liaison between the user and the system administrators.

Data delivery problems were mainly due to lack of storage systems local to the computing sites and insufficient network connectivity to storages. In our model, local storage systems were used to cache the application. Given its large size and the hundreds of concurrent jobs potentially running on each cluster, uncontrolled access from worker nodes to a shared file system (a typical configuration on many clusters) tended to make the system unstable. It is preferable storing the application in a local storage system like dCache and accessing it via an SRM interface.

Addressing data delivery problems, we learned that sites must be categorized according to their connectivity to global storages. Requests for data access from “slow” sites must be queued together, separately from requests from “fast” sites. In addition, as expected, sites with poor connectivity to storages are useless to run data intensive applications.

As the efforts toward commissioning resources diminished, the resource pool became more stable. Configuration and data delivery problems started to become more seldom. In this more reliable environment, problems at the global level became more apparent. In particular, the lack of a stable Grid-level resource selection service manifested in over- and under-subscription of cluster usage. Resource selection, in fact, was left to the operators of the infrastructure, responsible for job submission. This resulted in a less than optimal utilization of the resources as some clusters received fewer jobs than they could process, while others queued up jobs that eventually failed since grid services, such as data handling, had timed out. We learned that for a computational challenge of this magnitude, an automatic resource selection system is necessary to reduce the need for job recovery and for simplifying operations.

Another lesson learned in the final weeks of the activity is that DØ data reprocessing operations would have been more efficient if most job recoveries had been spread in time. This consideration is probably valid for all workflows that include an operationally intensive phase of failure recovery. Despite the automation of the job recovery procedure, in fact, identifying jobs to be resubmitted was considered a human intensive operation. Recovering jobs shortly after job failures, would have avoided a final “tail” of intensive operations right at the end the activity, a time where personnel focus tends to dwindle.

Daily production rate goal of 3 million events has been largely surpassed as seen in Fig. 6. In total 450 million events were reprocessed and made available to physicists for the summer 2007 conferences. Fig. 7 shows the integrated number of events produced vs. time.

VIII. SUMMARY

The DØ collaboration in the past several years has accomplished more large scale data challenges. In the last reprocessing, running from February to May 2007, about 90 TB of data were processed using fully distributed resources of OSG, LCG, CC-IN2P3, WestGrid, and Fermilab. In the case of OSG/LCG the resources were used on the opportunistic basis.

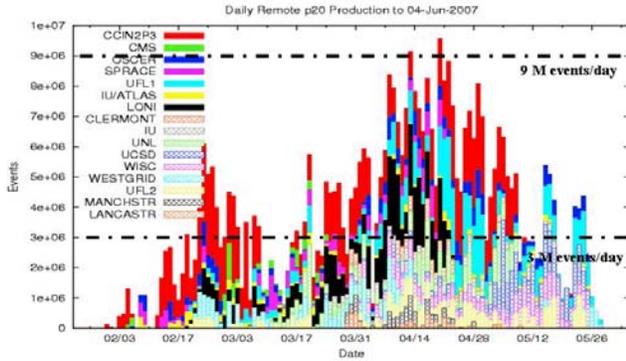


Fig. 6. Daily remote production contributions from the participating sites. The goal of 3 million events per day was largely surpassed.

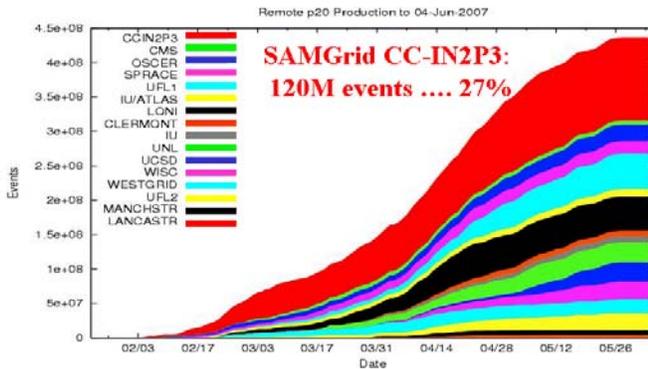


Fig. 7. Total integrated production gain per day. About 65% of all reprocessed data was done on different OSG clusters. But still a significant fraction was reprocessed via the native SAM-Grid at CC-IN2P3. The remaining fraction was processed on WestGrid (native SAM-Grid) and LCG clusters.

