Whirlwind Tour of HTCondor Tools

Greg Thain
Center for High Throughput Computing
University of Wisconsin - Madison

Rationale

In the past, we've started with a simple beginner's tutorial

The basics of job submissions, etc.

This is not that talk. That talk can be found on YouTube:

https://youtu.be/8jadQkAdU1k

This talk is a high-throughput talk trying to cover all the tools:

Every* HTCondor tool in 40 minutes

Greg Thain
Center for High Throughput Computing
University of Wisconsin - Madison

CENTER FOR HIGH THROUGHPUT COMPUTING

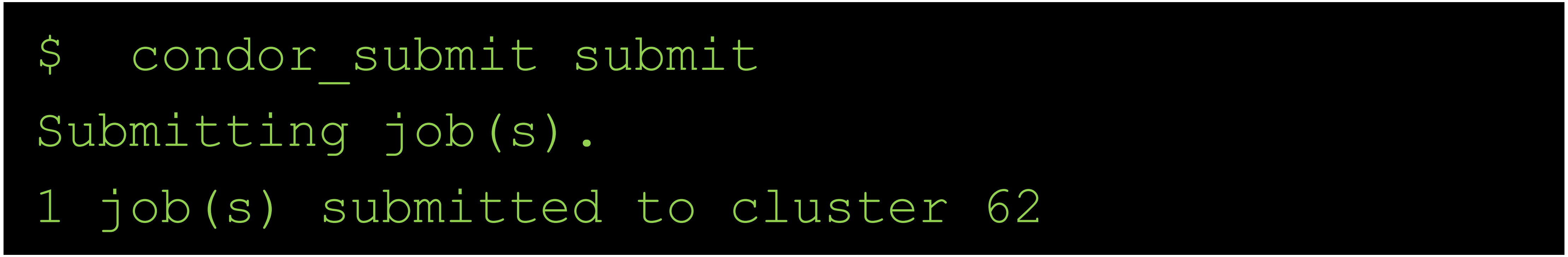
PATh PARTNERSHIP to ADVANCE THROUGHPUT COMPUTING

HTCondor Software Suite

- Commands for managing jobs
  - Condor_submit,condor_transfer_data, condor_submit_dag, condor_rm condor_suspend / Condor_continue,condor_hold / condor_release condor_q / condor_history, condor_qedit, condor_q -better
- Commands for managing execution points
  - condor_off, condor_on, condor_restart, Condor_drain, condor_now, condor_vacate, condor_config_val-set,condor_reconfiig, condor_status,
- Commands for working with running jobs
  - Condor_ssh_to_job, condor_tail, condor_evicted_files, condor_chirp, Condor_vacate_job
- Commands for debugging and testing
  - Condor_version, Classad_eval, condor_who, condor_top, condor_fetchlog,condor_transform_jobs, condor_starter –classad, condor_gpu_discovery, condor_power_state
- Command(s) for managing submitters
  - condor_userprio, condor_qusers
- The new way (experimental)
  - htcondor tool

Commands for managing jobs

condor_submit

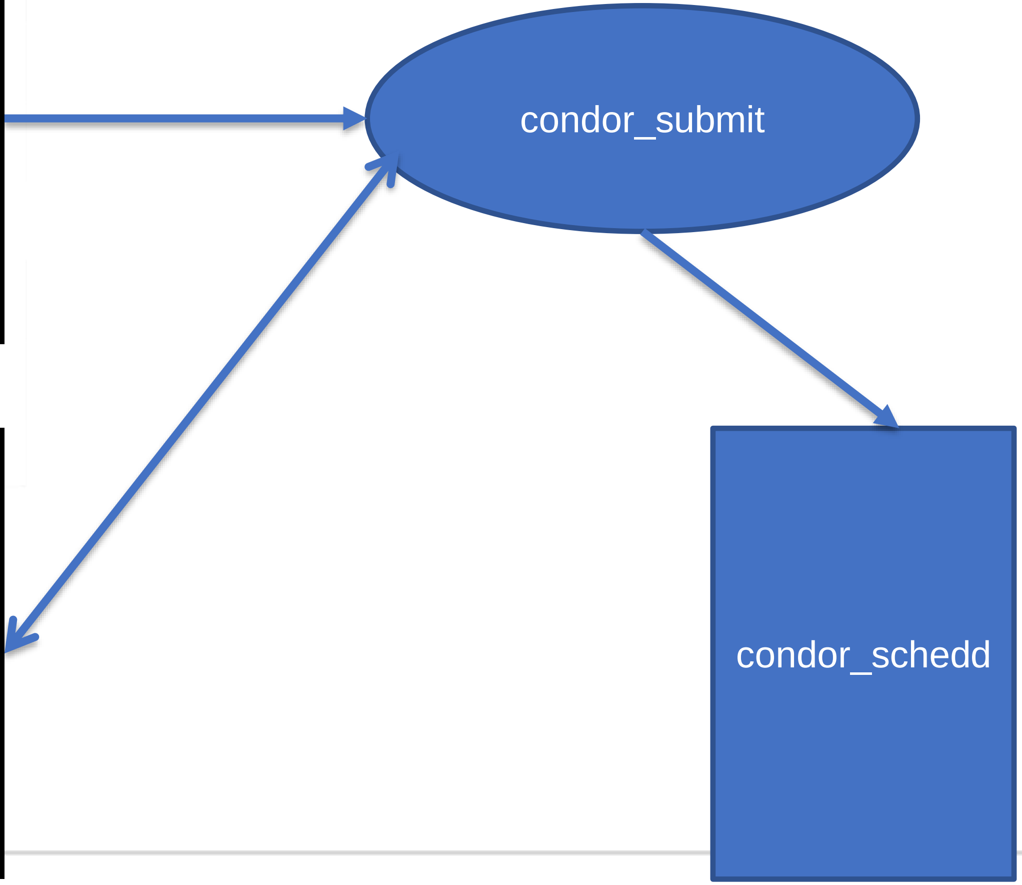


```
$  condor_submit submit
Submitting job(s).
1 job(s) submitted to cluster 62
```

condor_submit animation

```
executable = calculate
arguments = one two three
queue
…
```

```
Cmd = "calculate"
Qdate = 34056
Args = "one two three"
…
```



condor_submit -dry-run classad_file

```
$  condor_submit -dry-run /dev/stdout submit
Dry-Run job(s)
ClusterId=1
In="/dev/null"
EnteredCurrentStatus=1664855180
Environment=""
PeriodicHold=false
…
```

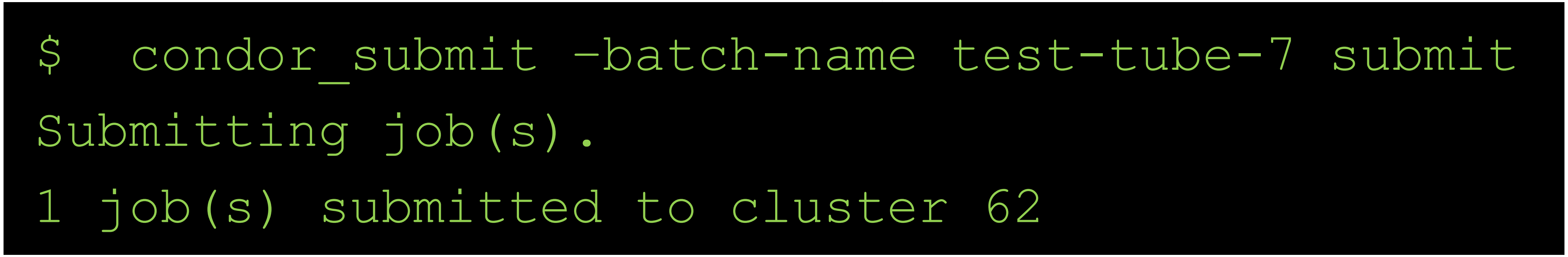
condor_submit –i (interactive)

```
$  condor_submit -i
```

