What Are My Jobs Doing?

HTCondor Week 2022
Christina Koch
Before We Start: Submit Sample Jobs

There will be a live demo as part of this talk. If you want to follow along, and have an account on an HTCondor Access Point:

• Log in to your access point
• Run:
  • git clone https://github.com/CHTC/job-info-examples
  • cd job-info-examples
  • condor_submit simplejobs.submit
Motivation

We talk a lot about how to submit jobs…

…and not so much about how to get information about them after you submit.

This talk will cover different tools that you can use to answer questions about your jobs, like:

• What is my job doing?
• What resources is it using?
• Why is it on hold?
• Etc.
Tools for Learning About Jobs

HTCondor's job attribute information
• Accessed via condor_q, or condor_history

Observing job or outputs directly with HTCondor commands, e.g.
• condor_tail
• condor_ssh_to_job

Files
• HTCondor log files
• Standard error/standard output files
Is my job waiting (idle) or running?

- `condor_q` (with or without `-nobatch`)
- `condor_watch_q` for an updating display

```
$ condor_watch_q
BATCH    IDLE  RUN  DONE  TOTAL  JOB_IDS
ID: 5130  -    5    -    5     5130.0 ... 5130.4

[============================================================]

Total: 5 jobs; 5 running

Updated at 2022-05-23 10:12:42
Input ^C to exit
```
What is my (running) job doing?

Look at standard error and standard output files (usually have relevant messages).

• `condor_tail JobID`
• `condor_tail -stderr JobID`

Log into the computer where the job is running to explore.

• Log into the job using `condor_ssh_to_job JobID`
• Use shell commands to look at files, working directory

If your HTCondor system uses opportunistic resources, you may not always have the access needed for `condor_ssh_to_job` to work.
How long has my job been running?

HTCondor command: `condor_q -run`

- To see current (not cumulative) run time: `condor_q -run -current`

File option: the log file

```
$ condor_q -run

-- Schedd: submit.univ.edu : <>@ 05/22/22 10:15:19

<table>
<thead>
<tr>
<th>ID</th>
<th>OWNER</th>
<th>SUBMITTED</th>
<th>RUN_TIME</th>
<th>HOST(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5130.0</td>
<td>alice</td>
<td>5/23 10:12</td>
<td>0+00:01:53</td>
<td><a href="mailto:slot1_9@e3156.univ.edu">slot1_9@e3156.univ.edu</a></td>
</tr>
<tr>
<td>5130.1</td>
<td>alice</td>
<td>5/23 10:12</td>
<td>0+00:01:54</td>
<td><a href="mailto:slot1_44@e1038.univ.edu">slot1_44@e1038.univ.edu</a></td>
</tr>
<tr>
<td>5130.2</td>
<td>alice</td>
<td>5/23 10:12</td>
<td>0+00:01:53</td>
<td><a href="mailto:slot1_25@e328.univ.edu">slot1_25@e328.univ.edu</a></td>
</tr>
</tbody>
</table>
```
How many resources is my job using?

To answer this question, we need to talk about... Attributes!
Job Attributes

HTCondor stores a list of information about each job. This information is stored in this format:
- **AttributeName** = value

You can find a list of attributes for a single job by running:
- `condor_q -l JobID`

You can print out specific attributes by using the “format” or “auto-format” flags with an HTCondor command:
- `condor_q -af Attribute1 Attribute2`
How many resources is my job using?

Use the “Request” and “Usage” attributes.

For example, for memory, use RequestMemory and RequestUsage

- `condor_q -af RequestMemory MemoryUsage`
- To summarize, add the “sort” and “uniq” commands:
  - `condor_q -af RequestMemory MemoryUsage | sort | uniq -c`

Can also swap out Memory with CPUs or Disk

Note that the “Usage” attributes are not always updated in real-time.
What resources did my job use?

Can answer the same questions for finished jobs using condor_history

- `condor_history user -limit 2 -af RequestMemory MemoryUsage`

Use the “-limit” flag to get results more quickly.

This is really useful for summarizing test results!!

- `condor_history user -limit 100 -af RequestMemory MemoryUsage | sort | uniq -c`

Again, can look at Disk and CPU usage in the same way.
Where did my job run?

Attribute option: answerable using condor_history!

- `condor_history user -limit 2 -af LastRemoteHost`
- This can be useful when looking for patterns of failures or successes.
- Only shows the last place a job ran.

File option: the log file (shows all places a job ran)

If a job is still running, can use:

- `condor_q -run`
- `condor_q -af RemoteHost`
Interlude: Submit More Jobs

Are you still in the “job-info-examples” folder?
Run:
  * condor_submit complexjobs.submit
What did I run? What is still running?

The “Cmd” and “Args” attributes are useful for recovering details of a job (executable, arguments):

- `condor_q user -af:jh Cmd Args`
- `condor_history user -limit 10 -af:jh Cmd Args`

We’re using the “j” (for JobID) and “h” (for ”header”) in addition to the auto-format option above -- this prints out the JobID along with the requested attributes.
Where are my files? and other questions.

Can also look up:
- Where the job was submitted: `Iwd`
- Where the log file is: `UserLog`
- Where stdout and stderr are: `Out`, `Err`
- A job’s exit code: `ExitCode`
- A job’s requirements: `-af:r` `Requirements`

Time attributes
- There *are* timestamps for different events but they are in epoch time, so you have to do some work to read them.
Where can I see all the attributes?

See the manual for a list of all the attributes HTCondor can use.

- Manual > Appendix > ClassAd Attributes > Job ClassAd Attributes

For the JobStatus attribute (which we will see soon), can use this command to print out a list of the JobStatus codes:

- `condor_q -help status`
Why is my job on hold?

**Condor command:** condor_q -hold

**Use job attributes:**
- condor_q -af HoldReasonCode HoldReason

**File option:** Log file

```shell
$ condor_q -af:j HoldReason
5131.1 Error from slot1_24@e2353.univ.edu: SHADOW at 128.105.244.190 failed to send file(s) to <128.105.69.31:40955>: error reading from /home/alice/job-info-examples/test.txt: (errno 2) No such file or directory; STARTER failed to receive file(s) from <128.105.244.190:9618>
```
How do I see just “x” jobs?

So far, we have simply been selecting specific jobs using their JobID or all the jobs associated with a single user.

What if you want to select a more specific set of jobs, like all jobs that are still idle?

This can be done with attributes and the “-constraint” option to `condor_q`

```
• condor_q -constraint JobStatus == 1'
```

The constraint option works with other HTCondor job-related commands (condor_hold, condor_rm, condor_release, etc.)
What kind of computers are there?

We can use similar principles to learn about the computers (machines) in an HTCondor Pool using `condor_status`.

To get a list of attributes, use the “long” option

```bash
• condor_status -l [name]
```

Then look at certain attributes (like Machine, TotalCpus, CPUModel) using the “auto format” flag.

```bash
• condor_status -af Machine TotalCpus CpuModel | sort | uniq -c
```
Questions

What other information would you like to be able to learn about your jobs?
Acknowledgements

This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATH Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.