

# THE SAM-GRID / LCG INTEROPERABILITY SYSTEM: A BRIDGE BETWEEN TWO GRIDS

Gabriele Garzoglio<sup>\*</sup>, Andrew Baranovski, Parag Mhashilkar, FNAL, Batavia, IL 60510, USA  
Tibor Kurca<sup>†</sup>, IPN / CCIN2P3, Lyon, France  
Frédéric Villeneuve-Séguier, Imperial College, London, UK  
Anoop Rajendra, Sudhamsh Reddy, University of Texas at Arlington, Arlington, TX 76019, USA  
Torsten Harenberg, University of Wuppertal, Wuppertal, Germany

## *Abstract*

The SAM-Grid system is an integrated data, job, and information management infrastructure. The SAM-Grid addresses the distributed computing needs of the experiments of RunII at Fermilab. The system typically relies on SAM-Grid services deployed at the remote facilities in order to manage computing resources. Such deployment requires special agreements with each resource provider and it is a labour intensive process. On the other hand, the DZero VO has also access to computing resources through the LCG infrastructure. In this context, resource sharing agreements and the deployment of standard middleware are negotiated within the framework of the EGEE project.

The SAM-Grid / LCG interoperability project was started to let DZero users retain the user-friendliness of the SAM-Grid interface, allowing, at the same time, access to the LCG pool of resources. This "bridging" between grids is beneficial for both the SAM-Grid and LCG, since it minimizes the deployment efforts of the SAM-Grid team and exercises the LCG computing infrastructure with data intensive production applications of a running experiment.

The interoperability system is centred on job "forwarding" nodes, which receive jobs prepared by the SAM-Grid and submit them to LCG. We discuss the architecture of the system and how it addresses inherent issues of service accessibility and scalability. We also present the operational and support challenges that arise to operate the system in production.

## INTRODUCTION

The SAM-Grid system [1] is the meta-computing infrastructure used by the Run II experiments at Fermilab. It provides distributed data, job, and information management services. The system relies on central services, maintained at Fermilab, as well as distributed services, deployed at the computing clusters of the collaborating institutions. As grid technologies become part of the standard middleware available at computing

centres, computing resources become more easily accessible. Today, these standard services are the preferential ways the SAM-Grid manages resources on the grid, whereas, in the past, deployment of SAM-Grid specific services was the only way to access computing resources.

Some features of the SAM-Grid system are of fundamental importance for the computing of Run II experiments and, even in a grid environment, they must be preserved. This paper describes how the SAM-Grid has been integrated with the LHC Computing Grid (LCG) environment, so that a wider range of resources are made accessible to the Run II experiment, still preserving crucial feature of the SAM-Grid.

This paper is organized as follows. We first describe what features of the SAM-Grid system are important for the Run II experiments. We then describe the architecture of the interoperability system and how it has been deployed. Before concluding, we report our experience and lessons learned on operating the system.

## THE SAM-GRID SYSTEM

The Run II experiments rely on several features of the SAM-Grid system for their computing activities. For this reason, the goal of the integration with LCG was retaining the critical features of the SAM-Grid framework, enabling, at the same time, access to the pool of resources deployed by EGEE. These critical SAM-Grid features are summarized hereby.

### *Integrated data handling*

The SAM-Grid system is fully integrated with SAM [2], the data handling system of the Run II experiments. The SAM system provides four essential services for the experiments:

1. reliable data storage, either directly from the detector or from data processing facilities around the world
2. data distribution to and from all of the collaborating institutions, today on the order of 70 per experiment
3. data cataloguing for content, provenance, status, location, processing history, user-defined datasets, etc.
4. distributed resources management, in order to optimize usage and, ultimately, data throughput,

<sup>\*</sup> garzoglio@fnal.gov

<sup>†</sup> on leave from IEP SAS Kosice, Slovakia

enforcing, at the same time, the policies of the experiments.

### Integrated Application Management

The SAM-Grid system has knowledge of the typical applications running on the system [3]. This knowledge is used to optimize resource usage and to enforce experiment policies. In detail, the SAM-Grid provides:

- Job Environment Preparation: dynamic software deployment, configuration management, and workflow management
- Application-sensitive Policies: the SAM-Grid allows the implementation of different policies on data access and local job management. More in detail, different types of applications can access data through different data access queues, each configured with its own policy settings. In addition, different types of applications can be submitted to a local scheduler using different local policies (generally enforced using different job queues)
- Job Aggregation: the job request to the system is automatically split at the level of the local scheduler into multiple parallel instances of the same process. The multiple jobs are aggregated and presented to the user as the single initial request. This allows resource optimizations and user friendliness in the management of the job.

