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

HTCondor / ARC CE
Workshop
Barcelona 2016

Todd Tannenbaum
Center for High Throughput Computing
Department of Computer Sciences
University of Wisconsin-Madison
Challenge Areas

“...we have identified six key challenge areas that we believe will drive HTC technologies innovation in the next five years.”

- Evolving resource acquisition models
- Hardware complexity
- Widely disparate use cases
- Data intensive computing
- Black-box applications
- Scalability
Release Timeline

› Stable Series
  • HTCondor v8.4.x - introduced Aug 2015
      (Currently at v8.4.4)

› Development Series
  • HTCondor v8.5.3 frozen, in beta test, release to
      web later this month.

› HTCondor v8.6.0 expected summer 2016.
Some enhancements for HTCondor v8.4

› Scalability and stability
  • Goal: 200k slots in one pool, 10 scheddss managing 400k jobs
  • Resolved developer tickets: 240 bug fix issues (v8.2.x tickets), 234 enhancement issues (v8.3 tickets)

› Docker Job Universe

› Tool improvements, esp condor_submit

› IPv6 mixed mode

› Encrypted Job Execute Directory

› Periodic application-layer checkpoint support in Vanilla Universe

› Submit requirements

› New packaging
Scalability Enhancement Examples
Elimination of File Locking on Job and Schedd Event Logs

This lock is no match for the power of POSIX file append semantics!

- File lock requests on Linux are not scheduled
- Schedd observed blocked for minutes!
Condor_shadow resources

A condor_shadow spawned for each running job

Upon spawn, shadow authenticates to schedd, startd (on execute host)
This authentication uses CPU, Memory

Solution:

Shadow Diet!!
Eliminate Authentication!
Authentication Speedups

- FS (file system) and GSI authentication are now performed asynchronously
  - So now a Condor daemon can perform many authentications in parallel
  - CMS pool went from 200 execute nodes (glideins) per collector to 2000
- Can cache mapping of GSI certificate name to user name
  - Mapping can be heavyweight, esp if HTCondor has to contact an external service (LCMAPS…)
  - Knob name is `GSS_ASSIST_GRIDMAP_CACHE_EXPIRATION`
Faster assignment of resources from central manager to schedd

Negotiator can ask the schedd for more than one resource request per network round trip.

\[ \text{NEGOTIATOR\_RESOURCE\_REQUEST\_LIST\_SIZE} = 20 \]
Impact of multiple resource requests
Negotiation times for 1000 slot pool

# of job autoclusters

- 8.2.8 LAN
- 8.3.5 LAN 20reqs
- 8.3.5 LAN 100reqs
- 8.2.8 WAN
- 8.3.5 WAN 20reqs
- 8.3.5 WAN 100reqs
ClassAd Projection Improvements

- Less CPU required to send big projections of ClassAds

"ClassAds. This is the weapon of sysadmin. Not as clumsy or random as a grep or regex. A more elegant weapon for a more civilized age..."
Eliminate CCB service pauses
Query Responsiveness

› Improvement: Collector will not fork for queries to small tables

• Load Collector with 100k machine ads
• Before change: ~4.5 queries/second
• After change: ~24.4 queries/second

› Improvement: Schedd condor_q quantum adjusted (to 100ms)

• Load schedd with 100k jobs ads, 40Hz job throughput
• Before change: ~135 seconds per condor_q
• After change: ~22 seconds per condor_q
Max parallel Running Jobs single HTCondor Pool in latest series

Max Parallel Running Jobs

HTCondor Version

7.0.1  7.1.3  7.3.1  7.5.5  8.3.0  8.3.1  8.3.2

2,000  10,000  20,000  90,000  95,000  150,000  200,000
HTCondor cgroup support now manages swap space in addition to CPU/MEM.

- [Also a lot of progress on "Lark" project to manage network resources]
  - request_network_bandwidth

New job universe to support Containers

(Black Box Applications)
Docker Universe Job
Is still a job

› Docker containers have the job-nature
  • `condor_submit`
  • `condor_rm`
  • `condor_hold`
  • Write entries to the job event log(s)
  • `condor_dagman` works with them
  • Policy expressions work.
  • Matchmaking works
  • User prio / job prio / group quotas all work
  • Stdin, stdout, stderr work
  • Etc. etc. etc.*
Many condor_submit improvements

You submit your jobs with *that* script??!? You’re braver than I thought!
More ways to Queue 'foreach'

Queue <N> <var> in (<item-list>)
Queue <N> <var> matching (<glob-list>)
Queue <N> <vars> from <filename>
Queue <N> <vars> from <script> |

› Iterate <items>, creating <N> jobs for each item
› In/from/matching keywords control how we get <items>
› There's more. See the manual for details.
Example: Queue in

Args = $(Item)
Queue 2 in ( alpha, beta delta gamma )