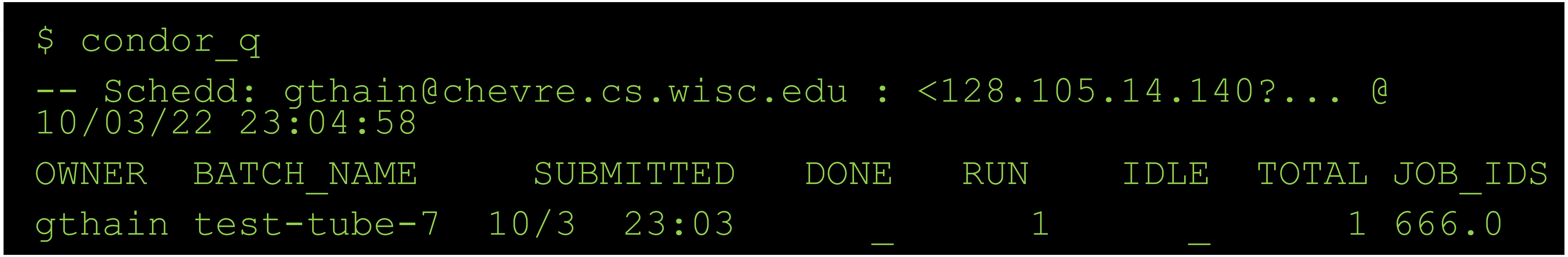
OR

```
$  condor_submit -i some_submit_file
```

condor_submit -batch-name experiment#



```
$  condor_submit -batch-name test-tube-7 submit
Submitting job(s).
1 job(s) submitted to cluster 62
```



```
$ condor_q
-- Schedd: gthain@chevre.cs.wisc.edu : <128.105.14.140?... @
10/03/22 23:04:58
OWNER  BATCH_NAME     SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
gthain test-tube-7  10/3  23:03      _      1      _      1 666.0
```

condor_submit -spool

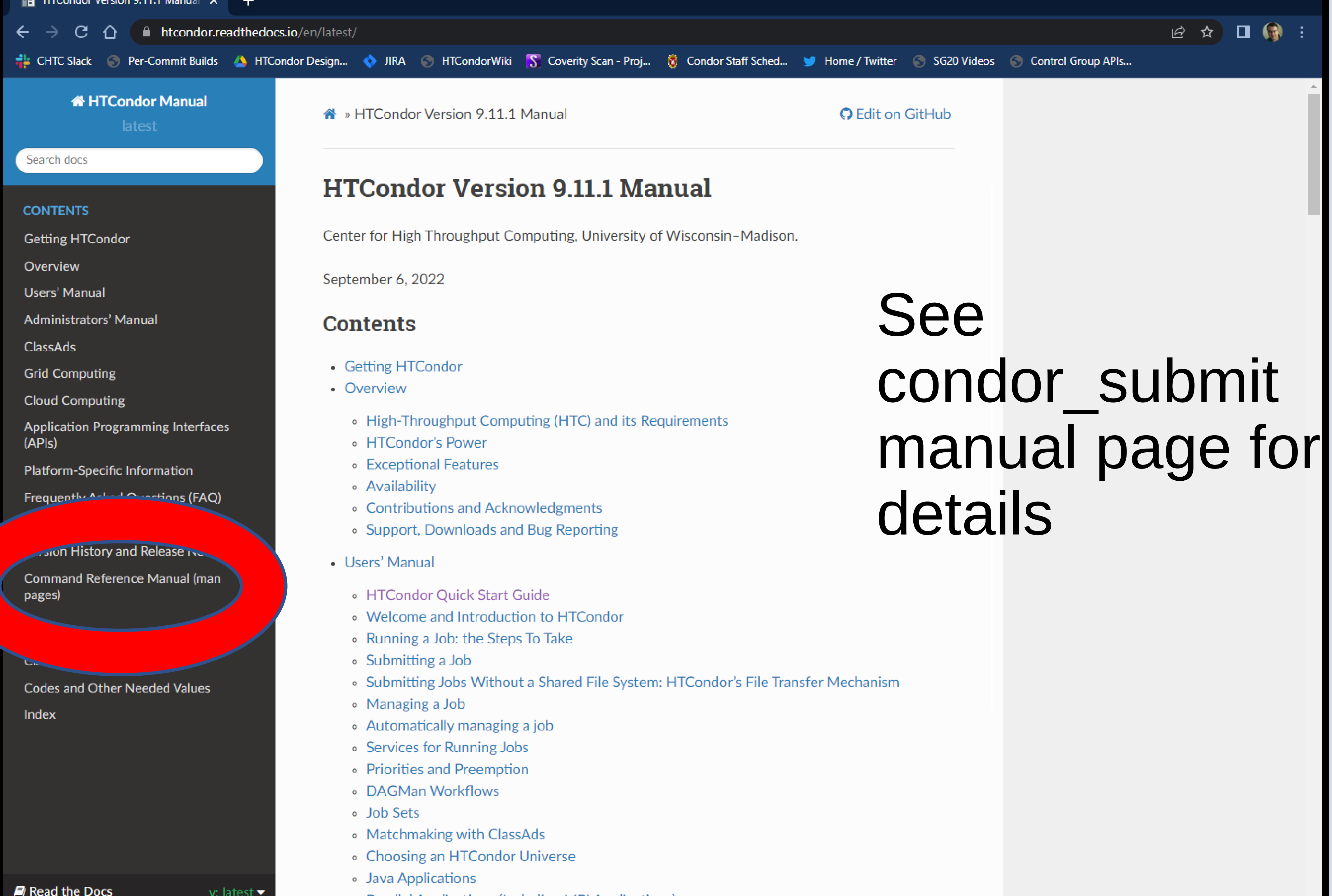
```
$  condor_submit -spool submit…
```

condor_submit –remote sched_name

```
$  condor_submit -remote sched_name
```

condor_transfer_data

```
$ condor_transfer_data cluster.proc
```



» HTCondor Version 9.11.1 Manual

CONTENTS

- Getting HTCondor
- Overview
- Users' Manual
- Administrators' Manual
- ClassAds
- Grid Computing
- Cloud Computing
- Application Programming Interfaces (APIs)
- Platform-Specific Information
- Frequently Asked Questions (FAQ)
- sion History and Release N
- Command Reference Manual (man pages)
- C
- Codes and Other Needed Values
- Index

HTCondor Version 9.11.1 Manual

Center for High Throughput Computing, University of Wisconsin–Madison.