### SAM-GRID TO LCG JOB FORWARDING

In order to maintain the advantages of the SAM-Grid system, using at the same time the resources provided by LCG, we have implemented the following architecture.

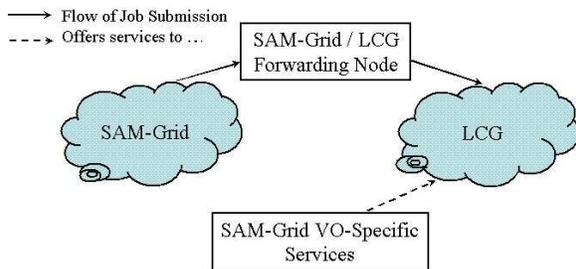


Figure 1: A high-level diagram of the SAM-Grid to LCG forwarding architecture.

Forwarding nodes act as an interface between the SAM-Grid and LCG. To the SAM-Grid, a forwarding node is an execution site, or, in other words, a gateway to computing resources. Jobs submitted to the forwarding node are submitted in turn to LCG, using the LCG user interface. LCG jobs are in turn dispatched to LCG resources through the LCG Resource Broker. A VO-specific service, SAM<sub>i</sub>, offers remote data handling services to jobs running on LCG.

The multiplicity of resources and services is represented in the diagram below.

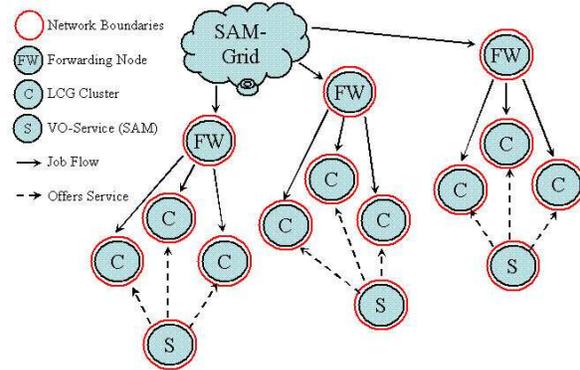


Figure 2: Multiplicity diagram of the forwarding architecture.

This same architecture is currently being deployed to integrate the SAM-Grid system with the Open Science Grid.

Main issues to consider when implementing this architecture are service accessibility, usability of the resources, and scalability. We discuss these issues in the section on “problem faced and lessons learned”.

### Production Configuration

The system is used in production to run DZero montecarlo and data reprocessing jobs. The configuration of the system is the following:

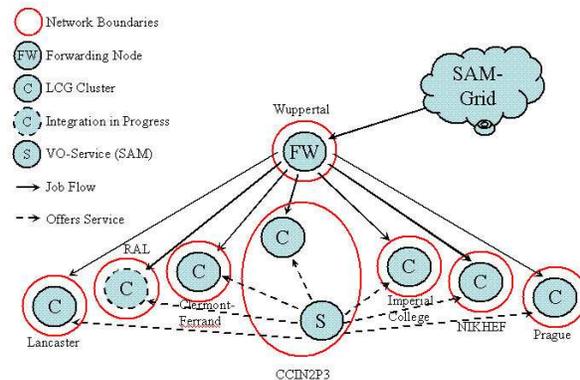


Figure 3: Diagram of the forwarding architecture for the production system.

The system runs hundreds of jobs per day processing hundreds of Gigabytes of data.

### PROBLEMS FACED AND LESSONS LEARNED

Deploying and operating the SAM-Grid to LCG forwarding infrastructure exposed a series of problems. We expose hereby the list of the most relevant issues.

#### Local cluster configuration

Configuration problems on even a single worker node on the grid can significantly lower the job success rate [4]. These worker nodes tend to fail jobs very quickly,

thus appearing to the batch system often in “idle” state. All queued jobs, therefore, tend to be submitted to the failing nodes, with catastrophic consequences for the job success rate.

Typical configuration problems at worker nodes include time asynchrony, which causes security problems, and scratch disk management problems, such as “disk full” errors.

### *Scratch management is responsibility of the site OR the application*

DZero jobs impose the following requirements on the local scratch space management system. Jobs typically fail in writing scratch information on network file systems, such as NFS, because of intensive I/O. Therefore, scratch space must be locally mounted to the worker node. In addition, jobs typically need more than 4 GB of local space.

SAM-Grid uses job wrappers to do “smart” scratch management, in order to find a scratch area that satisfies the requirements above. Possible choices for scratch management areas are made available to the job through the LCG job managers (environment variables \$TMPDIR, etc.). Sites that accept jobs from DZero must support this configuration of the job managers.

