Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Care and Feeding of HTCondor Cluster

Steven Timm

European HTCondor Site Admins Meeting

8 December 2014

Disclaimer

- Some HTCondor configuration and operations questions are more religion than science.
- There are at least three HTCondor religions within Fermilab itself, maybe more.
- You will probably hear speakers before me and after me tell you the exact opposite things to do from what I say.
- What I am showing is one way that works, it is not the only way.

Background:

- I had lead role in the computing farms and later FermiGrid from 2000-2013
  - (Since Nov. 2013 I do exclusively cloud projects).
- FermiGrid started out with 28 cores under HTCondor and now has built up to almost 30000 cores.
  - Charter site in US Open Science Grid, has delivered 10^9 hours
  - Currently versions from 7.6 to 8.3 in production at FNAL
- I will focus mostly on how we maintain HTCondor on the grid clusters because I believe that is closest to what most admins are dealing with now.
- We also have several large batch submission services which use HTCondor and GlideinWMS to submit locally and to the grid and the cloud. That introduces all kind of complexities that don't affect a purely local cluster.

Good Sites Start With Good Policy

- Early Management Directives:
  - Special security enclave for nodes where grid jobs run
  - All major users and clusters must interoperate.
  - Unified unix user mapping across all clusters
  - Keep more than one batch system in production
  - Have users submit to the cluster the same way grid users do—
    - That way they know the commands if they need to branch out.
- Your site is not the same as Fermilab
- But nevertheless you should think about these things at setup
  - So you don't run into a security fault that is hard to change.

Authentication/Authorization

- How do the condor daemons authenticate to each other?
  - Many methods available (KERBEROS, POOL_PASSWORD, GSI, etc)
  - At Fermilab we use GSI
    - Make proxy of the head node host cert, share it with all machines.
    - To do this you need a list of the nodes that should be in your condor pool.
- How do users get authorized?
  - All jobs could run as “nobody” but don’t do it.
  - Create set of unix users—one user per group/VO
  - Use UID_DOMAIN and FILESYSTEM_DOMAIN to have unified set of uid/gid across whole cluster
    - i.e. user is identified as nova@fnal.gov everywhere

Authentication/Authorization 2

- Don’t share home directories on worker nodes with interactive nodes.
  - Risk of rogue jobs dumping auto-executables into home directories.
  - In modern condor setup the grid users don’t need a shared home directory on the worker nodes at all.
  - Historically OSG has used “gLexec” to map each user DN to a different account.. This is gradually going away.
  - Don’t give the accounts a shell
- A few experimental “admins” have non-privileged login
- Nss_Db is a very good (but badly documented) way to distribute passwd/group to large number of workers
  - File is stored locally on each worker node (pushed via puppet)
  - Uses Berkeley DB to search the map

FILESYSTEMS, MEMORY, ETC.

- Disk Usage
  - At our request HTCondor added a feature to have each job execute in its own disk directory/partition:
  - `SLOTn_EXECUTE=/path/to/slotn_scratch`
  - This is a lifesaver—keeps one job from sucking up all disk space and taking the node down with it.
  - Disk usage reports in job classads tend to be inaccurate and also lag 10-20 minutes behind real time.
- Memory usage:
  - Memory reported in `condor_q` output is `ImageSize`
  - Much better to constrain on `ResidentSetSize`
  - We kill high memory jobs with a `SYSTEM_PERIODIC_REMOVE` expression in the `schedd`
  - Our machines have 2GB per job slot—we kill jobs at 10GB `ResidentSetSize`.

Designing your condor masters

- I recommend two collector/negotiator machines using Condor High Availabilty as documented in the manual.
  - Condor_negotiator runs on one or the other, collector runs on both
  - Replication daemon keeps the negotiator data (Accountantnew.log) fresh between the 2 machines.
  - COLLECTOR_HOST=fnpccm2.fnal.gov,fnpccm1.fnal.gov
  - Note that all condor_advertise will attempt to go to both machines in order..so if first one in list is down there can be timeouts.
- Clear network bandwidth is a must
  - Use TCP connections wherever you can.
- We run our collector/negotiator on virtual machines
  - KVM, 2 cores, 8GB RAM

Designing your condor_schedd

- Fermi General Purpose Grid has 9300 slots, 5 schedd's
  - That's overkill, three would be enough.
  - The extra two are for functional separation for opportunistic jobs
- Suggest having 4GB of RAM per 1000 running jobs.
  - Ours have 16-20GB of RAM, 4-6 cores. (they do grid functions)
- Fast cores are more important than lots of cores.
  - Lots of schedd functions still single-threaded
- Fast disk important too – lots of small files
  - We don't have SSD's but I have always wanted them.
  - In place of that we have a RAID array of 15K SAS disks.
  - Whatever you do, don't put the schedd spool areas, or the areas where user logs are written, on NFS.
    - Explicitly *not supported* by HTCondor
    - Recipe for horrible system crashes

Keeping track of your nodes

- We maintain a flat file of all nodes that should be there
- Then have a script /usr/local/bin/condorcheck.sh to compare that to the output of condor_status –master
- Nowadays we could use puppet exported resources to make the list, but we don’t, at least not yet.