September 6, 2022

Contents

- Getting HTCondor
- Overview
  - High-Throughput Computing (HTC) and its Requirements
  - HTCondor's Power
  - Exceptional Features
  - Availability
  - Contributions and Acknowledgments
  - Support, Downloads and Bug Reporting
- Users' Manual
  - HTCondor Quick Start Guide
  - Welcome and Introduction to HTCondor
  - Running a Job: the Steps To Take
  - Submitting a Job
  - Submitting Jobs Without a Shared File System: HTCondor's File Transfer Mechanism
  - Managing a Job
  - Automatically managing a job
  - Services for Running Jobs
  - Priorities and Preemption
  - DAGMan Workflows
  - Job Sets
  - Matchmaking with ClassAds
  - Choosing an HTCondor Universe
  - Java Applications

See condor_submit manual page for details

condor_submit one liners

```
$  condor_submit - <<EOF
Executable = /bin/sleep
Arguments = 3600
Request_memory = 1G
Request_Disk = 1000M
Queue
EOF
```

condor_submit one liners

```
$  condor_submit executable=/bin/sleep \
        arguments=3600 -queue 1 - < /dev/null
```

Every other HTCondor tool in 30 minutes

HT Center for High Throughput Computing

PATh Partnership to Advance Throughput Computing

HTCondor Software Suite

condor_submit_dag

```
$ condor_submit_dag dag_file
```

Doesn't have options as condor_submit

Only option of interest is "-force"

See:

https://youtu.be/OuIBf6x24r0

If only one command, condor_submit

But if you know two, the second should be …

condor_rm

```
$ condor_rm cluster.proc
```

```
$ condor_rm cluster
```

```
$ condor_rm username
```

```
$ condor_rm -all
```

condor_rm -const

```
$ condor_rm -const 'some classad expr'
```

```
$ condor_q -const 'some classad expr'
```

condor_rm –reason "Reason String"

```
$ condor_rm -reason 'Call Greg - this exploded'
```

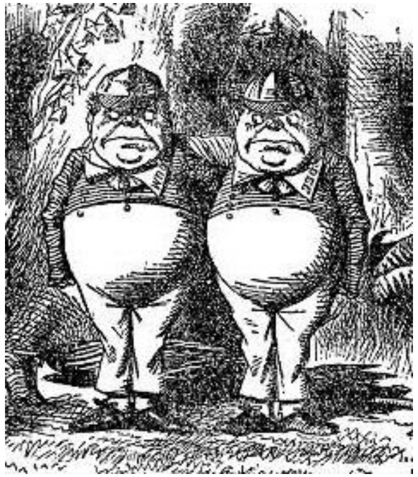
All of the "act on jobs" commands have these same options

```
$ some_condor_command cluster.proc
```

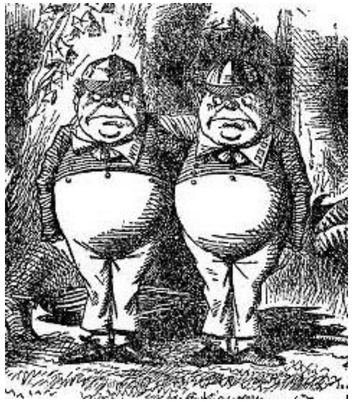
```
$ some_condor_command cluster
```

```
$ some_condor_command username
```

```
$ some_condor_command -all
```

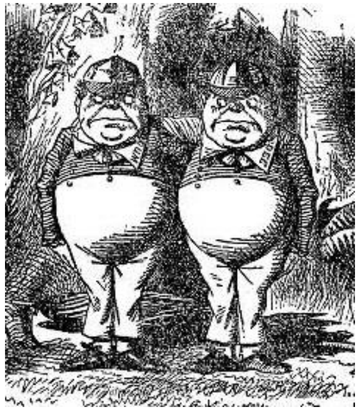


The rest of the "act on jobs" commands…



Three sets of twins

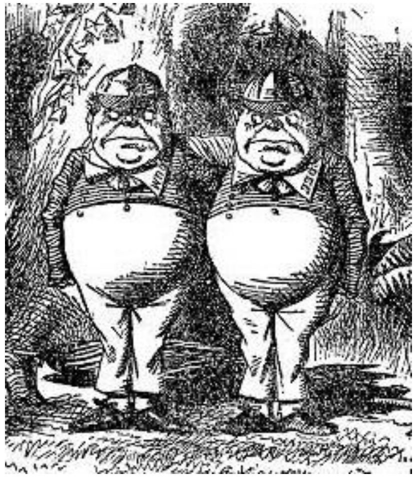
And an extra…



condor_suspend / condor_resume

```
$ condor_suspend cluster.proc (or username or …)
```

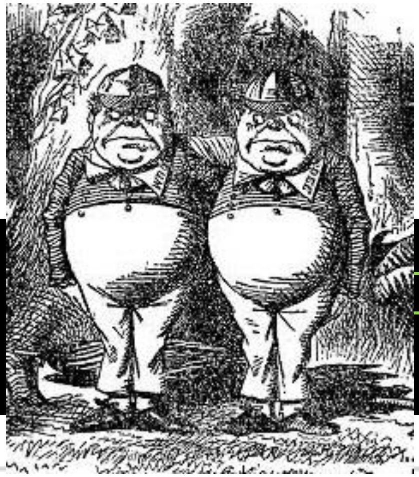
```
$ condor_resume cluster.proc (or username or …)
```



condor_hold / condor_release

```
$ condor_hold cluster.proc (or username or …)
```

```
$ condor_release cluster.proc (or username or …)
```

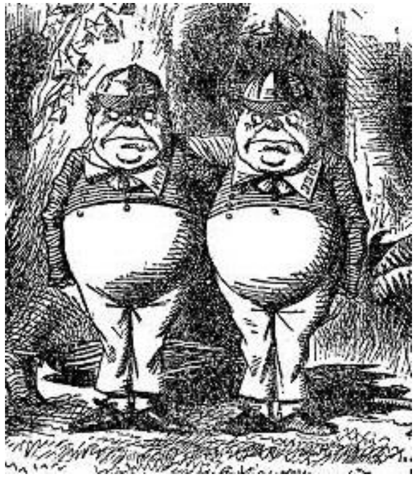


```
$ condor_hold -re                    L GREG NOW'
```

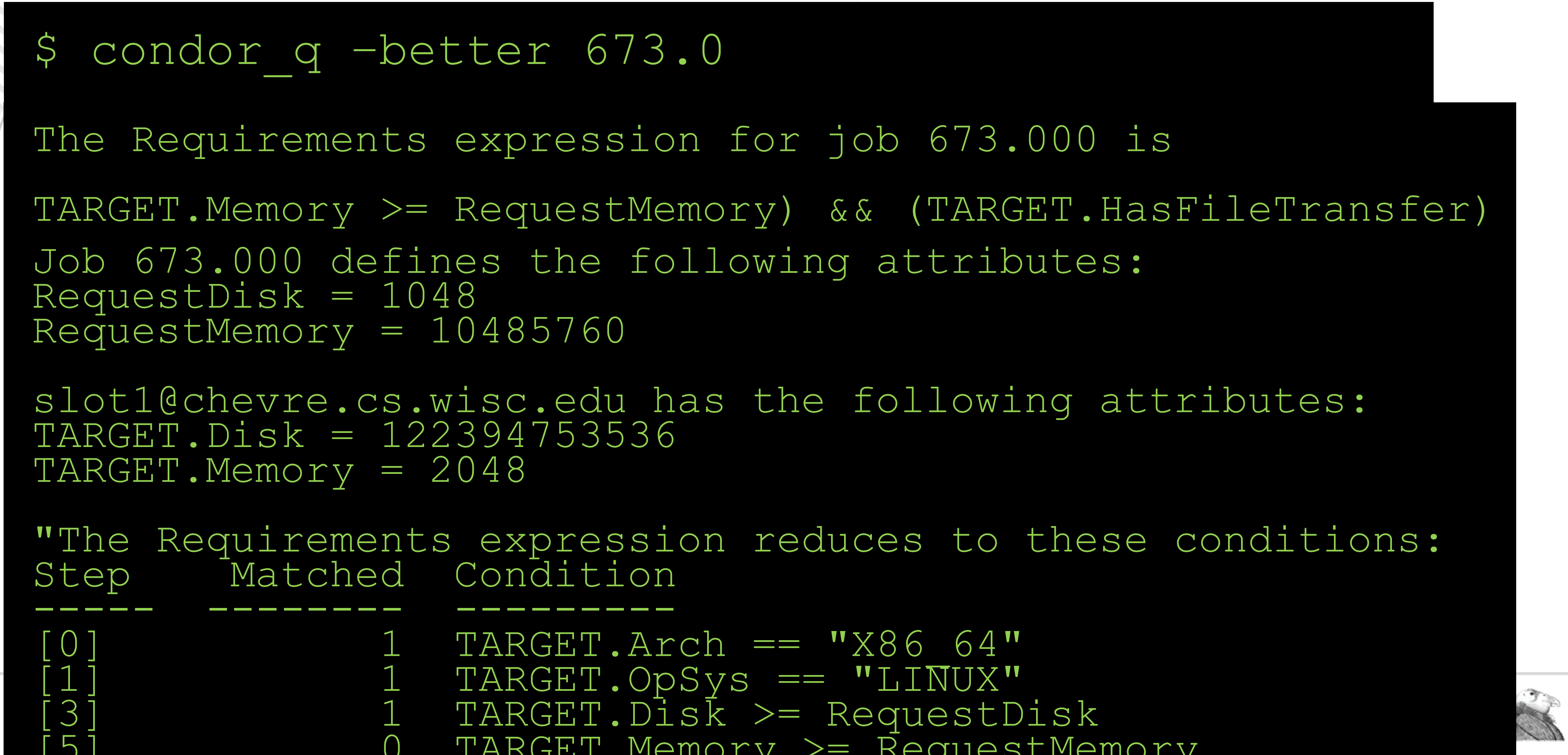
condor_q / condor_history

```
$ condor_q cluster.proc (or username or …)
```

```
$ condor_history cluster.proc (or username or …)
```



