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

HTCondor Week 2021

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

› Stable Channel *(bug fixes only)*
  • Previously known as "stable series"
  • HTCondor v9.0.x – first introduced Apr 2021
    (Currently at v9.0.1)

› Current Channel *(has new features)*
  • Previously known as "development series"
  • HTCondor v9.1.x (Currently at v9.1.0)

› Big news is we now have v9.0!
  • Contains 71 documented enhancements
    introduced in the past 14 v8.9.x releases.

› Detailed Version History in the Manual
Typical Installation procedure before HTCondor v9.0… a bit daunting…

› Figure out what OS and distribution using
› Add proper repositories and signing keys with system packaging commands (yum / apt commands…)
› Install binaries
› *If you want a security beyond host-based authentication*
  • Read Security section of the manual
  • Configure security settings
› Figure out if your system is using systemd
  • Hint: likely yes if on bare metal, likely no if in a container…
› Start up HTCondor services, and at reboot
When you get HTCondor, start with the central manager, then add the submit machine(s), and then add the execute machine(s).

Central Manager

```bash
sudo curl -fsSL https://get.htcondor.org | GET HTCONDOR_PASSWORD=$htcondor_password
```

Submit

```bash
sudo curl -fsSL https://get.htcondor.org | GET HTCONDOR_PASSWORD=$htcondor_password
```

Execute

```bash
sudo curl -fsSL https://get.htcondor.org | GET HTCONDOR_PASSWORD=$htcondor_password
```

At this point, users logged in on the submit machine should be able to see execute machines in the pool (using `condor_status`), submit jobs (using `condor_submit`), and see them run (using `condor_q`).

Creating a Multi-Machine Pool using Windows or Containers

If you are creating a multi-machine HTCondor pool on Windows computers or using containerization, please see the “Setting Up a Whole Pool” section of the relevant installation guide:
You may have questions…

› "I don't want to pipe something off the internet into a root shell on my production machines…"
  • You don't have to… Also, get_htcondor has a non-root mode that just displays the commands it would use to do the install (without doing anything).

› "Is this configuration secure by default?"
  • Yes. HTCondor services will authenticate across servers via tokens and use encryption on the network.

› "How about upgrading an existing pool to v9.0?"
  • Tune in to Todd Miller's talk "Upgrading to HTCondor 9.0" Weds Morning @ 9am Chicago
So what's new in v9.0.x?
New functionality for end-users

- Transfer job data to/from web servers, Box.com, Amazon S3, Google Drive, MS OneDrive via file transfer plugins now supporting uploads, authentication, and improved error handling
- Improved support for GPUs, including GPU utilization info and support for using GPUs in containers
- New condor_watch_q tool that efficiently provides live job status updates
- New tools and mechanisms to support jobs that checkpoint
- Fixed interactive submission of Docker jobs
- Many improvements to the Python API, including Python 3 support and new bindings for DAGMan and chirp
- DAGMan now provides a method for inline jobs to share submit descriptions
- You may now change some DAGMan throttles while the DAG is running
- You can assign priorities to DAGMan (scheduler universe) jobs
- And more…
New functionality for end-users

- Transfer job data to/from web servers, Box.com, Amazon S3, Google Drive, MS OneDrive via file transfer plugins now supporting uploads, authentication, and improved error handling
- Improved support for GPUs, including GPU utilization info and support for using GPUs in containers
- New condor_watch_q tool that efficiently provides live job status updates
- New tools and mechanisms to support jobs that checkpoint
- Fixed interactive submission of Docker jobs
- Many improvements to the Python API, including Python 3 support and new bindings for DAGMan and chirp
- DAGMan now provides a method for inline jobs to share submit descriptions
- You may now change some DAGMan throttles while the DAG is running
- You can assign priorities to DAGMan (scheduler universe) jobs
- And more…
New functionality for sys admins

- Fundamental security model changes:
  - Secure by default
  - Authorization via Tokens: IDTOKENS, SciTokens
  - OAuth2 workflow to interoperate with other services

- New packaging
  - Native packaging has releases, release candidates, and daily builds
  - Official containers on Docker Hub designed to work well with k8s
    - Conda

- Push job information into Elastic Search
- More powerful classad transform language for HTCondor-CE, schedd
- Dataflow mode for jobs
- Container support improvements
- Cgroup limit improvements
- Cap the number of cores allocated to individual users
- New platforms (e.g. EL8, Ubuntu 20)
- And more…
New functionality for sys admins

