



HTCondor Week 2015: Implementing an HTCondor service at CERN

Iain Steers, Jérôme Belleman, Ulrich Schwickerath – IT-PES-PS

Outline

The Move

Environment

Grid Pilot

Local Jobs

Conclusion

Why Move?

Several primary reasons:

- Scalability
- Dynamism
- Open-Source and Community

Other reasons as well.

Scalability

- Hard limit on the number of nodes LSF can support. We continue to get closer.
- Even below, LSF based system has become very difficult to manage.

Community

- LSF is proprietary, was owned by Platform Inc. now IBM.
- Most sites have moved or are moving from their systems to HTCondor.
- From experience HTCondor seems to have a brilliant community.

Some Numbers

What we'd like to achieve:

Goals	Concerns with LSF
30 000 to 50 000 nodes	6 500 nodes max
Cluster dynamism	Adding/Removing nodes requires reconfiguration
10 to 100 Hz dispatch rate	Transient dispatch problems
100 Hz query scaling	Slow query/submission response times

Our Compute Environment

CERN is a heavy user of the Openstack project.
Most of our compute environment is now virtual.
Configuration of nodes is done via Puppet.



Configuration and Deployment

Using the HTCondor and ARC CE Puppet modules in the HEP-Puppet Github, plus some of our own.

Node lists for the HTCondor configs are generated via puppetdb queries and hiera.

Works very nicely at the moment.

Why just Grid?

- Less to do, with regard to Kerberos/AFS etc.
- Means early-adopters can help us find problems.
- Can have a PoC running whilst local-work is ongoing.
- Smaller number of jobs compared to local.

Worker Nodes

8-Core flavour VMs with 16GB RAM partitioned to 8 job slots.

Standard WLCG nodes with gLExec running a HTCondor version of MachineJob Features.

MachineJob Features required by the experiments for communicating with their jobs and making decisions about the machine.

Queues

Biggest departure from our current approach, visible from the outside world for Grid jobs.

Following the HTCondor way with a single queue, seems to simplify most things.

Management

The HTCondor Python Bindings have more than stepped up to the plate here.

One of my favourite features/parts of HTCondor, can query anything and everything.

Most of the management of the pool will be done via the Python-bindings. Most of our monitoring uses the classad projections to avoid unnecessary clutter.

Accounting

Written a Python library that uses the classads library. Accountant daemon running that picks up HTCondor jobs as they finish.

Backend agnostic, currently sends jobs data to an accounting db, elasticsearch and our in-house monitoring.

Data also being pumped to HDFS and our analytics cluster for later analysis.

Monitoring



Current Status

What have we achieved so far?

- All Grid items are at least PoC ready.
- Grid Pilot is open to ATLAS and CMS with 96 cores.
- Already had 40,000 ATLAS jobs, CMS starting shortly.

The Future

Local jobs and taking the Grid PoC to production.

Distributed Schedulers

80% of our job-load will be local submission.

Several thousand users all wanting to query schedulers...

Distributed Schedulers

Problem: How do we assign users to a scheduler?

Suggestions from Miron and Todd on this:

- A schedd that answers queries about all schedds.
- Hash job ids to schedds and embed in classad.

One of ours:

- DNS delegated personalized addressing for each user. e.g. `isteers.condor.cern.ch`

We like these ideas, but we'd like to know how others have achieved this?

User Areas and Authorization

Kerberos is needed for local jobs, for things such as access to user areas.

Although not related directly to HTCondor, this is a good opportunity to review our existing solutions.

Problem: Security of tickets and how to renew appropriately.

Miron suggested letting the schedds handle renewal of tickets and passing to worker nodes.

Group Membership

Easy for Grid jobs with *relatively* neat classad expressions.

Not so obvious for local submission with 100s of group.subgroup combinations.

Python classad functions may be our solution here but we have scaling concerns.

Open Issues

Open Issues with no clear (or multiple) solutions:

- Mapping users to schedulers.
- Kerberos tickets and handling.
- Enforcing group membership.

Conclusion

Local jobs and production service by end of year.

Questions?



www.cern.ch