Open Science Grid

Engage Virtual Organization

Match Making with ReSS

Mats Rynge, RENCI

<rynge@renci.org>

Revision 1 – May 7, 2007

Table of Contents

1. Overview

The Engage VO has users with many different kinds of codes. For simple serial applications without any dependencies between the jobs, a matchmaking system is provided to distribute the jobs across available resources. This system is modeled after similar systems from USCMS[1], DZero and Star, and is based on Condor and ReSS.

The system is controlled by a set of cron jobs:

1. Every 5 minutes: pull ReSS ads using the command: condor_status -any -long constraint 'StringlistIMember("VO:engage";GlueCEAccessControlBaseRule)' -pool osg-ress-1.fnal.gov -direct osg-ress-1.fnal.gov
2. Every 4 hours: verify the sites from step 1. Details about verification can be found below.
3. Every 2 minutes: results from 1 and 2, together with some additional variables (CurMatches, MaxMatches, ...), are combined into the final ad and inserted into the local Condor install with condor_advertise.

2. Querying ReSS

Before ReSS became readily available, Engage had a system which pulled VORS for information, verified sites, and the published the result in the local Condor install. When Engage transitioned to use ReSS, it was decided that validating the sites was very useful, and should be kept. The result was that instead of using the ReSS ads directly, Engage would fetch the ads, validate and in some cases modify the content of the ad, and the publish the result in the local Condor installation.

Every 5 minutes, ReSS is queried with this command:

```
condor_status -any -long -constraint \
  'StringlistIMember(\"VO:engage\";GlueCEAccessControlBaseRule)'  \
  -pool osg-ress-1.fnal.gov -direct osg-ress-1.fnal.gov
```

1 http://www.cs.wisc.edu/condor/USCMS_matchmaking.html

3. Local ClassAd Content

As mentioned above, a set of local variables is added to the ReSS ad. Currently, the variables added are:

- **WantAdRevaluate** = True
- **UpdateSequenceNumber**, set to the seconds since
- **CurMatches**, number of jobs currently matched against the site
- **MaxMatches**, number of maximum jobs we want on this site
- **Requirements** = (CurMatches < MaxMatches)
- **Rank**, a simple calculated rank on how good the site is

The site verification jobs can add classad content as well. The key is then perpended with “Engage”.

4. Site Verification

Site verification is done using two jobs to each site: one to jobmanager-fork and one to the scheduler. The test are run to make sure the site is working form the Engage VO's point of view and is hence a little bit different from the testing OSG is doing. Some of the tests are fatal and some just results in a class ad.

1. Did both jobs run without any problems?
2. Clean $HOME and Globus temporary file locations, and check that we are not using up too much space.
3. Are OSG_DATA, OSG_APP and OSG_WN_TMP defined and writable? Are they writable from worker nodes as well? OSG_APP may or may not be writable, but this gets advertised in the final class ad.
4. Can we find mpirun and/or mpiexec?

The site verification scripts can also define variables for inclusion in the Condor classad. For example, if mpirun and mpiexec is found, the locations will be available in the ad as:

```
EngageMPIExec = "/usr/bin/mpiexec"
EngageMPIRun = "/usr/bin/mpirun"
```