The SAM-Grid / LCG interoperability system: a bridge between two Grids

Gabriele Garzoglio, Andrew Baranovski, Parag Mhashilkar
Anoop Rajendra*, Sudhamsh Reddy *, Tibor Kurca **, Torsten Harenberg ***
Computing Division, Fermilab, Batavia, IL; * University of Texas at Arlington, TX; ** IPN / CCIN2P3, Lyon, France; *** University of Wuppertal, Wuppertal, Germany

INTRODUCTION

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 labor 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 centered around 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.

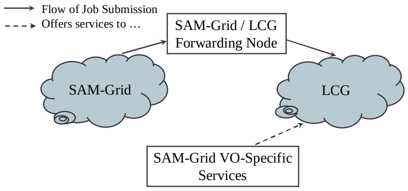
WHY NOT SUBMITTING JOBS TO LCG DIRECTLY?

SAM-Grid implements features that make the system user friendly for the RunII experiments.

- *Transparent access to the data*: the SAM-Grid system is fully integrated with SAM, the data handling system of the experiments;
- *Integrated Application Management*: the SAM-Grid system has knowledge of the typical applications running on the system. 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.

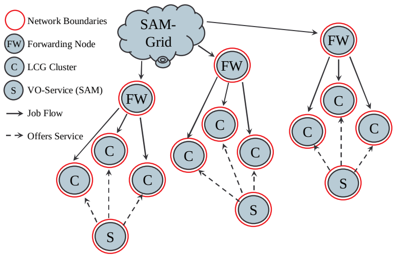
SAM-GRID TO LCG JOB FORWARDING

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.



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 SAM installation offers data handling services to jobs running on LCG.

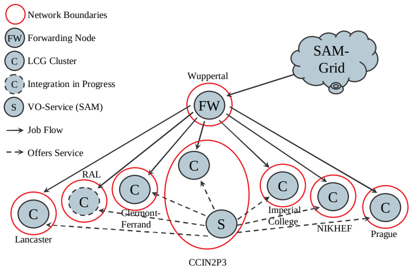
The multiplicity of resources and services is represented in the diagram below.



Main issues to consider when implementing this architecture are service accessibility, usability of the resources, and scalability.

SAM-GRID TO LCG FORWARDING SYSTEM: THE PRODUCTION CONFIGURATION

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



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. This is the list of the most relevant issues.

Local cluster configuration
Misconfiguration of even a single worker node on the grid can significantly lowers the job success rate.
Misconfigured worker nodes tend to fail jobs very quickly, thus appearing to the batch system often in "idle" state.
All jobs queued, therefore, tend to be submitted to the misconfigured nodes, with catastrofic consequences for the job success rate.
Typical worker nodes misconfigurations include time asynchrony (causes security problems) and scratch disk management problems (disk full).

Scratch management is responsibility of the site OR the application
DZero has special requirements on local scratch space management. The jobs cannot run on NFS because of intensive I/O and they 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. 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 JDL and the UI 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). A problem to overcome is that user credentials, delegated to the gatekeeper at the forwarding node, are limited 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 unlimited 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 job "clusters" 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, scheduling such job clusters all on the same site is challenging.

SAM data handling configuration
- *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
For some computing activities, the experiments run cluster certification procedures. For example, for DZero data reprocessing, the SAM-Grid clusters process a well known dataset, then physicists compare the output with reference results.
What does it mean to certify LCG? Should every individual cluster be certified? Should the LCG as a whole be certified? We currently certify every LCG cluster to which SAM-Grid submits jobs.

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. Running "operations" means being responsible for the production of the data (routine job submission/monitoring, troubleshooting, facility maintenance and upgrade, …), being the contact point for the support at that facility.
It is not clear that 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.