- Produces 8 jobs (2 for each item)
- It unrolls to this submit file:

```
Item=alpha
Step=0
Queue
Step=1
Queue
Item=beta
Step=0
Queue
...
```
Queue matching files

Queue 3 Item matching (*.dat, m*)

- Produces 3 jobs for each file that matches *.dat or m* (or both)
- $(Item)$ holds each filename in turn
Queue from

Queue from <filename>
• Read <filename> and treat lines as items

Queue from <script> |
• Execute <script> and treat output lines as items
**Condor_q new arguments**

- **-limit <num>**
  - Show at most <num> records

- **-totals**
  - Show only totals

- **-dag <dag-id>**
  - Show all jobs in the dag

- **-autocluster -long**
  - Group and count jobs that have same requirements
  - ...perfect for provisioning systems
Tool improvement questions?
New in 8.4 is support for “mixed mode,” using IPv4 and IPv6 simultaneously.

A mixed-mode pool’s central manager and submit nodes must each be reachable on both IPv4 and IPv6.

Execute nodes and (other) tool-hosting machines may be IPv4, IPv6, or both.

ENABLE_IPV4 = TRUE
ENABLE_IPV6 = TRUE
IPv6 questions?
Jobs can request (or admins can require) that their scratch directory be encrypted in realtime

- /tmp and /var/tmp output also encrypted
- Put `encrypt_execute_directory=True` in job submit file (or `condor_config`)

Only the `condor_starter` and job processes can see the cleartext

- Even a root ssh login / cron job will not see the cleartext
- Batch, interactive, and `condor_ssh_to_job` works
Periodic Application-Level Checkpointing in the Vanilla Universe

- Experimental feature!
- If requested, HTCondor periodically sends the job its checkpoint signal and waits for the application to exit.
- If it exits with code 0, HTCondor considers the checkpoint successful and does file transfer, and re-executes the application.
- Otherwise, the job is requeued.
Submit Requirements

› Allow administrator to decide which jobs enter the queue via a SUBMIT_REQUIREMENTS constraint
› Rejection (error) message may be customized
› In v8.5.1 can also set attributes as immutable
DAGMan changes (cont)

- Node status file:
  - Format is now ClassAds
  - More info (retry number, procs queued and held for each node)
  - Fixed bug: final DAG status not always recorded correctly
  - ALWAYS-UPDATE option
  - Now works on Windows

Good, good! Everything is proceeding as DAGMan has foreseen!
DAGMan changes (cont)

- **dagman.out file:**
  - Node job hold reason in `dagman.out`
  - DAG_Status in `dagman.out`
- `-DoRecovery` command-line option
- Stricter checking of SPLICE syntax
- No (unused) command socket
- Stork no longer supported
HTCondor RPM Packaging

More Standard Packaging

• Matches OSG and Fedora package layout
• Built with rpmbuild
• Source RPM is released
  • Can rebuild directly from the source RPM
  • Build requirements are enforced by rpmbuild
• Partitioned into several binary RPMs
  • Pick and choose what you need
# HTCondor Binary RPM Packages

<table>
<thead>
<tr>
<th>RPM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condor</td>
<td>Base package</td>
</tr>
<tr>
<td>condor-all</td>
<td>Includes all the packages in a typical installation</td>
</tr>
<tr>
<td>condor-bosco</td>
<td>BOSCO – Manage jobs on remote clusters</td>
</tr>
<tr>
<td>condor-classads</td>
<td>HTCondor classified advertisement language</td>
</tr>
<tr>
<td>condor-classads-devel</td>
<td>Development support for classads</td>
</tr>
<tr>
<td>condor-debuginfo</td>
<td>Symbols for libraries and binaries</td>
</tr>
<tr>
<td>condor-externals</td>
<td>External programs and scripts</td>
</tr>
<tr>
<td>condor-externals-libs</td>
<td>External libraries</td>
</tr>
<tr>
<td>condor-kbddd</td>
<td>HTCondor Keyboard Daemon</td>
</tr>
<tr>
<td>condor-procd</td>
<td>HTCondor Process Tracking Daemon</td>
</tr>
<tr>
<td>condor-python</td>
<td>Python Bindings for HTCondor</td>
</tr>
<tr>
<td>condor-static-shadow</td>
<td>Static Shadow (Use 32-bit shadow on 64-bit system)</td>
</tr>
<tr>
<td>condor-std-universe</td>
<td>Standard Universe Support</td>
</tr>
<tr>
<td>condor-vm-gahp</td>
<td>VM Universe Support</td>
</tr>
</tbody>
</table>
More Standard Packaging

• Matches debian package layout
• Built with pbuilder
• Source package is released

<table>
<thead>
<tr>
<th>deb</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condor</td>
<td>Base Package</td>
</tr>
<tr>
<td>condor-dbgs</td>
<td>Symbols for libraries and programs</td>
</tr>
<tr>
<td>condor-dev</td>
<td>Development files for HTCondor</td>
</tr>
<tr>
<td>condor-doc</td>
<td>HTCondor documentation</td>
</tr>
<tr>
<td>libclassad-dev</td>
<td>Development files for Classads</td>
</tr>
<tr>
<td>libclassad7</td>
<td>Classad runtime libraries</td>
</tr>
</tbody>
</table>
Statistics

- Time spent on DNS Lookups
  - Special counter for ‘slow’ lookups
- Counter for ResourceRequestsSent
- Per-user file transfer stats in Submitter ads
- New knob acts a whitelist for Statistics publication to the Collector
What to do with all these statistics?

