What’s new in HTCondor?
What’s coming?

HTCondor Week 2018
Madison, WI -- May 22, 2018

Todd Tannenbaum
Center for High Throughput Computing
Department of Computer Sciences
University of Wisconsin-Madison
Release Timeline

› Stable Series
  • HTCondor v8.6.x - introduced Jan 2017
    Currently at v8.6.11

› Development Series (*should be 'new features' series*)
  • HTCondor v8.7.x
    Currently at v8.7.8

› New v8.8 stable series coming this summer

› Detailed Version History in the Manual
Version 8.7.8

Release Notes:

- HTCondor version 8.7.8 released on May 10, 2018.

New Features:

- `condor_annex` may now be setup in multiple regions simultaneously. Use the `-aws-region` flag with `-setup` to add new regions. Use the `-aws-region` flag with other `condor_annex` commands to choose which region to operate in. You may change the default region by setting `ANNEX_DEFAULT_AWS_REGION`. ([Ticket #6632](#)).
- Added default AMIs for all four US regions to simplify using `condor_annex` in those regions. ([Ticket #6633](#)).
- HTCondor will no longer mangle `CUDA_VISIBLE_DEVICES` or `GPU_DEVICE_ORDINAL` if those environment variables are set when it starts up. As a result, HTCondor will report GPU usage with the original device index (rather than starting over at 0). ([Ticket #6584](#)).
- When reporting `GPUsUsage`, HTCondor now also reports `GPUsMemoryUsage`. This is like `MemoryUsage`, except it is the peak amount of GPU memory used by the job. This feature only works for nVidia GPUs. ([Ticket #6544](#)).
- Improved error messages when delegation of an X.509 proxy fails. ([Ticket #6575](#)).
- `condor_q` will no longer limit the width of the output to 80 columns when it outputs to a file or pipe. ([Ticket #6643](#)).
- Submission of jobs via the Python bindings Submit class will now attempt to put all jobs submitted in a single transaction under the same ClusterId. ([Ticket #6649](#)).
- Added support for `condor_schedd` option in the Python bindings. ([Ticket #6619](#)).
- Eliminated SOAP support. ([Ticket #6648](#)).

Bugs Fixed:

- Fixed a problem where, when starting enough `condor_annex` instances simultaneously, some (approximately 1 in 100) instances would neither join the pool nor terminate themselves. ([Ticket #6638](#)).
- When using HAD to the point of running out of resources, drivers terminated idle jobs which could
Enhancements in HTCondor v8.4

› Scalability and stability
  • Goal: 200k slots in one pool, 10 schedds managing 400k jobs
› Introduced Docker Job Universe
› IPv6 support
› Tool improvements, esp condor_submit
› Encrypted Job Execute Directory
› Periodic application-layer checkpoint support in Vanilla Universe
› Submit requirements
› New RPM / DEB packaging
› Systemd / SELinux compatibility
Enhancements in HTCondor v8.6

› Enabled and configured by default: use single TCP port, cgroups, mixed IPv6 + IPv4, kernel tuning
› Made some common tasks easier
› Schedd Job Transforms
› Docker Universe enhancements: usage updates, volume mounts, conditionally drop capabilities
› Singularity Support
HTCondor Singularity Integration

› What is Singularity?
  Like Docker but...
  • No root owned daemon process, just a setuid
  • No setuid required (post RHEL7)
  • Easy access to host resources incl GPU, network, file systems

› HTCondor allows admin to define a policy (with access to job and machine attributes) to control
  • Singularity image to use
  • Volume (bind) mounts
  • Location where HTCondor transfers files
Whats cooking in the kitchen for HTCondor v8.7 and beyond
Docker Job Enhancements

- Docker jobs get usage updates (i.e. network usage) reported in job classad
- Admin can add additional volumes
  - That all docker universe jobs get
- Condor Chirp support
- Conditionally drop capabilities
- Support for condor_ssh_to_job
Two uses for `condor_ssh_to_job`
- Interactive session alongside a batch job
  - Debugging job, monitoring job
- Interactive session alone (no batch job)
  - Jupyter notebooks, schedule shell access
  - p.s. Jupyter Hub `batchspawner` supports HTCondor

Can tell the schedd to run a specified job immediately! Interactive sessions, test jobs
- No waiting for negotiation, scheduling
  - `condor I_am_impatient_and_waiting`?
HTCondor Python API

› HTCondor V8.7 eliminated web-service SOAP API, long live Python!
› Started with Python 2, now also support Python 3
› Packged into PyPI repository (Linux only)
  
  pip install htcondor

› Several API improvements and simplifications, e.g. much easier to submit jobs
  • Can use condor_submit JDL directly, including queue for each
  • Starting to think about higher-level job submit abstractions