### *Grid services configuration*

- Resubmission of non-reentrant jobs: Some jobs should not be resubmitted in case of failure and must be recovered as a separate activity. We experienced problems overriding retrials of job submission from the LCG Job Description File and the User Interface configuration.
- Broker input sandbox space management: on some brokers, disk space was not properly cleaned up, requiring administrative intervention to resume the job submission activity.

### *Handling of user credentials for job forwarding*

The forwarding node accepts jobs from the SAM-Grid via the GRAM protocol (Globus gatekeeper). The user credentials are made available at the forwarding node by delegating them to the gatekeeper. These delegated user credentials, though, have limited privileges and cannot be used directly to submit grid jobs to LCG.

We use an online credential repository (MyProxy) to address the problem. Users upload their credentials to MyProxy before submitting the job. After the job has entered the forwarding node, the delegated limited credentials of the user are used to retrieve full privileged credentials from MyProxy. These fresh credentials are then used to submit the job to LCG.

### *Job Failure Analysis*

We experienced difficulties in analyzing the output of failed jobs. In particular, we could not retrieve the output of “aborted” jobs (“Maradona” server fails in handling the output).

### *Scheduling policies for “clusters” of jobs are difficult to express on LCG*

Jobs submitted to the SAM-Grid tend to be “large”. The SAM-Grid needs to split these jobs into parallel instances of the same process in order to execute them in a reasonable time.

These “clusters” of jobs tend to have the same characteristics and, in our experience, are most efficiently executed on the same computing cluster.

Since the LCG Job Description Language does not provide ways of referencing previously scheduled jobs, it is challenging to schedule such job clusters on the same cluster.

### *SAM data handling configuration*

We have experienced problems with three aspects of the data handling services:

- Service accessibility: SAM had to be modified to allow service accessibility for jobs within private networks (pull-based vs. call-back interfaces).
- Communication reliability: In order to serve jobs running on the grid, SAM is configured to accept TCP-based communications only, as UDP does not work in practice on the WAN.
- System usability: Sites hosting the SAM data handling system must allow incoming network traffic from the forwarding node and from all LCG clusters (worker nodes) to allow data handling control and transport. The SAM system should be modified to provide port range control.

### *Certification of LCG for DZero computing activities*

The experiments typically run cluster certification procedures for some computing activities. For example, for DZero data reprocessing, clusters are certified by processing a well known dataset and comparing its output with a reference result.

Through the forwarding node, the SAM-Grid “sees” LCG as single large cluster. System certification, therefore, could in principle be done on the system as a whole, rather than on a cluster-by-cluster basis, as it is done today.

Certification procedures for computing systems are highly discussed topics within the DZero collaboration.

### *Operation and support of the SAM-Grid / LCG interoperability system*

In DZero, institutions get credit for the computing cycles used by the collaboration. Collaborators at an institution tend to run their share of operations submitting jobs to their facility. Collaborators that run “operations” are responsible for the production of the data (routine job submission/monitoring, troubleshooting, facility maintenance and upgrade, etc.) and are the contact point for the support of the system at that facility.

The collaboration is discussing whether this operational and accounting model can be reused on the grid, where

jobs can run on institutions that are not part of the collaboration.

## CONCLUSIONS

Users of the SAM-Grid have access to the pool of LCG resources via the “interoperability” system described hereby. This mechanism increases the resources available to the DZero collaboration without increasing the cost of system deployment.

The SAM-Grid is responsible for job preparation, for data handling, and for interfacing the users to the grid. LCG is responsible for job handling (resource selection and scheduling).

DZero is using the system for production activities. We have described the problems and lessons learned operating the infrastructure.

## REFERENCES

- [1] I. Terekhov et al., "Meta-Computing at D0", Nuclear Instruments and Methods in Physics Research, Section A, NIMA14225, vol. 502/2-3 pp 402 - 406.
- [2] V. White et al., "D0 Data Handling", in Proceedings of Computing in High-Energy and Nuclear Physics (CHEP01), Beijing, China, Sep. 2001
- [3] Gabriele Garzoglio, Andrew Baranovski, Parag Mhashilkar, Ljubomir Perković, Anoop Rajendra, “A Case for Application-Aware Grid Services”, in Proceedings of Computing in High Energy Physics 2006 (CHEP06), Mumbai, India, Feb 2006
- [4] A. Nishandar, D. Levine, S. Jain, G. Garzoglio, I. Terekhov, "Black Hole Effect: Detection and Mitigation of Application Failures due to Incompatible Execution Environment in Computational Grids", in Proceedings of Cluster Computing and Grid 2005 (CCGrid05), Cardiff, UK, May 2005