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

HTCondor Week 2013

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

› Development Series
  • HTCondor v7.9.6 frozen, in beta test (release candidate for v8.0.0)
  • Series v7.9.x now dead, v8.1.x in ~four weeks.

› Stable Series
  • End of May: Condor v8.0.0
  • v7.8.8 will likely be the last v7.8.x released
  • Last Year: Condor v7.8.0 (May 10th 2012)

› 16 releases since Condor Week 2012
Six key HTC challenge areas
Challenge 1

Evolving Resource Acquisition Models

Cloud services – fast and easy acquisition of compute infrastructure for short or long time periods.

› Research effective management of large homogenous workloads on homogenous resources

› Policy-driven capabilities to temporarily augment local resources

› React to how cloud providers offer resources
Challenge 2

Hardware Complexity

As the size and complexity of an individual compute server increases, so does the complexity of its management.

› Modern servers have many disparate resources leading to disparate job mixes

› Increased need for effective isolation and monitoring
Challenge 3

Widely Disparate Use Cases

As a result of increased demand for higher throughput, HTC technologies are being called upon to serve in a continuously growing spectrum of scenarios.

› Increasing need from non-admins

› Must continue to be expressive enough for IT professionals, but also tuned for intended role, aware of target environment, and approachable by domain scientists
Challenge 4

Data Intensive Computing

Due to the proliferation of data collection devices, scientific discovery across many disciplines will continue to be more data-driven.

› Increasingly difficult to statically partition and unable fit on a single server.

› Integration of scalable storage into HTC environments.
Contemporary HTC users, many of whom have no experience with large scale computing, are much less knowledgeable about the codes they run than their predecessors.

- Goal: “You do not need to be a computing expert in order to benefit from HTC.”
- Unknown software dependencies, requirements
- Often environment must change, not application
Challenge 6

Scalability

Sustain an order of magnitude greater throughput without increasing the amount of human effort to manage the machines, the jobs, and the software tools.

› Grouping and meta-jobs.

› Submission points that are physically distributed (for capacity), but logically unified (for management)
Official Ports for v8.0.0

› Compatible w/ v7.8.x

› Binary packages available for
  • Windows XP SP3+ (runs on 32bit or 64bit)
  • Debian 5 (x86_64)
  • Debian 6 (x86, x86_64)
  • RHEL 5 (x86, x86_64)
  • RHEL 6 (x86_64)
  • MacOS 10.7 (x86_64)

› Adding RHEL 7, Windows 8, Debian 7

› Of course source code as well

› Continue to push into distro repositories
New goodies with v7.8

- Scheduling:
  - Partitionable Slot improvements
  - Drain management
  - Statistics

- Improved slot isolation and monitoring

- IPv6

- Diet! (Shared Libs)

- Better machine descriptions

- Absent Ads

- …
New goodies with v8.0

- HTCondor-CE
- Bosco
- DAGMan additions
- EC2 Spot, OpenStack
- Several new tools
- ClassAd Compression
- Generic Slot Resources
- Python Interfaces
- Job Sandboxing

- Interactive jobs
- Open development process progress
- Security policy maturation
- Many more…
Memory, CPU, Disk no longer hard coded – can define new machine (startd) resources.

In condor_config:
MACHINE_RESOURCE_BoosterRockets = 25

In condor_submit:
request_cpu = 1
request_BoosterRockets = 4
Python Interface

- Some HTCondor client API choices:
  - Command line
  - DRMAA Version 1.x (C bindings)
  - Web Service (SOAP) : built-in or Aviary contrib
  - REST: condor-agent contrib

- And now… Python!
  - Built on top of HTCondor’s shared libraries
  - Linux only
  - Interact with ClassAds, Collector, Schedd
Job Sandboxing

Real-time protection on Linux of: CPU cores, /tmp, run-away processes, memory, processes running as the same user as the job

ASSIGN_CPU_AFFINITY = true
MOUNT_UNDER_SCRATCH = /tmp, /var/tmp
BASE_CGROUP = htcondor
CGROUP_MEMORY_LIMIT_POLICY = hard
USE_PID_NAMESPACES = true

Also have chroot support!
Let’s add some spice…

YOU USED SO MUCH OIL
THE US IS TRYING TO
INVADE THE PLATE

© 20thC. Fox/Everett / Rex Features
SousDo Chef TJ
New Tools
New Tools in the Kitchen