condor_q -better



```
$ condor_q -better 673.0

The Requirements expression for job 673.000 is

TARGET.Memory >= RequestMemory) && (TARGET.HasFileTransfer)
Job 673.000 defines the following attributes:
RequestDisk = 1048
RequestMemory = 10485760

slot1@chevre.cs.wisc.edu has the following attributes:
TARGET.Disk = 122394753536
TARGET.Memory = 2048

"The Requirements expression reduces to these conditions:
Step    Matched  Condition
-----  --------  ---------
[0]           1  TARGET.Arch == "X86_64"
[1]           1  TARGET.OpSys == "LINUX"
[3]           1  TARGET.Disk >= RequestDisk
[5]           0  TARGET.Memory >= RequestMemory
```

condor_q querying other schedds

```
$ condor_q -name other_schedd
```

```
$ condor_q -name other_schedd -pool remote_cm
```

```
$ condor_q -jobad file_with_ad
```

condor_q -better

```
$ condor_q -better -jobads job -slotads sl 673.0
```

Analyzes the job in the file "job"

Against the slot in file sl

Useful for "why job doesn't match **this** slot?

And a better condor_q for loops…

```
$ condor_watch_q
```

Instead of watch condor_q

condor_qedit

```
$ condor_qedit 10 'AttributeName = "StringValue" '
$ condor_qedit 10 'AttributeName = True'
$ condor_qedit 10 'AttributeName = 100.3'
$ condor_qedit 10 'AttributeName = someExpr > 100'
```

Quoting is Tricky! Both
classad AND
shell quoting in play!

condor_qedit –edits some_file

```
$ condor_qedit 10 -edits some_file
some_file:
StringAttribute = "StringValue"
NumericAttribute = 10.0
ExprAttribute = foo > 100
```

This avoids shell quoting headaches

But classad quoting still applies – be careful

What's the problem with condor_history?

Don't worry, I'll wait

And wait…

And wait…

Why is condor_history slow?

1. There's rather a lot of information about each job

2. And there's a huge number of jobs

Solving problem #2 first…

- Add -match 1, if you know there's only one job, i.e.

```
$ condor_history match 1 -const "some constraint"
```

Going the right direction helps…

If you suspect the job is in the older half of the history..

```
$ condor_history -forwards -match 1 -const …
```

If you suspect the job is in the newer half of the history..

```
$ condor_history -backwards -match 1 -const …
```

Fastest to solve both problems!

```
$ condor_history -userlog logfile …
```

1. You do have a userlog (job log), don't you
2. Only a handful of attributes are in the logfile
   (which is why it is so fast….)

Or read from a more constrained source…

Works with job eventlog dagman nodes.log or global event..

```
$ condor_userlog filename
```

But the real solution to slow history is…

Use a real database!

condor_adstash

```
$ condor_adstash -standalone -se_host host:port
```

- Tool / daemon

- Loops, copying job data from schedd logs into elastic search
- Only historical jobs (can be from startd also)

- Super useful for debugging, performance, etc.

Commands that only work on already running* jobs

*Not all universes supported

condor_ssh_to_job

```
$ condor_ssh_to_job 673.0
Welcome to slot1@chevre.cs.wisc.edu
$
```

Doesn't work in all environment

Basis of condor_submit -i

condor_ssh_to_job -X

```
$ condor_ssh_to_job -X 673.0
```

Enables forwarding of X protocol

(if ssh'ing to AP, also add –X there)

condor_ssh_to_job –ssh <ssh program> here

Scp to job

```
$ condor_ssh_to_job -ssh scp 673.0 remote:from_ep to_ap
```

"remote:" placeholder for real machine name

Other way works too

```
$ condor_ssh_to_job -ssh scp 673.0 from_ap remote:to_ap
```

Rsync to job over condor_ssh_to_job

```
$ rsync -v -e "condor_ssh_to_job" 673.0:output .
```

"673:" jobid place holder for real machine

condor_tail

```
$ condor_tail 673.0
```

Copies last 1024 (or –maxbytes xx) from stdout

Or -stderr

Or optional filename

```
$ condor_tail -follow 673.0 some_interesting_file
```

condor_evicted_files

```
$ condor_evicted_files list 673.0
$ condor_evicted_files get 673.0 my_output_file
```

Allows inspection and fetching of files
From spool directories of evicted jobs.

Also works with checkpoint files

condor_vacate_job

```
$ condor_vacate_job 673.0
```

Helpful if you don't want to hold the job
But want it to start somewhere else
If worker node is slow or seems broken or…

condor_chirp

- Chirp is the only command that MUST RUN IN THE JOB
  - On the EP!
- Also have python bindings

- Chirp commands run from the EP to the AP

- Many subcommands of chirp, but the big ones are…

Get attribute from AP's copy of job ad

```
$ condor_chirp get_job_attr SomeAttribute
```

Prints to stdout the value of "SomeAttribute"

The job id is always the running job

Set attribute from AP's copy of job ad

```
$ condor_chirp set_job_attr JobAttr '"some_value"'
```

Sets the value of "JobAttr"
The job id is always the running job

CAREFUL WITH QUOTING! (as condor_qedit)

Append to job log

```
$ condor_chirp ulog 'Some message'
```

Appends 'some message' to the job log

Super useful for debugging, timing,
job progress

condor_wait

```
$ condor_wait job_log_file
```

Block until job in job log file exits

Commands for EPs and other HTCondor services

condor_off

```
$ condor_off -startd
$ condor_off -negotiator
$ condor_off -schedd
```

Turns off named condor daemon

```
$ condor_off
```

Turns off every daemon, except master…

```
$ condor_off -master
```

condor_on

```
$ condor_on -startd
$ condor_on -negotiator
$ condor_on -schedd
```

Turns on named condor daemon

```
$ condor_on
```

Turns on every daemon, except master…

condor_restart

```
$ condor_restart -startd
$ condor_restart -negotiator
$ condor_restart -schedd
```

Restarts named condor daemon

```
$ condor_restart
```

Restarts every daemon, except master…

```
$ condor_restart -master
```

draining

```
$ condor_drain machine_or_slot
```

Cancelling draining

```
$ condor_drain -cancel machine_or_slot
```

Immediately running high priority job

```
$ condor_now idle_job_id running_job_id
```