- Fundamental security model changes:
  - Secure by default
  - Authorization via Tokens: IDTOKENS, SciTokens
  - Oauth2 workflow to interoperate

- New packaging
  - Native packaging has releases, release candidates, daily builds
  - Official containers on Docker Hub designed to work well with k8s
  - Conda
  - Push job information into Elastic Search

- More powerful classad transform language for HTCondor-CE, schedd

- Dataflow mode for jobs

- Container support improvements

- Cgroup limit improvements

- Cap the number of cores allocated to individual users

- New platforms (e.g. EL8, Ubuntu 20)

- And more…
What's coming to v9.1.x and beyond?
Organize Jobs into *Job Sets*

› Today job "clusters" mostly behave as expected
  • Can remove all jobs in a cluster
  • Can edit all jobs in a cluster

› But some operations are missing
  • Append jobs to a set (in a subsequent submission)
  • Move an entire set of jobs from one schedd to another
  • Job set *aggregates* (for use in polices?)
Job Sets, cont

- Users want to think about a set of jobs as it relates to their mental model (and NOT based upon when they submit). For instance:
  - Set of jobs analyzing genome 52
  - Set of jobs doing analysis on image captures from date xxx

- Experimenting with new command line user interface
  - `condor <noun> <verb>`
  - "condor set submit", "condor set status", …
  - "condor job submit", "condor job status", …
  - Legacy tools (condor_q, condor_submit, …) not going anywhere…
HTCondor Annex

› Instantiate an HTCondor Annex to dynamically add additional execute slots into your HTCondor environment

› Continue work on mechanisms to enable end-users to provision an Annex on
  • Local Clusters (e.g. campus Slurm HPC clusters)
  • Clouds
  • HPC Centers / Supercomputers

› Run a job set as a campaign at an HPC site
No internet access to/from HPC nodes? File-based communication between execute nodes

JobXXX

request
input
input
input
output
output
output
status.1
status.2
status.3

condor_starter

condor_starter'

Read more about our current approach at http://tiny.cc/f158cz
HTCondor has long been able to detect GPU devices and schedule GPU jobs (CUDA/OpenCL).

More recently also:

- Monitor/report job GPU processor utilization
- Monitor/report job GPU memory utilization

In the works: concurrent jobs on one GPU device. Questions:

- Specify GPU memory for scheduling?
- Limit to jobs from the same user? Same job set?
- NVIDIA Multi-Instance GPU (MIG) for partitioning?
- We want to hear from you!
HTCondor has long been able to detect GPU devices and schedule GPU jobs (CUDA/OpenCL).

More recently also:

- Monitor/report job GPU processor utilization
- Monitor/report job GPU memory utilization

In the works:

- Concurrent jobs on one GPU device

Questions:

- Specify GPU memory for scheduling?
- Limit to jobs from the same user? Same job set?
- NVIDIA Multi-Instance GPU (MIG) for partitioning?

We want to hear from you!

Improve GPU Scheduling

19
Unified Container Runtime Support

› From "Docker Universe" to "Container Universe" … just jobs with a container image specified regardless of runtime
  • Allow jobs to specify specific containers, allow admins to specify default containers

› Podman support

› Private images from docker hub compliant repositories
  • Cache images locally, but do not reuse private images across users
HTCondor-CE ability to provision resources from a k8s cluster.

Officially support easy "one command" installation of an HTCondor pool onto k8s.

- Note: we have an initial prototype of this work already available
Yet More Security Work…

- Federal Information Processing Standard (FIPS) compliance for mainstream HTCSS product
- Continue work on documentation improvements
- Improve some rough edges: allow an IDTOKEN alone to work well with remote administrative tools like condor_off, condor_drain, condor_reconfig
- Implement plan to replace GSI and Grid Community Toolkit functionality. See timeline at: http://wiki.htcondor.org/index.cgi/wiki?p=PlanToReplaceGridCommunityToolkit
  - Note: HTCSS v9.1.0 already released with ARC-CE REST interface support
Yet More Security Work:

- Federal Information Processing Standard (FIPS) compliance for mainstream HTCSS product
- Continue work on documentation improvements
- Improve some rough edges: allow an IDTOKEN alone to work well with remote administrative tools like condor_off, condor_drain, condor_reconfig
- Implement plan to replace GSI and Grid Community Toolkit functionality. See timeline at:
- Note: HTCSS v9.1.0 already released with ARC-CE REST interface support

Questions? See Brian's talk this afternoon, and join in today's Town Hall Discussion!
Thank You!

Follow us on Twitter!
https://twitter.com/HTCondor

This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATH Project.