- **condor_tail**
  - Fetch output from running jobs
  - Follow (tail) stdout, stderr or other file

- **condor_submit --interactive**
  - Schedule interactive shell, no logins on execute machines required, job removed if user goes away

- **condor_ping**
  - Check communication path and security

- **condor_qsub**
  - Use qsub syntax to submit HTCondor jobs
  - Useful is you have scripts designed to submit to SGE or PBS
New Tools in the Kitchen, cont

› condor_q -better-analyze
  • More detailed matchmaking analysis
  • Analyze machine START expressions
  • Match summary for multiple jobs/machines

› condor_who
  • Query local STARTD(s) about running jobs
  • Does not require access to the collector

› condor_gather_info
  • Supply a job id, it gathers debugging info from logs about that job
First up: Contestant Nathan
HTCondor in Matlab

- Useful for users who like to live in Matlab
  - No need to drop to a shell or editor
  - Comfortable environment
  - Don’t use submit files. Transparent to user

Credit and Questions:
Giang Doan - gdoan at cs.wisc.edu
% >> DrawGraphGUI
Please enter parameters:

Start  End  Number of points  Function
          20  20  a.out

Click on "Start" to execute a new task, or "EXIT" to exit application.

Figure 1: Graph2D

Figure 2: Histogram

Completed: 198.60%

Run-time distribution of the jobs
How does it taste?

Next up: Contestant Todd
Cooking with Clouds
Improved Support for EC2

“The nice thing about standards is that there’s so many of them to choose from.”

• Amazon
• Nimbus
• Eucalyptus
• OpenStack
Amazon Spot Instances

- User: cheap but unreliable resources.
- HTCondor: complicated resource life-cycle.
  - Spot instance request is a different object in the cloud than the running instance.
  - We restrict certain features to ensure that only one of those objects is active a time to preserve our usual job semantics.
I will bravely claim that It Just Works™.

However, because too much whitespace is bad space, I’ll mention here that we also substantially improved the efficiency of our status updates by batching the requests, making one per user-service pair rather than one per job.
Eucalyptus

› Version 3 requires special handling, so we added a per-job way to specify it.
OpenStack

› Restrictive SSH key-pair names for all.
› Added handling for nonstandard states
  • SHUTOFF doesn’t exist
  • STOPPED is impossible
  • We terminate and report success for both.
How does it taste?

Next up: Contestant Alan
The power of HTCondor

The ease of use of Microsoft Kinect*

* The CHTC and HTKinect are not connected with Microsoft in any way.
HTKinect 0.13 PRERELEASE May 24 2013 BuildID: 120303
x86_64_rhap_6.3
Connecting to HTCondor on puffin.cs.wisc.edu...
Connected.

HTCondor 7.9.8 PRERELEASE May 24 2013 BuildID: 120298
x86_64_rhap_6.3
Scanning for user...
No user detected, please enter camera view
Horse detected... Unable to process
Chevrolet Impala detected... Unable to process
Nerd detected... accepted

HTKinect ready
> Wipe
HTKinect ready

> Scan

% condor_q

<table>
<thead>
<tr>
<th>ID</th>
<th>OWNER</th>
<th>SUBMITTED</th>
<th>RUN_TIME</th>
<th>ST</th>
<th>PRI</th>
<th>SIZE</th>
<th>CMD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended

HTKinect ready

> Wipe
HTKinect ready
> Forward
% condor_submit default.submit
Submitting job(s).
1 job(s) submitted to cluster 62.

HTKinect ready
> Scan
% condor_q
% condor_q
ID     OWNER     SUBMITTED    RUN_TIME  ST PRI SIZE CMD
62.0    adesmet  5/2  16:51   0+00:00:00 I  0  97.7 sleep 1200

1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended

HTKinect ready
> Wipe
HTKinect ready
> Hug
% condor_hold 62
Cluster 62 held.

HTKinect ready
> Thumbs Down
% condor_rm 62
Cluster 62 has been marked for removal.