Swaps out the running job for an idle job

  must be from same submitter

Evict running job

```
$ condor_vacate machine_or_slot
```

Much like condor_vacate_job

- Machine owner permission
- Can be just one slow

Job goes to "I"dle

Knob management

```
$ condor_config_val -summary
```

Emits diff from defaults of all knobs (from disk)

```
# from /nobackup/gthain/personal-condor/condor_config.local
#
RunBenchmarks = true
STARTD_DEBUG = D_FULLDEBUG D_COMMAND D_JOB D_MACHINE
STARTER_DEBUG = D_FULLDEBUG D_COMMAND
SHARED_PORT_DEBUG = D_FULLDEBUG D_NETWORK
MEMORY = 819200
```

Knob management

```
$ condor_config_val -verbose SOME_KNOB
```

Emits current values of one knob
AND the file & line number where it is

Knob management

```
$ condor_reconfig
```

Tells HTCondor services to re-read config file

Collector querying

```
$ condor_status
```

Please don't run this…

```
$ condor_status -l | grep some_attribute
```

It works – usually

Did you get the regxep right?

Are you sure?

Is it anchored?

Did you remember case-insensitive attrs?

Can be slow

Rather, run this:

```
$ condor_status -af some_attribute
```

Evalutes by default (against itself, no match)

```
$ condor_status -af WithinResourceLimits
undefined
$ condor_status -af:l WithinResourceLimits
WithinResourceLimits = undefined
$ condor_status -af:r WithinResourceLimits
 (MY.Cpus > 0 && TARGET.RequestCpus <= MY.Cpus &&…
```

Usually naked –af is good enough for literals

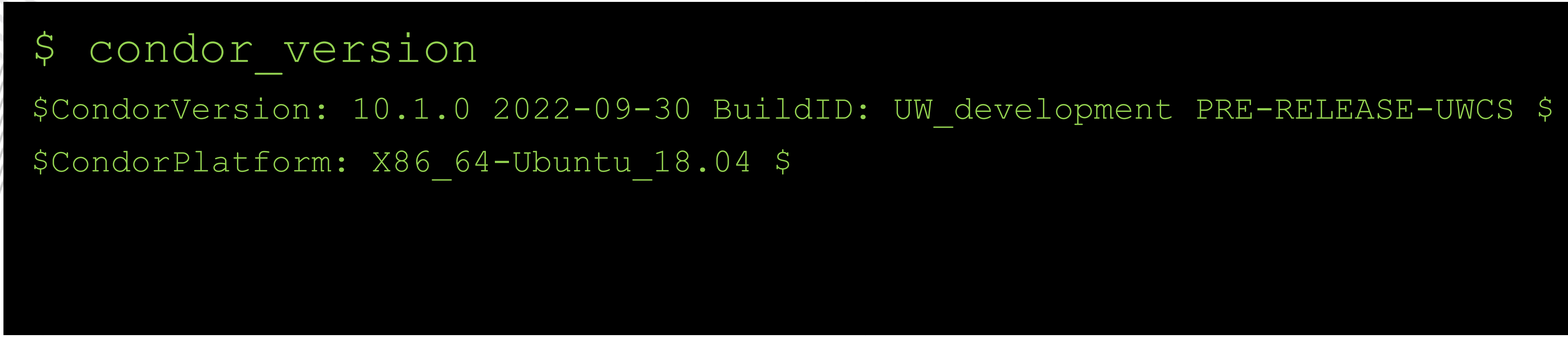
```
$ condor_status -af Name
slot1@chevre
$ condor_status -af:r Name
"slot1@chevre"
$ condor_status -af:lr Name
Name = "slot1@chevre"
```

condor_status – not just for slot ads

```
$ condor_status -submitters
$ condor_status -schedd
$ condor_status -master
$ condor_status -negotiator
```

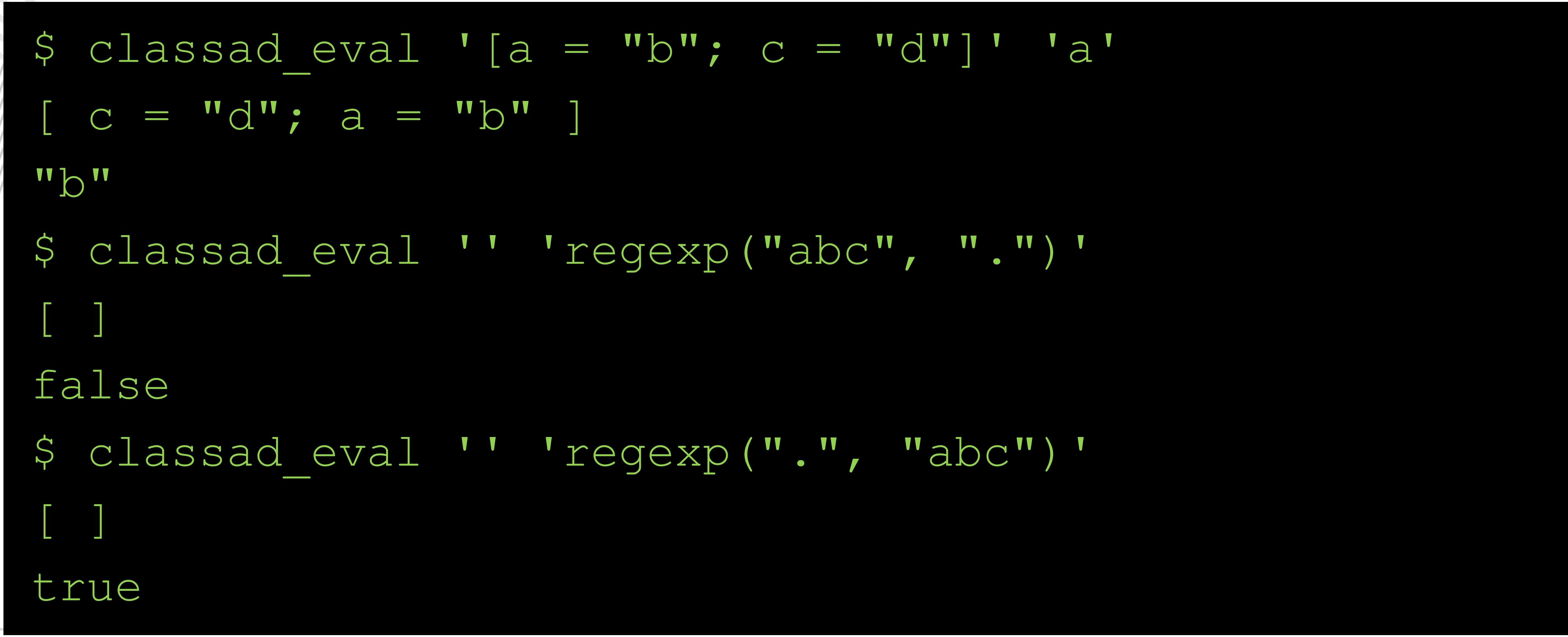
Commands for debugging and testing HTCondor services