› Aggregate and send them to Ganglia!
  • `condor_gangliadiad` introduced in v8.2

› In addition (or instead) of sending to Ganglia, aggregate and make available in JSON format over HTTP

› View some basic historical usage out-of-the-box by pointing web browser at central manager (modern CondorView)...

› ….Or upload JSON to influxdb, couchdb, …
Enabled by default and/or easier to configure

- Kernel tuning, cgroups enabled
- Enforce slot sizes
  - use policy: preempt_if_cpus_exceeded
  - use policy: hold_if_cpus_exceeded
  - use policy: preempt_if_memory_exceeded
  - use policy: hold_if_memory_exceeded
- Shared Port
- IPv6
New condor_q default output

- Only show jobs owned by the user
- Batched output (-batch, -nobatch)
- Proposed new default output of condor_q will show summary of current users jobs.

```
-- Submitter: adam     Schedd: submit-3.chtc.wisc.edu
OWNER    IDLE  RUNNING  HELD  SUBMITTED   DESCRIPTION   JOBIDs
adam     -      1       -    3/22 07:20  DAG: 221546  230864.0
         -      -       1    3/23 08:57  AtlasAnlysis 263203.0
         -      1       -    3/27 09:37  matlab.exe   307333.0
133      21     -       -    3/27 11:46  DAG: 311986  312342.0 ... 313304.0

In the last 20 minutes:
  0 Job(s) were Completed
  5 Job(s) were Started
  1 Job(s) were Held

  263203.0  5/11 07:22 Error from slot1@eee.chtc.wisc.edu: out of disk
```
HTCondor and Kerberos

› HTCondor currently allows you to authenticate users and daemons using Kerberos.

› However, it does NOT currently provide any mechanism to provide a Kerberos credential for the actual job to use on the execute slot.
HTCondor and Kerberos

- We have added support in HTCondor 8.5.X to allow an opaque security credential to be obtained by `condor_submit` and stored securely in the `condor_credd` daemon.
- This credential is then moved with the job to the execute machine.
- Before the job begins executing, the `condor_starter` invokes a call-out to do optional transformations on the credential.
For example, the call-out can use that opaque credential to obtain a fresh Kerberos TGT for the user, and also obtain AFS tokens for the job.

This work in collaboration with CERN will allow user their jobs to use Kerberos-authenticated services and also write directly into AFS home directories.

Expected to appear in HTCondor 8.5.4.
Grid Universe

› Reliable, durable submission of a job to a remote scheduler
› Popular way to send pilot jobs
› Supports many “back end” types:
  • HTCondor
  • PBS
  • LSF
  • Grid Engine
  • Google Compute Engine
  • Amazon EC2
  • OpenStack
  • Deltacloud
  • Cream
  • NorduGrid ARC
  • BOINC
  • Globus: GT2, GT5
  • UNICORE
Native SLURM and OpenStack Support

› Speak native SLURM protocol
  • No need to install PBS compatibility package

› Speak OpenStack’s NOVA protocol
  • Better supported than OpenStack's EC2 compatibility interface
Scalable mechanism to grow pool into the Cloud

› Leverage efficient AWS APIs such as Auto Scaling Groups and Spot Fleets
  • Implement a “lease” so charges cease if lease expires

› Secure mechanism for cloud instances to join the HTCondor pool at home institution

condor_annex --set-size 2000
  --lease 24 --project “144PRJ22”
More condor_drain / condor_defrag options

› Backfill pre-emptable jobs while draining?
› Drain and shutdown?

I am altering the slot. Pray I don't alter it any further!
Transformation of job ad upon submit

› Allow admin to have the schedd add/edit job attributes upon submission
  
  (use case: insert group attributes based upon owner)
Docker 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
  - Why?
    - CVMFS
    - Large shared data
  - Details

https://htcondor-wiki.cs.wisc.edu/index.cgi/tktview?tn=5308
Potential Future Docker Universe Features?

› Advertise images already cached on machine?
› Support for condor_ssh_to_job?
› Package and release HTCondor into Docker Hub?
› Network support beyond NAT?
› Run containers as root??!?!?
› Automatic checkpoint and restart of containers! (via CRIU)
What do YOU need?
Thank You!

Please help us, high throughput computing. You’re our only hope!