The SAM-GRID project: architecture and plan.

Andrew Baranovski[1], Gabriele Garzoglio[1], Hannu Koutaniemi[4], Lee Lueking[1], Siddharth Patil[2], Ruth Pordes[1], Abhishek Rana[2], Igor Terekhov[1], Sinisa Veseli[1], Jae Yu[2], Rod Walker[3], Vicky White[1]

[1]*Computing Division, Fermi National Accelerator Laboratory, Batavia, IL, USA*

[2]*University of Texas at Arlington, Arlington, USA*

[3]*Imperial College, London, UK*

[4]*Espoo-Vantaa Institute of Technology, Espoo, Finland*

Abstract

SAM is a robust distributed file-based data management and access service, fully integrated with the DZero experiment at Fermilab and in phase of evaluation at the CDF experiment. The goal of the SAM-Grid project is to fully enable distributed computing for the experiments. The architecture of the project is composed of three primary functional blocks: the job handling, data handling, and monitoring and information services. Job handling and monitoring/information services are built on top of standard grid technologies (Condor-G/Globus Toolkit), which are integrated with the data handling system, provided by SAM. The plan is devised to provide the users incrementally increasing levels of capability over the next two years.

1. Introduction

Goal of the SAM/Grid project is enabling fully distributed computing for DZero and CDF, by enhancing the distributed data handling system of the experiments (SAM), incorporating standard Grid tools and protocols, and developing new solutions for Grid computing.

The project is funded by PPDG and GridPP. Within these organizations, our group is working with both computer scientists, like the Condor Team, and physicists, like UTA and Imperial College teams. The Project is also taking advantage of the collaboration with other groups working on Grid technologies, such as EDG and DataTAG. Through these collaborations, we promote interoperability and code re-use via modularization and standardization.

2. The SAM & JIM Architecture

The architecture of the project results from three major considerations

1. Both the DZero and CDF experiments have large and widely geographically distributed collaborations;
2. SAM is a mature data handling system, distributed in nature, which we believe is well suited to play a major role in a Grid-like environment, for its proven qualities of scalability, robustness and flexibility;
3. The development teams of both Condor-G[I] and the Globus Toolkit[II], the Grid technologies that the architecture exploits, are part of PPDG and GridPP, like the SAM Grid project. We believe we are in a favorable position to influence the development of such

software packages to best match our needs.

There are 3 principal novelties of this architecture, with respect to other grid projects like the European DataGrid[III]:

1. The central role of SAM as the data handling system of the Grid;
2. The critical role of the Monitoring and Information Services to serve as information repository/provider for both the data handling and job management components;
3. The use of the Condor Match Making Service (MMS) as the core of the Resource Broker. This approach leads to a very "thin" Brokering Service Layer, as opposed to other designs that incorporates into it the ability of manipulating "general" (and not always defined) knowledge about the Grid.

Fig. 1 shows the components of the architecture.

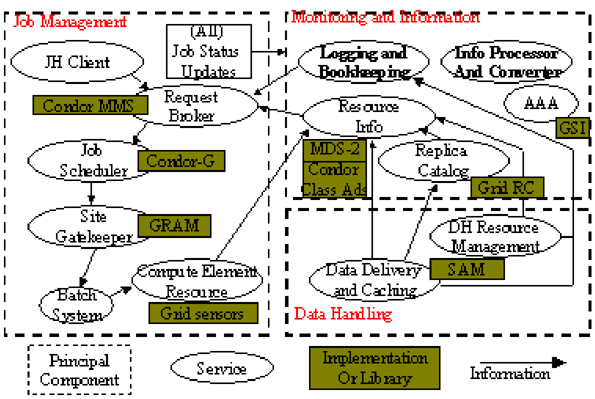

Fig. 1: The components of the SAM/Grid Architecture

The Data Handling System: SAM.

The core component of a SAM installation is the Station. The Station is a set of services that tie together a collection of resources dedicated to the experiment, such as network, mass storage systems, disk servers, and batch systems. The Station is responsible for retrieving and storing files exploiting the resources available to it as well as interacting with other Stations. Remote resources, in fact, are accessible to a Station via delegation to other Stations. During this process, which involves file routing, data replication is achieved by the use of disk caches. The reader can access more information at [IV].