Version



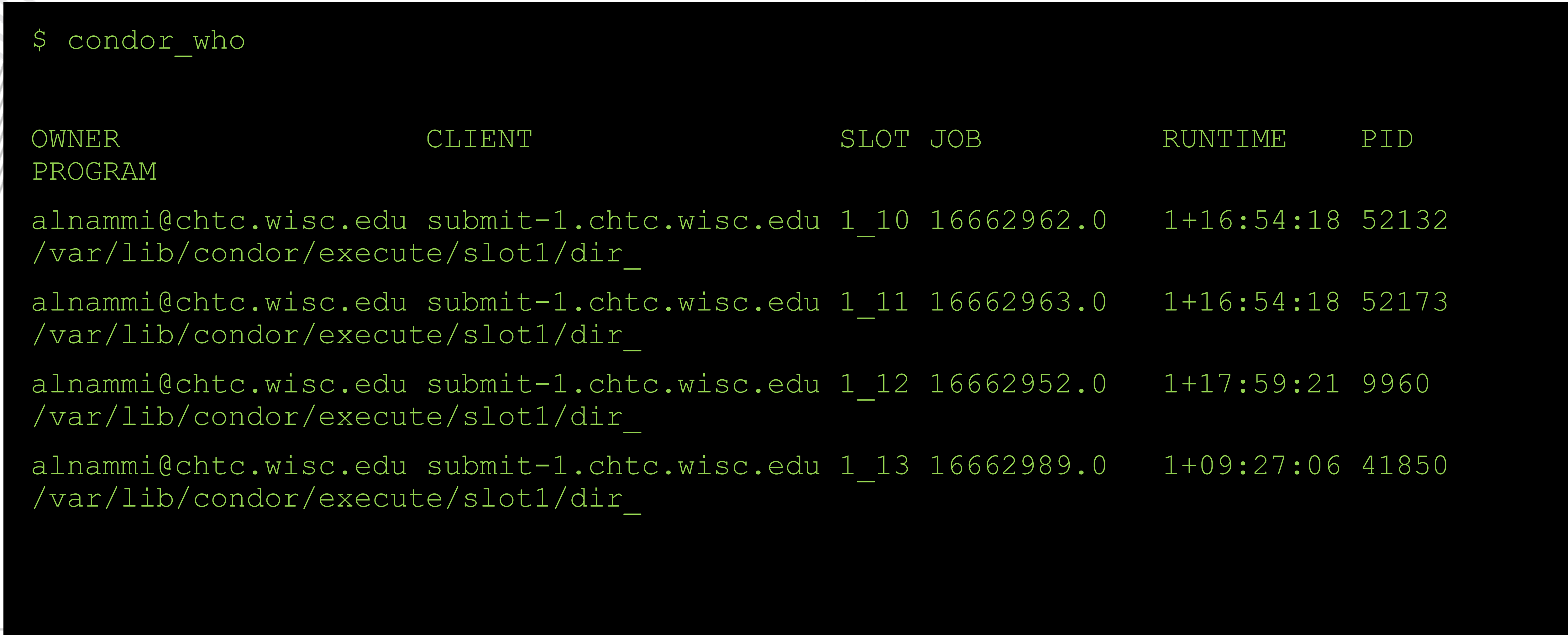
```
$ condor_version
$CondorVersion: 10.1.0 2022-09-30 BuildID: UW_development PRE-RELEASE-UWCS $
$CondorPlatform: X86_64-Ubuntu_18.04 $
```

ClassAd Debugging



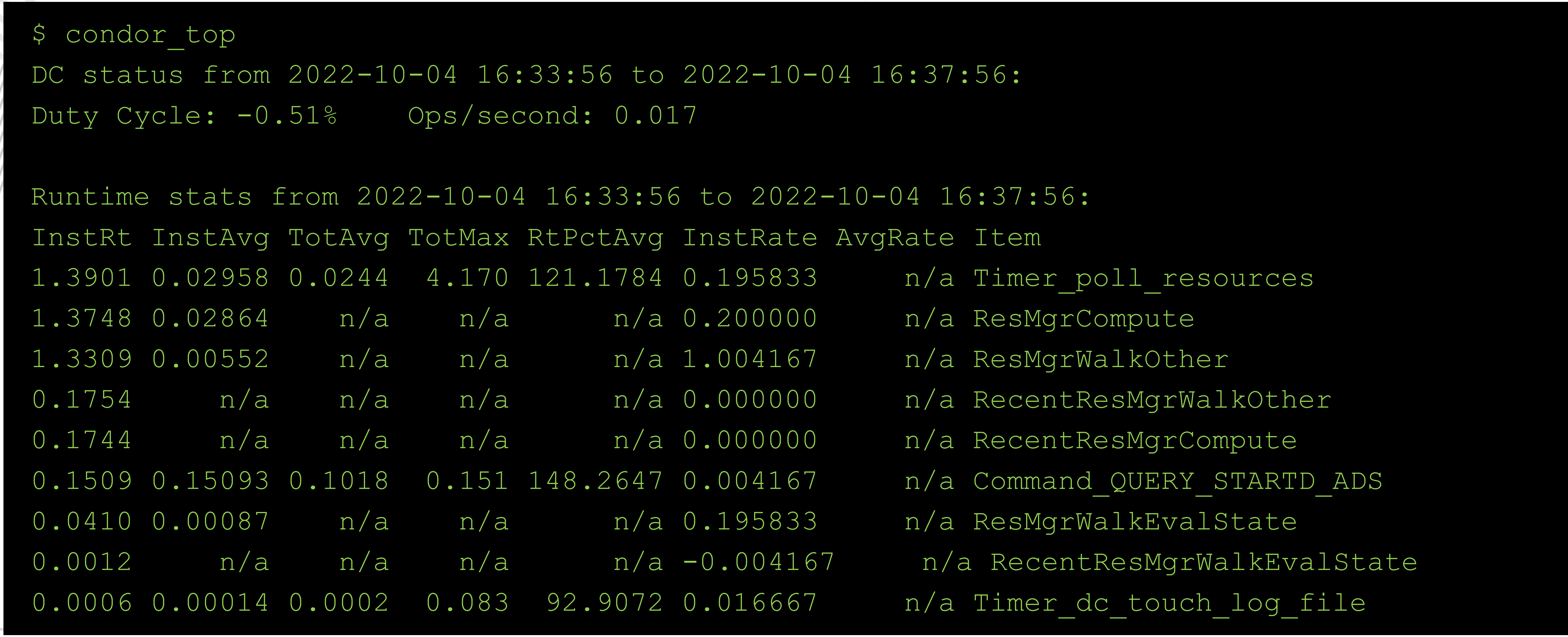
```
$ classad_eval '[a = "b"; c = "d"]' 'a'
[ c = "d"; a = "b" ]
"b"
$ classad_eval '' 'regexp("abc", ".")'
[ ]
false
$ classad_eval '' 'regexp(".", "abc")'
[ ]
true
```

What's running on my EP?



```
$ condor_who


OWNER                 CLIENT                 SLOT JOB            RUNTIME    PID
PROGRAM
alnammi@chtc.wisc.edu submit-1.chtc.wisc.edu 1_10 16662962.0   1+16:54:18 52132
/var/lib/condor/execute/slot1/dir_
alnammi@chtc.wisc.edu submit-1.chtc.wisc.edu 1_11 16662963.0   1+16:54:18 52173
/var/lib/condor/execute/slot1/dir_
alnammi@chtc.wisc.edu submit-1.chtc.wisc.edu 1_12 16662952.0   1+17:59:21 9960
/var/lib/condor/execute/slot1/dir_
alnammi@chtc.wisc.edu submit-1.chtc.wisc.edu 1_13 16662989.0   1+09:27:06 41850
/var/lib/condor/execute/slot1/dir_
```