HTKinect ready
> Wipe
HTKinect ready
> Checklist
% cat TODO.txt
- Finish HTCondor Week slides
- Send money order to Nigerian prince
- Call tech support; get cup holder fixed
- Write design document for mixed mode IPv4/IPv6 mode

HTKinect ready
> Spyglass
% ls ~/private
HTCondor-Week-budget.xls  My_Little_Pony_episodes/
my-D&D-movie-script.doc  Twilight-fan-fiction/

HTKinect ready
> Empty Trash
% sudo rm -rf /
ERROR: connection to puffin.cs.wisc.edu lost
Next up: Contestant Dan

How does it taste?
File Transfer Management

› Old features:
  
  • Limits:
    • MAX_CONCURRENT_UPLOADS=10
    • MAX_CONCURRENT_DOWNLOADS=10
  
  • Monitoring:
    condor_status –schedd –long
    • TransferQueueMaxUploading/Downloading
    • TransferQueueNumUploading/Downloading
    • TransferQueueNumWaitingToUpload/Download
    • TransferQueueUpload/DownloadWaitTime
Problem: Mr. BigData submits thousands of jobs that transfer GBs of data
   • Hogs transfer queue for many hours

New in 7.9:
   • Equal share between users in transfer queue
   • Or can have equal share between some other grouping of jobs:
     TRANSFER_QUEUE_USER_EXPR
     – e.g. group by destination grid site
Better Visibility

› Jobs doing transfer used to be in ‘R’ state
  • Hard to notice file transfer backlog

› In 7.9 they display in condor_q as
  ‘<’ (transferring input)
  ‘>’ (transferring output)

› The transfer state is in job ClassAd attributes:
  • TransferringInput/Output = True/False
  • TransferQueued = True/False
condor_rm BigData

• This used to put jobs in transfer queue into ‘X’ state
  • Stuck in ‘X’ until they finish the transfer!

• In 7.9, removal is much faster

• Also applies to condor_hold
New controls on max transfer size:
  • Submit-node configuration:
    • MAX_TRANSFER_INPUT_MB
    • MAX_TRANSFER_OUTPUT_MB
  • Job submit file:
    • max_transfer_input_mb
    • max_transfer_output_mb
  If exceeded, job is put on hold
  • At submit time, if possible
  • Otherwise, at transfer time
Monitoring I/O Usage

- `condor_status --schedd --long --statistics "TRANSFER:2" --direct "schedd_name"`

  - Aggregate and per-user metrics averaged over 1m, 5m, 1h, 1d, and/or whatever you configure:
    - Bandwidth - bytes/s
    - Network load - transfers blocked in read/write
    - Disk load - transfers blocked in read/write
Limitations of New File Transfer Queue Features

- Doesn’t apply to grid or standard universe
- Doesn’t apply to file transfer plugins
- Windows still has the problem of jobs hanging around in ‘X’ state if they are removed while transferring
How does it taste?

Next up: Contestant Jaime
BOINC

› Volunteer computing
  • 250,000 volunteers
  • 400,000 computers
  • 46 projects
  • 7.7 PetaFLOPS/day

› Based at UC-Berkeley
You Got BOINC in My HTCondor!

› BOINC state in HTCondor

• Run BOINC jobs when no HTCondor jobs available
• Supported in HTCondor for years
• Now generalized to Backfill state
You Got HTCondor in My BOINC!

- Now we complete the circle
- HTCondor will submit jobs to BOINC
  - New type in grid universe
HTCondor and BOINC

› Two great tastes that taste great together!
How does it taste?

Next up: Contestant Zach
Condor module for integration…

…with Facebook!
Submit a Job

Upload job input data

enter your submit file here...
Good Morning! Just sipping some coffee and testing this submit node:

universe = vanilla
executable = /bin/sleep
arguments = 300
queue

Zach Miller, Jaime Frey, and 4 others like this.

View 4 more comments

Zach Miller  You should set "Notification = Never otherwise you extra email"
19 hours ago · Like · 2

Todd Tannenbaum  OMG you are so right i got like a million emails and crashed the whole internet LOL
19 hours ago · Like
SHARE THIS IF YOU DEPEND ON YOUR CLUSTER! I KNOW THAT MOST OF YOU WONT DO IT BUT MY REAL COLLABORATORS WILL!!!
ugh, don't wanna run any of your jobs. i just want someone to negotiate with but i h8 it when people don't explain themselves

Todd Tannenbaum  what is wrong?
Yesterday at 12:27am  ·  Like

condor_q - analyze  i don't want to talk about it. UNKNOWN REASONS.
Yesterday at 12:49am via mobile  ·  Like
How does it taste?