Upgrades

- Don’t run any odd-numbered development version (8.1,8.3)
  - No matter what Brian B. says.
- Don’t run any .0 version (8.0.0, 8.2.0, etc.)
  - The .1 upgrade always comes up very quickly.
- Test at scale before upgrading your production.
- Always a couple new “features” or things that work differently.
- RPM Packages sometimes change location of config files between versions.

Configuration Management / Packaging

- For best results leave the config files that come with the RPM untouched. (/etc/condor/config.d) and the /etc/init.d/condor script
  - Can customize /etc/sysconfig/condor, that doesn't get clobbered
- Puppet could let you configure each node differently if you want. Don't do it! Keep base condor_config same.
  - Use nodename-based conditionals
  - AFSNODES=("node1","node2", "node3")
- Special nodes such as schedd put in a special file
- Document in your extra files what you changed and why
  - Sometimes a setting becomes no longer necessary, should check over time.

Downtimes/Maintenance

- /etc/init.d/condor has a function called “peaceful-off”
- Condor_startd will let all running jobs finish and then exit.
- Useful if you want to idle everything before a downtime.
- Versions 8.0 and greater it is possible to yum update worker nodes in place and new jobs start with the new version of the condor_startd and old jobs don’t die.
- Not so with condor_schedd or collector/negotiator, have to restart the services to make things upgrade clean.
- Also can set MAX_JOBS_SUBMITTED=0 on the schedd
  - All jobs currently running will finish (and can be monitored while they finish) but no new ones can be submitted.

Downtimes/Maintenance 2

- For a full idle we do both:
  - First set MAX_JOBS_SUBMITTED=0
  - Then put all the workers into drain so no new jobs start
  - Jobs that haven't started remain in the queue and start after downtime.
  - Can keep jobs from starting after the downtime until you want them to start by MAX_JOBS_RUNNING=0

To Pre-empt or not to Pre-empt

- Corner cases of very long-running or hung jobs happen frequently.
- Lots of jobs at Fermilab write stuff to database while running, auto-restart is not always practical.
- In General Purpose Grid we have a tiered system:
- Each VO gets a quota of non-preemptable jobs and then can use opportunistic cycles unlimited above that.
- Also make heavy use of priorities to penalize non-Fermilab opportunistic users (10^6->10^9 priority factors).
- This system is scheduled to change pretty soon, see my talk tomorrow.
- CMS Tier 1 @FNAL doesn't pre-empt but uses even more extreme priority factors (10^26).
- Make sure you have buy-in from high management and that they know what you are doing.
- In our setup we use MaxJobRetirementTime to give pre-empted jobs 24 hours to finish. Less than 1% of jobs get pre-empted.

Filling pattern

- HTCondor allows for vertical fill (fill up one node first, then the next, and so forth, or horizontal fill (distribute jobs equally among all nodes.
- FermiGrid has used horizontal fill for most of time we've been up and running.
- NEGOTIATOR_POST_JOB_RANK = (RemoteOwner =?= UNDEFINED) * SlotID
- This will fill slot1 first on all nodes, then slot2, etc.

The customized settings that help us do what we do

- PREEMPTION_REQUIREMENTS
- SYSTEM_PERIODIC_REMOVE = (NumJobStarts > 10) || (ResidentSetSize>=10000000) || (JobRunCount>=1 && JobStatus==1 && ResidentSetSize>=10000000)
- MAX_JOBS_RUNNING (needs to be more than N of slots)
- REQUEST_CLAIM_TIMEOUT=600
- CLAIM_WORKLIFE=3600
- JobLeaseDuration=3600
- MAX_JOBS_SUBMITTED
- GSI_AUTHZ_CONF=/dev/null
- EVENT_LOG, EVENT_LOG_MAX_ROTATIONS
- MAX_HISTORY_ROTATIONS

Log files

- EVENT_LOG—all userlog information in all jobs together in one log file. HUGE time saver. Enable it.
  - (best new feature in HTCondor in 5 years)
- Increase all log sizes and debug levels, don't be stingy
  - MAX_SCHEDD_LOG=500000000 for example
  - SCHEDD_DEBUG=D_PID,D_COMMAND,D_SECURITY
- We ran with D_FULLDEBUG on pretty much everything until a couple of years ago.

Getting Features from HTCondor team

- Fermilab requested and got:
  - Kerberos Authentication
  - X.509 Authentication
  - Separate execution disk partitions
  - Partitionable slots
  - Integrated support for gLexec
  - VOMS support / support for external LCMAPS callout
  - Cloud support (OpenNebula, OpenStack, Google)
  - Extensions to quota system
  - Many, many more.
- Come to Madison, Wisconsin for HTCondor week
- Capture karaoke performances on video
- Negotiate appropriately.

Conclusions:

- An HTCondor pool is a living and evolving system
- Keep track of the configuration decisions you made, and why.
- Review them from time to time.
- No substitute for being on the system and watching its performance.