Why is HTCondor slow



```
$ condor_top
DC status from 2022-10-04 16:33:56 to 2022-10-04 16:37:56:
Duty Cycle: -0.51%    Ops/second: 0.017

Runtime stats from 2022-10-04 16:33:56 to 2022-10-04 16:37:56:
InstRt InstAvg TotAvg TotMax RtPctAvg InstRate AvgRate Item
1.3901 0.02958 0.0244  4.170 121.1784 0.195833     n/a Timer_poll_resources
1.3748 0.02864    n/a    n/a      n/a 0.200000     n/a ResMgrCompute
1.3309 0.00552    n/a    n/a      n/a 1.004167     n/a ResMgrWalkOther
0.1754     n/a    n/a    n/a      n/a 0.000000     n/a RecentResMgrWalkOther
0.1744     n/a    n/a    n/a      n/a 0.000000     n/a RecentResMgrCompute
0.1509 0.15093 0.1018  0.151 148.2647 0.004167     n/a Command_QUERY_STARTD_ADS
0.0410 0.00087    n/a    n/a      n/a 0.195833     n/a ResMgrWalkEvalState
0.0012     n/a    n/a    n/a      n/a -0.004167     n/a RecentResMgrWalkEvalState
0.0006 0.00014 0.0002  0.083  92.9072 0.016667     n/a Timer_dc_touch_log_file
```

Remote fetching of HTCondor debug log

```
$ condor_fetchlog machineName startd
$ condor_fetchlog chevre STARTER.slot1_3
```

```
10/08/22 18:13:17 Now in new log file /nobackup/gthain/personal-
condor/log/StartLog
10/08/22 18:13:17 Publishing ClassAd 'mips' to slot1 [InSlotList matches]
10/08/22 18:13:17 Publishing ClassAd 'kflops' to slot1 [InSlotList matches]
10/08/22 18:13:17 Publishing ClassAd 'mips' to slot1 [InSlotList matches]
10/08/22 18:13:17 Trying to update collector <128.105.14.141:4210>
10/08/22 18:13:17 Attempting to send update via TCP to collector
chevre.cs.wisc.edu <128.105.14.141:4210>
10/08/22 18:13:17 slot1: Sent update to 1 collector(s)
10/08/22 18:13:37 Swap space: 24358084
```

Debugging schedd job transforms

```
$ condor_transform_ads -rules my_rules -in my_input
```

```
# Output:
DiskUsage = 2500000
Err = "/dev/null"
MemoryUsage = 5
NumCheckPoints = 0
RequestDisk = ( 5000000 / 1024 )
ResidentSetSize = 500
```

Wait, What's a Schedd job Transform?

https://research.cs.wisc.edu/htcondor/HTCondorWeek2017/presentations/TueKnoeller_Schedd.pdf

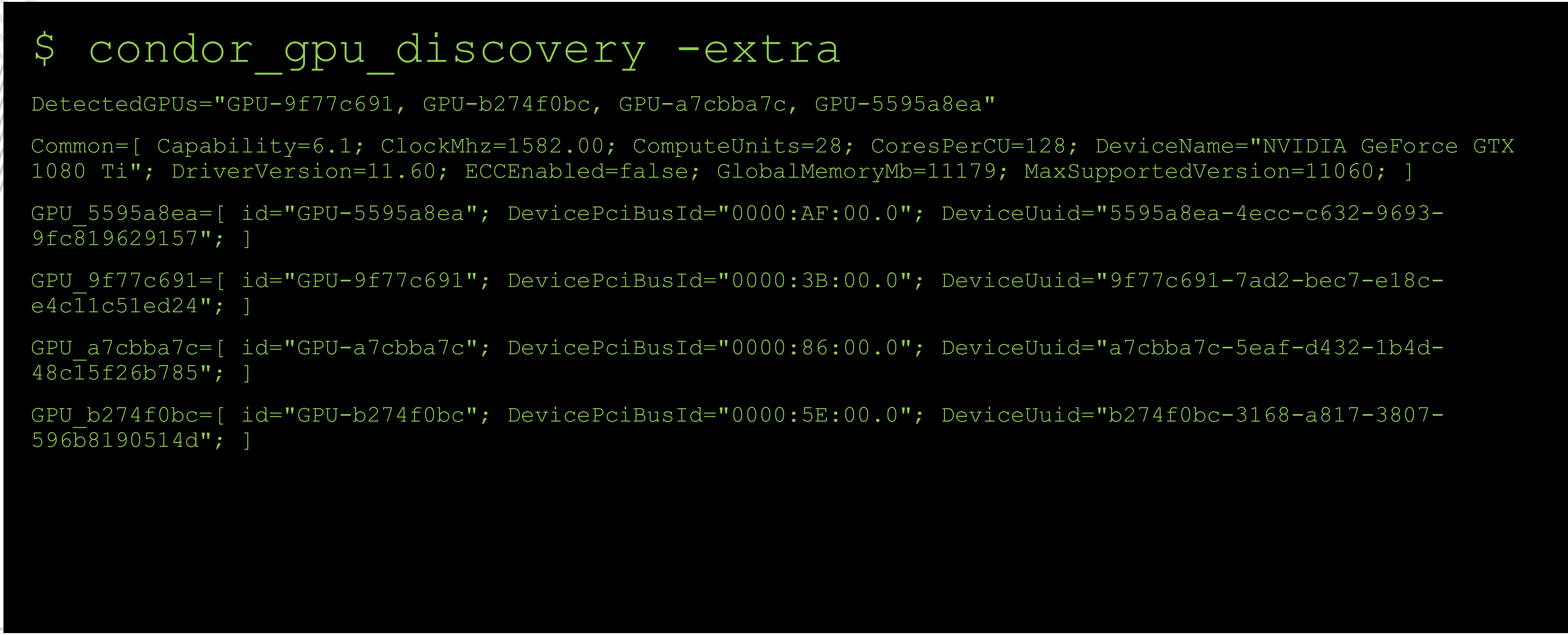
https://htcondor.readthedocs.io/en/latest/classads/transforms.html#classad-transforms

And, three commands to query machine

EP attributes

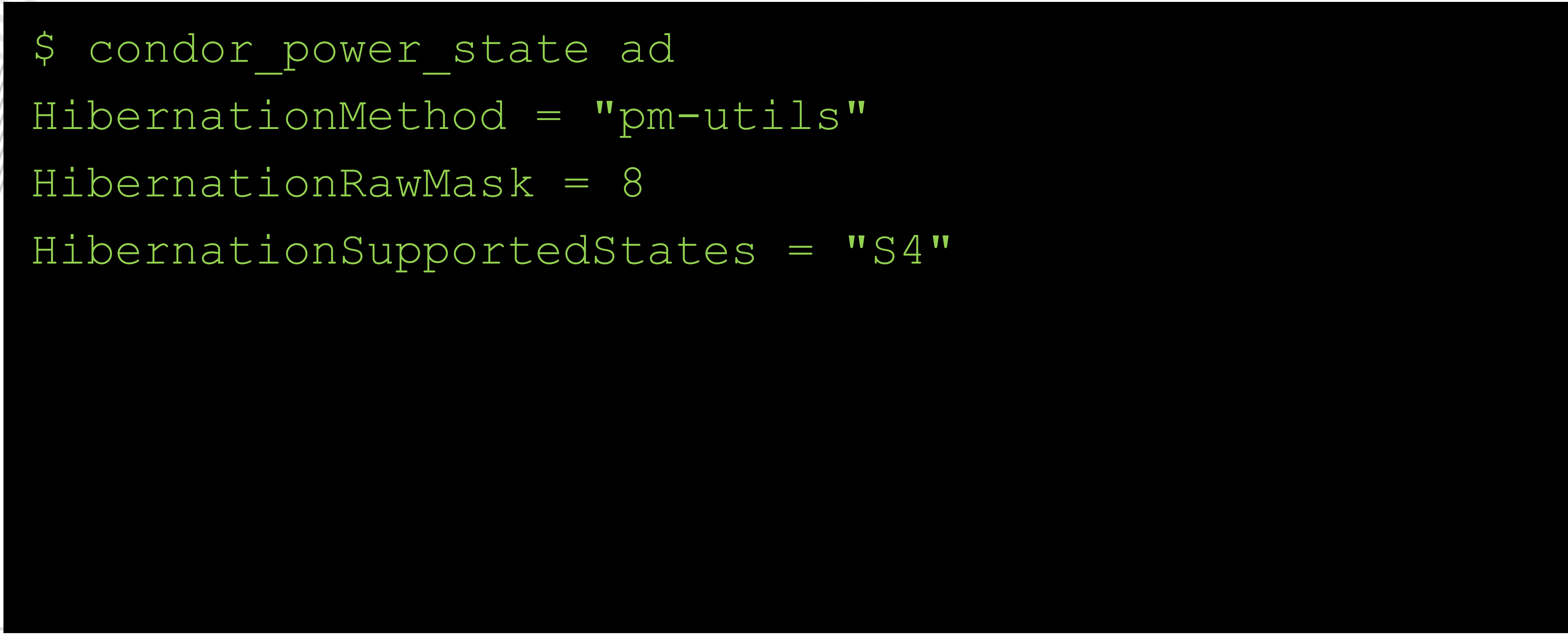
```
$ condor_starter -classad
CondorVersion = "$CondorVersion: 9.12.0 2022-09-13 BuildID: 605774 PackageID: 9.12.0-0.605774 RC $"
IsDaemonCore = True
HasFileTransfer = True
HasJobTransferPlugins = True
HasPerFileEncryption = True
HasReconnect = True
HasMPI = True
HasTDP = True
HasJobDeferral = True
..
```

