

Enabling “Minimal Resource Brokering” on the SAM-Grid

Gabriele Garzoglio

Nov 20 2007

Introduction

As part of the DZero Grid Production Computing Initiative, DZero has requested to evaluate SAM-Grid resource brokering capabilities. The request is formulated as follows:

Request Title: “Minimal Resource Brokering”

Status: Functionality is not uniform, among LCG, OSG, or native SAM-Grid sites

Attached Documentation: none

Description: SAM-Grid should be able to avoid/suppress/delay submission to sites (native, LCG or OSG) which can't handle additional input. It is suggested to use the number of pending=no-yet-running batch jobs as an indicator. The limit per site may be advertised individually.

This document describes the brokering capabilities of SAM-Grid within this context. It notes that job submission delays can be achieved with the current infrastructure and discusses how these capabilities address the DZero request.

Summary

The current configuration of the SAM-Grid brokering system allows sites (Native SAM-Grid and OSG/LCG Forwarding Nodes) to define

- the maximum number of jobs authorized to be running at the same time
- the maximum number of jobs authorized to be in “submitting” state at the same time

These parameters can be configured by site administrators changing the SAM-Grid site configuration.

These features satisfy the request by the DZero collaboration.

Basic SAM-Grid Resource Brokering

The SAM-Grid system provides mechanisms to select SAM-Grid “Execution” resources. An Execution Resource, or Execution Site, is a set of computing and storage resources available to the SAM-Grid system to run jobs. Execution Sites can be of two kinds:

- 1) Native SAM-Grid: the execution environment is a local batch system with dedicated storage resources. All jobs submitted are executed within a single local cluster.
- 2) Grid Forwarding Node: the execution environment is a Grid (OSG or LCG) with distributed storage resources. Grid jobs are forwarded to different clusters.

It should be noted that when the SAM-Grid broker selects a forwarding node for job execution, the underlying grid (OSG or LCG) will undergo an internal process of resource selection. This process is transparent to the SAM-Grid broker and does not

affect the process of resource selection described in this document.

Execution sites advertise their characteristics to the SAM-Grid broker. These characteristics include the Grid entry-point (GRAM URL) to access site computing resources, the name of the SAM station that the jobs will use for data handling services, and the URL of the local XML database that the jobs will use to send monitoring events.

Users select Execution sites by adding a list of SAM station names to the Job Description File, when submitting jobs. This list is translated into a set of constraints for the resource broker. The broker will randomly select one of the sites that advertise a station name specified in the user's list. Typically users select execution sites by specifying a single station name.

This basic resource brokering allows jobs to specify constraints to the broker, in order to select resources with the desired SAM stations. In turn, resources can specify constraints to the broker, in order to select jobs with the desired characteristics. This reciprocal mechanism is used to regulate the flow of jobs from the SAM-Grid to the site and it is described in detail in the next sections.

Controlling the Rate of Job Submission to SAM-Grid Sites.

The SAM-Grid broker is based on the Condor Match Making Service. Job and resource characteristics are described in the form of classads, a set of "attribute = value" pairs. When matching jobs to resources, Condor considers "requirement" expressions from both job and resource classads; if either requirement conditions evaluate to FALSE, the match fails and the job will not be submitted to that resource.

This mechanism is used by the SAM-Grid to regulate the flow of jobs, as described in the white paper "The SAM-Grid Job Matching Policy: Proposed Design" [1]. This section summarized the relevant parts of the white paper to show how SAM-Grid controls the flow of jobs to an execution site (Native or Forwarding interfaces)

The simplest way to address the DZero request of limiting the number of jobs running at a site could be implemented as follows (in reality, as shown in the next section, the current implementation offers more control). Sites could advertise

- **CurrentJobs**: the total number of jobs running on the system. This can be obtained by the jim_advertise process with the ps command.
- **MaxJobs**: the maximum number of running jobs at the site as per site policy. It can be configured as part of the site configuration
- **Requirements** = (CurrentJobs < MaxJobs)

Every 5 minutes, the match maker evaluates all jobs / resource matches: this process is called match making cycle. By evaluating the "requirements" attribute above, the match maker would check that the given site does not run more jobs than the policy limit. In other words, sites with the "Requirements" attribute evaluating to FALSE would not be considered for job submission.

The following section discusses the current implementation of the SAM-Grid brokering.

Current implementation

DZero expressed the need to stop job submission if a certain number of jobs is already running at a site. Another important aspect of job management is stopping job submission when too many jobs are in "submitting" state at the site. Jobs are in "submitting" state

when the Execution Site interface is in the process of submitting them to the underlying job handling system (Batch System ,OSG, or LCG). This process is very resource intensive and is controlled together with the limit on the total number of jobs running at a site. This is how this is achieved.

The site publishes a classad to the match maker every 5 minutes. On top of the standard site information discussed in the previous sections, the site also advertises:

- **MaxJobs**: the maximum number of jobs at the site in any state (Running + Submitting). It is configured as part of the site configuration
- **MaxSubmittingJobs**: the maximum number of jobs in “Submitting” state, as allowed by policy. It can be configured as part of the site configuration
- **CurrentJobs**: the total number of jobs currently running on the system. This value is updated every 5 minutes. This is obtained by the jim_advertise process with the ps command.
- **CurrentSubmittingJobs**: the current number of jobs in “Submitting” state. This value is updated every 5 minutes. This is obtained by the jim_advertise process with the ps command.
- **CurMatches**: this attribute is incremented by the match maker every time a new job is submitted to the site. The initial value for this attribute is set to CurrentJobs.
- **JobsMatchedSinceLastAdvertisement** = CurMatches - CurrentJobs : this is a positive number, initially 0, that increases as jobs are submitted to the resource.
- **Requirements** = (CurMatches < MaxJobs) && (JobsMatchedSinceLastAdvertisement + CurrentSubmittingJobs) < MaxSubmittingJobs

The match maker will stop submitting jobs to the site when the Requirements attribute evaluates to FALSE. This happens in two circumstances:

- 1) when the total number of running jobs + the jobs submitted during the current match making cycle (CurMatches) is equal to or greater than the total jobs allowed at the site by policy (MaxJobs)
- 2) when the total number of jobs in Submitting state (CurrentSubmittingJobs) + the jobs submitted during this match making cycle (JobsMatchedSinceLastAdvertisement) is equal to or greater than the total jobs allowed in Submitting state (MaxSubmittingJobs)

This brokering policy satisfies the DZero requirement on minimal resource brokering.

References

- [1] G. Garzoglio, P. Mhashilkar, D. Wicke, “The SAM-Grid Job Matching Policy: Proposed Design”, White Paper at <http://www-d0.fnal.gov/computing/grid/doc/JobSubmissionPolicies.pdf>