Next up: Contestant Greg
HTCondor Scheduling:
Can do ANYTHING:

Start = (((((RealExperiment == "atlas") && (VirtualMachineID >= 7)) && (TARGET.RACF_Group == "short") ||
TARGET.RACF_Group == "dial" || Owner = "usatlas2" || (stringListMember("acas0201",
"acas0200,acas0201,acas0202,acas0203,acas0204") &&
TARGET.RACF_Group == "lcg-ops") ||
(stringListMember("acas0201", "acas0200,acas0201,acas0202,acas0203,acas0204") &&
TARGET.RACF_Group == "lcg-dteam")) &&
(RemoteWallClockTime < 5400))) || ((RealExperiment == "atlas") && (VirtualMachineID < 7) &&
(VirtualMachineID >= 5)) && ((TARGET.RACF_Group == "usatlas" || TARGET.RACF_Group == "usatlas-grid"
||
(stringListMember("acas0201", "acas0200,acas0201,acas0202,acas0203,acas0204") &&
TARGET.RACF_Group == "lcg-atlas") ||
TARGET.RACF_Group == "bnl-local") && (((vm7_Activity == "Busy") + (vm7_Activity ==
"Retiring") + (vm8_Activity == "Retiring") + (vm8_Activity == "Busy")) < 2))) || ((RealExperiment ==
"atlas") && (VirtualMachineID >= 3) && (VirtualMachineID < 5)) && ((TARGET.RACF_Group == "grid"
||
(stringListMember("acas0201", "acas0200,acas0201,acas0202,acas0203,acas0204") == FALSE &&
TARGET.RACF_Group == "lcg") && (((vm7_Activity == "Busy") + (vm7_Activity == "Retiring") +
(vm8_Activity == "Retiring") + (vm8_Activity == "Busy") + (vm5_Activity == 
"Retiring") + (vm6_Activity == "Retiring") + (vm6_Activity == "Busy"))) < 2))) ||
(((RealExperiment == "atlas") || (RealExperiment != "atlas" && FALSE == FALSE && TRUE == FALSE &&
LoadAvg < 1.400000 && TotalVirtualMemory > 200000 && ((Memory * 1024) - ImageSize) > 100000)) &&
((VirtualMachineID >= 1) && (VirtualMachineID < 3)) && ((TARGET.RACF_Group == "gridgr01"
||
TARGET.RACF_Group == "gridgr02" || TARGET.RACF_Group == "gridgr03" || TARGET.RACF_Group ==
"gridgr04") ||
TARGET.RACF_Group == "gridgr05" || TARGET.RACF_Group == "gridgr06" || TARGET.RACF_Group ==
"gridgrXX" || TARGET.RACF_Group == "gridgr08" || TARGET.RACF_Group == "gridgr09" ||
TARGET.RACF_Group == "gridgr10" || TARGET.RealExperiment != "atlas") && (((vm7_Activity == "Busy") + (vm7_Activity ==
"Retiring") + (vm8_Activity == "Retiring") + (vm8_Activity == "Busy") + (vm5_Activity == 
"Retiring") + (vm6_Activity == "Retiring") + (vm6_Activity == "Busy") +
(vm3_Activity == "Busy") + (vm3_Activity == "Retiring") + (vm4_Activity == "Retiring") +
(vm4_Activity == "Busy"))) < 2))))) && (Owner != "jalex" && Owner != "grau" && Owner != "smithj4")
&& FALSE == FALSE)
Existing Scheduling Problems

› Assumes Preempt / Resume
› Assumes every machine a snowflake
  • Every job unique also

› Two tiers of provisioning + scheduling
› Difficult to configure, debug or monitor
› Partitionable slot infelicities
Planned for 8.1

- Slot splitting in the negotiator
- Negotiator knows “consumption policies”
Defining higher level semantics

- “Owned Resources” + Overflow
  - Condo model as first class
- Provision machines and jobs in sets
- Ganglia interface to negotiator
- Special case the one-schedd pool
- Switching to incremental model
- Remove need for STARTD RANK
Of Course…

- We won’t break anything existing
- “Provisioning on the side”…
- Have interesting/difficult scheduling reqs?
  - Please talk to me.
The Results
Thank you!

Questions?