

SA3 Partner Report: FOM (pronounce as Nikhef)

D. H. van Dok J. J. Keijser

EGEE-III INFSO-RI-222667 JRA1/SA3 All Hands Meeting,
Prague, 5–7 November 2008

Outline

Outline

Contents

1 Patch Certification/Rejection	1
2 LSF Gap analysis	3
3 Strategy document on batch system integration	4
4 SCAS client testing	5
5 What's up next?	5

An open question

I want to start by stating a known open question that is—to me—completely unclear, as far as the solution is concerned. This will give you all something to ponder over even if you find the rest of my talk too boring. I will come back to this at the end.

An open question

How do we get the declarations that the user specified in the job description from the user to the batch system?

1 Patch Certification/Rejection

The certification of patches is one activity that we have got involved in. We would be given those patches that relate to batch systems, and this turns out to be an ideal way to get intimate with the particularities of these systems. However, before we could get going we needed a testbed.

Testbed setup

We have a threesome of heavy-duty dual quadcore machines to run a virtual testbed on. We run Centos 5 with Xen 3.1, and we have a fully kickstart-driven installation procedure.

All the test machines are virtual machines, either paravirtualized (CentOS 4, CentOS 5) or fully virtualized (CentOS 3 and exotics). This poses some problem if the use of virtualization itself is the subject of study.

The management software is a home-grown collection of scripts, with kickstart and dnsmasq taking prominent positions. Each machine is tracked and managed by its unique and pre-scribed MAC address, which has a 1-on-1 relationship with the IP address.

Testbed setup

- three dual-quad-core Xeons
- Centos 5 with Xen
- home-grown kickstart-based installation procedure
- pre-defined MAC address is the key
- dnsmasq does all the hard work (DNS, DHCP, PXE)

Condor-as-a-batch-system

Patch rejection

Condor-as-a-batch-system

- *#1522* Started certification April 18 (!), rejected May 28.
- *#1856* Started May 28, certified June 10. Ultimately in production August 6.
- *#2079* Support for Condor 7.0.4. Started on August 15, rejected on September 10.
- *#2514* (ditto) started October 15, rejected October 23
- *#2593* (ditto) but now includes CREAM, will take more time to reject.

Patch *#1522*.

This was for Condor-utils on the LCG-CE. The patch was rejected, as it was dragging on too long, even though only a few minor issues were found.

The next attempt was passed (certified) in a reasonable timeframe. It should really have been rejected, because bugs were found (and fixed) along the way. This is not the way we agree to certify patches anymore.

Then came a patch for supporting condor 7.0. Some issues were found and the patch was therefore rejected. Another one like that, and then the one that also needs certification for CREAM.

Torque client

Torque patches

- *#1703* Assigned to Jan Just: Torque client metapackage on 64 bit worker nodes. Picked up on April 16, certified on June 5.
- *#2517* New version of Torque (2.3.3) and Maui. Involves a vulnerability. Work on this will start after all hands.

Fixes 'possible vulnerability <https://savannah.cern.ch/bugs/?42652>'

2 LSF Gap analysis

LSF Gap Analysis

Task *#7117*: LSF Gap Analysis

- Started June 13, ended June 24.
- Jobmanagers for the LCG-CE: CERN patches to reduce the number of queries to the batch system.
- BLAH scripts: provided by INFN. Scalability is the key.
- Requires `btools` in a source distribution (to be compiled against the specific version of LSF).
- Job priorities: LSF fair shares
- APEL: YAIM config in place

Overall conclusion:

- The state of integration is not bad
- But much is CERN specific, no guarantees or support for other sites

The current state of LSF integration

- Jobmanagers for the LCG-CE. CERN has its own patches for

```
/opt/globus/lib/perl/Globus/GRAM/Helper.pm and  
/opt/globus/setup/globus/lcglfs.in
```

to decrease the number of queries to the batch system. The situation leaves room for improvement, because there are still too many concurrent connections to the CE; this is mostly coming from the VDT side.

- BLAH scripts for CREAM. INFN is writing (and will support) LSF scripts. The earlier experiences were that scalability was not good, so it remains to be seen how this works under high load.
- Information providers. these are CERN made and should work on all sites. The site-specifics should only be in the configuration files. A set of tools, called `btools`, is recommended. This is a source-only distribution, sites will have to compile it for themselves against their specific version of LSF.

- Job priorities. Implemented through LSF fair shares; no change are required in the information providers, other than to the configuration files
- APEL. YAIM configuration script is in place. The general complaint about APEL is that detailed accounting is not possible, so a CPU 'normalization' has to be done to average the worker node performance. This is not a LSF specific issue, however.
- Overall the conclusion of the state of integration is not bad, but the support is worrisome. Ulrich Schwickerath, who is not in SA3 but operating the CERN clusters, has taken a load of integration work, but he can not be asked to generalize it to broader LSF support for other sites.

3 Strategy document on batch system integration

Millstone document

Originally MSA 3.2 was to be on porting *and* batch systems.

The batch system part was split off into a subdocument, which immediately improved both documents.

Strategy for batch systems

- focus on Torque, SGE, Condor
- for LCG-CE and CREAM-CE
- integration points:
 - jobmanagers
 - BLAH scripts
 - information providers
 - APEL (accounting)
- Cookbook on *How to integrate your favourite batch system*
- Integration check list
- more standardized tests

More in the joint session on Thursday.

MSA-3.2 'porting and batch systems' ended up splitting off a subdocument for batch systems.

Adding support for a new batch system involves the following four software integration points:

1. Jobwrappers (for the LCG-CE). For SGE and Condor these jobwrappers scripts are largely finished;
2. BLAHP scripts (for the CREAM-CE). As soon as up-to-date specifications of the CREAM/BLAHP interface is available, work can start on writing SGE and Condor specific BLAHP scripts;

3. Information Provider service (e.g. info-dynamic-scheduler plug-ins);
4. APEL (accounting info).

as well as a full set of documentation on how to set up and integrate a particular batch system with the gLite middleware. The packages that will be produced as a result of the integration effort will be available as separate installation packages of the gLite middleware. Any necessary changes made to the gLite middleware itself will be integrated with the main gLite ETICS repository.

4 SCAS client testing

During some of the somewhat less enthralling moments of the EGEE conference, Oscar and I were frantically working to stress-test the SCAS service.

Using our own testbed, a set of 16 SCAS clients (glexec) were quickly erected on a single box. This proved to be enough requests to bring the service to pump out 20 mappings per second.

SCAS client performance testing

photo: Onno Zweers

A later test with physical worker nodes showed a single service instance peaked out there, but multiple instances (up to four) showed a linear increase in the amount of mappings. Neither the network nor the CPU were anywhere near saturated.

5 What's up next?

Up next:

- patch #2593: Condor and CREAM
- patch #2517: New Torque and Maui
- task #8088: coordination over the convergence of info dynamic providers
- writing the 'cook book' and 'check list'

We've got work on some current patches. An additional task is to oversee the convergence for the info-dynamic-providers for the various batch systems.

Since the Condor patch now includes CREAM, that is one nodetype to start testing with. That's a bit of uncharted territory for us.

Writing the 'cook book' and 'check list', and updating the existing documentation is now becoming our priority.

Questions?

How do we get the declarations that the user specified in the job description from the user to the batch system?

This question is really touching several current problems.

- users can't expect to land their job in the declared environment, for instance with Python 2.5 as the default Python.
- sites cannot benefit from the known and declared behaviour of a job by using more efficient scheduling (e.g. backfilling).
- ...