The Job Management.

The architecture of the Job Management System can be broken down into the components shown in Fig. 1.

- **The user interface** is the client of the Job Handling system. Its input is the user provided description of the job via the Job Description Language (JDL) of the SAM Grid. This language must be expressive enough to represent the structure of any Monte Carlo and Analysis job of DZero and CDF. We are collaborating with the University of Arlington, Texas, in order to define this structure, which, together with the schema used to describe the resources of the Grid, leads to the JDL[V]. We are considering the use of DAGMan[VI] to implement the management of the job structure via CondorG.
- **The Request Broker** is based on the Match Making Service of Condor. The Condor MMS is a technology that is proven stable as a local batch system component. This architecture exploits the same technology at the level of the global environment.

  In order to use the Condor MMS for this purpose, two new features have been added to CondorG

  1. The possibility of specifying the target gatekeeper for the job at the time of the job/resource match, thus enabling automatic brokerage of the resource;
  2. The possibility of calling external scripts at the time of the job/resource match. We use this feature to define a rank for each match, for example

querying the Information System. These scripts implement the "intelligence" of the Brokering Service in a modular fashion.

- **The Job Submission Service** is implemented via CondorG. The service offers reliable submission of jobs to remote resources via the GRAM protocol. It also provides a persistent queuing mechanism in case the machine where this service runs needs to be shut down. Also, at the time of submission, CondorG, using GRAM, transparently starts up a GASS[VII] server to manage the transport of standard input/output and error.
- **GRAM[VIII]** is the submission protocol used by the Job Submission Service. We use the Globus Toolkit implementation of the GRAM clients/servers.
- The **Advertisement of Resources** is managed by Grid sensors that publish static and small-size dynamic collective information in the form of ClassAds**.**

The Monitoring and Information System.

- The **Resource Information Service** is based on top of 2 components, MDS[IX] and the Condor Collector. The architecture of MDS, based on LDAP, is well suited for monitoring run-time parameters, because it conveniently treats information as stale after a tunable expiration time. The Condor Collector, instead, transparently makes available to the Match Maker the characteristics of the resources advertised by the Grid sensors. An intermediate protocol translation layer, which uses XML as *lingua franca*, manages information exchange between the two services.
- **Logging and Bookkeeping services** are currently implemented using the logging facilities of SAM. We envision interfacing the Match Maker to these services to enable brokering decisions based on the statistical analysis of the history of the grid usage/condition.
- The **Replica Catalog** is implemented as a global resource via SAM. We look with interest to the developments of standard grid-enabled distributed RC services.

3. Current Infrastructure and future Plans

We currently run a prototype grid of 6 clusters, distributed between Illinois, Texas and the UK. We plan to expand such infrastructure as part of the project collaborative effort.

The 2-yr term goals of the project can be identified as:

- the reliable execution of structured, locally distributed Monte Carlo and SAM analysis jobs with basic brokering;
- Scheduling criteria for data-intensive jobs, full Job Handling – Data Handling interaction;
- Fully Distributed Monitoring and Information Services for Structured Jobs and Data Handling.

[I] The home page: http://www.cs.wisc.edu/condor

[II] The home page: http://www.globus.org

[III] The home page: http://eu-datagrid.web.cern.ch/eu-datagrid

[IV] Meta-Computing at DØ, Igor Terekhov et al., ACAT 2002, Proceedings

[V] Proposal for DØ Job Description Language, D. Meyer *et al.*, Internal Note 2002

[VI] The web page from Condor: http://www.cs.wisc.edu/condor/dagman

[VII] The web page from the Globus Toolkit: http://www.globus.org/gass

[VIII] The web page from the Globus Toolkit: http://www.globus.org/gram

[IX] A Directory Service for Configuring High-Performance Distributed Computations, S. Fitzgerald *et al.*, HPDC 1997. ftp://ftp.globus.org/pub/globus/papers/hpdc97-mds.pdf