We described the challenges of system commissioning, troubleshooting, and operations. Commissioning was approached as an iterative problem. Resources were added a site at a time, categorizing their network connectivity to storages and comparing the output of physics results with standard references. Troubleshooting required the development of a monitoring tool to categorize failures. Site configuration problems have been addressed with help from the OSG troubleshooting team. Operations were coordinated via the SAM-Grid system. The system understands the DØ workflow and its requirements and coordinates the selection of computing as well as storage resources.

Last but not least, we have to stress the very important human factor having a large effect on the success of each large project involving complex layers of different computational environments. Local administrators, SAM-Grid experts and LCG/OSG experts have to work very closely in order to solve the problems quickly and efficiently.

IX. FUTURE PLANS

The goal for the end of 2007 is to reach stable operation of the SAM-Grid/LCG(OSG) interfaces. These are being

intensively used for continuing Monte Carlo production. In addition also the primary data processing is being moved to the SAM-Grid interface with OSG. This step is necessary as the Fermilab is aggressively promoting the FermiGrid project – creation of a local computing grid from the local resources. Maintenance of existing functionality, the performance optimization and further automation are high priority tasks for the future.

Next application to be enabled in the grid environment is skimming (data selection based on different physics and/or trigger criteria). As the manpower is a real issue at DØ, moving the data analysis application to the grid is an open question. It requires further development, deployment and operations effort.

ACKNOWLEDGMENT

I would like to thank all the members of the DØ collaboration who have been helping us with the SAM-Grid deployment and operations, as well as all members of LCG and OSG consortia for their support in all phases of our SAM-Grid – LCG/OSG interoperability projects.

Special thanks are going to the SAM-Grid team of Fermilab whose constant development efforts, their involvement in the deployment and operation of those complex grid systems made the success of the DØ grid computing possible.

REFERENCES

- [1] The DØ Collab., "The DØ Upgrade: The Detector and its Physics", Fermilab Pub-96/357-E.
- [2] SAM: <http://www-d0.fnal.gov/computing/sam/>
- [3] Fermilab: <http://www.fnal.gov/>
- [4] G. Garzoglio et al., "The project: architecture and plan.", Nuclear Instruments and Methods in Physics Research, Section A, NIM A14225, vol. 502/2-3 pp 423 – 425; G. Garzoglio "A Globally Distributed System for Job, Data, and Information Handling for High-Energy Physics", Ph.D. Thesis, DePaul University, Chicago, March 2006, Fermilab-Thesis-2005-32
- [5] J. Apostolakis et al., "Architecture Blueprint Requirements Technical Assessment Group (RTAG)", Report of the LHC Computing Grid Project, CERN, Oct. 2002
- [6] OSG: <http://www.opensciencegrid.org>
- [7] CDF: <http://www-cdf.fnal.gov>
- [8] SAM-Grid: <http://samgrid.fnal.gov:8080/>
- [9] Enstore: <http://www-isd.fnal.gov/enstore/>
- [10] HPSS: http://cc.in2p3.fr/rubrique335.html?var_recherche=HPSS?lang=fr
- [11] M. Ernst et al., "Managed Data Storage and Data Access Services for Data Grids", Chep 2004, Interlaken, Switzerland, Sep. 2004
- [12] I. Bird et al., "SRM (Storage Resource Manager) Joint Functional Design", Global Grid Forum Document, GGF4, Toronto, Feb. 2002
- [13] CC-IN2P3: http://cc.in2p3.fr/cc_accueil.php3?lang=en
- [14] Westgrid: <http://www.westgrid.ca/>
- [15] GOC: www.grid.iu.edu/
- [16] BQS: http://cc.in2p3.fr/rubrique351.html?var_recherche=BQS?lang=fr