› MS Windows now supported
HTCondor "Annex"

› Instantiate an HTCondor Annex to dynamically add additional execute slots for jobs submitted at your site
  • Get status on an Annex
  • Control which jobs (or users, or groups) can use an Annex

› Want to launch an Annex on
  • Clouds
    • Via cloud API (or Kubernetes?)
  • HPC Centers / Supercomputers
    • Via edge service (HTCondor-CE)
Grid Universe

› Reliable, durable submission of a job to a remote scheduler
› Popular way to send pilot jobs (used by glideinWMS), key component of HTCondor-CE
› Supports many “back end” types:
  • HTCondor
  • PBS
  • LSF
  • Grid Engine
  • Google Compute Engine
  • Amazon AWS
  • OpenStack
  • Cream
  • NorduGrid ARC
  • BOINC
  • Globus: GT2, GT5
  • UNICORE
Added Grid Universe support for Azure.

**SLURM, Cobalt**

- Speak to Microsoft Azure
- Speak native SLURM protocol
- Speak to Cobalt Scheduler
  - Argonne Leadership Computing Facilities

**Also HTCondor-CE "native" package**

- HTCondor-CE started as an OSG package
- IN2P3 wanted HTCondor-CE without all the OSG dependencies….
- Now HTCondor-CE available stand-alone in HTCondor repositories
CPU cores!

FNAL HEPCloud NOvA Run
(via Annex at NERSC)
http://hepcloud.fnal.gov/
No internet access to HPC edge service?
File-based Job Submission

Schedd A

JobXXX

request
input
input
input

status.1
status.2
status.3
output
output
output

Schedd B
condor_annex tool

› Start virtual machines as HTCondor execute nodes in public clouds that join your pool
› Leverage efficient AWS APIs such as Auto Scaling Groups and Spot Fleets
   • Other clouds (Google, Microsoft) coming
› Secure mechanism for cloud instances to join the HTCondor pool at home institution
› Shut down idle instances, detect instances that fail to start HTCondor
› Implement a fail-safe in the form of a lease to make sure the pool does eventually shut itself off.
Work on Noisy Neighbor Challenges

- Already use cgroups to manage CPU, Memory… what about CPU L3 cache? Memory bus bandwidth?
- Working with CERN OpenLab and Intel on leveraging Intel Resource Directory Technology (RDT) in HTCondor
  - Monitor utilization
  - Assign shares
Multi-core challenges. Low priority user submits millions of 1-core jobs; subsequently high priority user submit a 4-core job. What to do?

• Option 1: Draining
  • *HTCondor can now backfill draining nodes with pre-emptible jobs*

• Option 2: Preemption
  • *HTCondor can now preempt multiple jobs, combine their resources, and assign to a higher priority job*
  • Initial implementation to gain experience at BNL; however we are still working on a cleaner model
Compute node management enhancements, cont.

- GPU Devices
- HTCondor can detect GPU devices and schedule GPU jobs

- New in v8.7:
  - Monitor/report job GPU processor utilization
  - Monitor/report job GPU memory utilization

- Future work: simultaneously run multiple jobs on one GPU device
  - Volta hardware-assisted Multi-Process Service (MPS)
Security: From identity certs to authorization tokens

- HTCondor has long supported GSI certs
- Then added Kerberos/AFS tokens for CERN, DESY
- Now adding standardized token support
  - SciTokens (http://scitokens.org)
  - OAuth 2.0 Workflow → Box, Google Drive, Github, …
Planning for US Federal Information Processing Standard (FIPS) Compliance

- Can do better than MD-5, 3DES, Blowfish
- AES has hardware support in most Intel CPUs

May motivate us to drop UDP communications in HTCondor

- Almost all communication in HTCondor is now asynchronous TCP anyway
- Anyone care if UDP support disappears?
Scalability Enhancements

› *Late materialization* of jobs in the schedd to enable submission of very large sets of jobs
  • More jobs materialized once number of idle jobs drops below a threshold (like DAGMan throttling)

› Central manager now manages queries
  • Queries (i.e., *condor_status* calls) are queued; priority is given to operational queries

› More performance metrics (e.g., in collector, DAGMan)
Job input files normally transferred to execute node over CEDAR, now can be sent over HTTP

- Enable caching (reverse and forward proxies), redirects
- More info from Carl Vuosalo's talk: https://tinyurl.com/yd6mya96

Proposed a project to manage data leases (size and time lease) for job output across HTCondor, Rucio, XRootD
Thinking about how to add "provisioning nodes" into a DAGMan workflow
  • Provision an annex, run work, shutdown annex

Working with Toil team so Toil workflow engine can submit jobs into HTCondor
Thank you!