GPU attributes



```
$ condor_gpu_discovery -extra

DetectedGPUs="GPU-9f77c691, GPU-b274f0bc, GPU-a7cbba7c, GPU-5595a8ea"

Common=[ Capability=6.1; ClockMhz=1582.00; ComputeUnits=28; CoresPerCU=128; DeviceName="NVIDIA GeForce GTX
1080 Ti"; DriverVersion=11.60; ECCEnabled=false; GlobalMemoryMb=11179; MaxSupportedVersion=11060; ]

GPU_5595a8ea=[ id="GPU-5595a8ea"; DevicePciBusId="0000:AF:00.0"; DeviceUuid="5595a8ea-4ecc-c632-9693-
9fc819629157"; ]

GPU_9f77c691=[ id="GPU-9f77c691"; DevicePciBusId="0000:3B:00.0"; DeviceUuid="9f77c691-7ad2-bec7-e18c-
e4c11c51ed24"; ]

GPU_a7cbba7c=[ id="GPU-a7cbba7c"; DevicePciBusId="0000:86:00.0"; DeviceUuid="a7cbba7c-5eaf-d432-1b4d-
48c15f26b785"; ]

GPU_b274f0bc=[ id="GPU-b274f0bc"; DevicePciBusId="0000:5E:00.0"; DeviceUuid="b274f0bc-3168-a817-3807-
596b8190514d"; ]
```

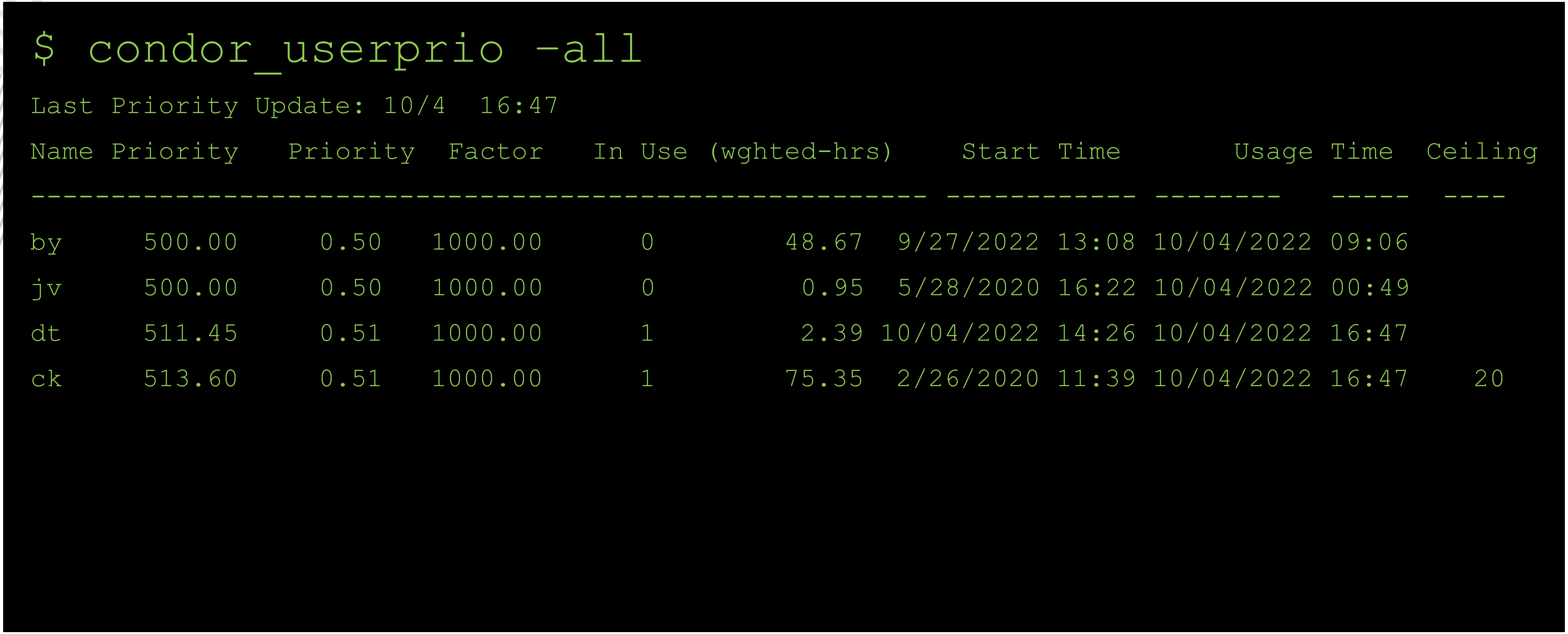
Power Saving detection



```
$ condor_power_state ad
HibernationMethod = "pm-utils"
HibernationRawMask = 8
HibernationSupportedStates = "S4"
```

Commands for submitters

condor_userprio



```
$ condor_userprio -all
Last Priority Update: 10/4  16:47
Name Priority   Priority  Factor   In Use (wghted-hrs)    Start Time       Usage Time   Ceiling
-------------------------------------------------------- ------------ --------   -----  ----
by     500.00     0.50   1000.00      0        48.67  9/27/2022 13:08 10/04/2022 09:06
jv     500.00     0.50   1000.00      0         0.95  5/28/2020 16:22 10/04/2022 00:49
dt     511.45     0.51   1000.00      1         2.39 10/04/2022 14:26 10/04/2022 16:47
ck     513.60     0.51   1000.00      1        75.35  2/26/2020 11:39 10/04/2022 16:47    20
```

Whole talk about prios, userprio, etc.

- https://youtu.be/aPUztmJ0n2s

condor_userprio –setceil

```
$ condor_userprio -setceil user ceiling
```

Imposes a per-pool upper bound on Cores a given submitter can get

condor_userprio –setfloor

```
$ condor_userprio -setfloor user ceiling
```

Imposes a per-pool lower bound on Cores a given submitter can get

condor_qusers

```
$ condor_quser -disable tannenba
```

Bans a user from submitting jobs

Shows usage without options

Also has a -enable

The one new command…

htcondor <noun> <verb>

- A fresh start
- Written in python
- Influenced by git, k8s, etc.
- Extensible!

- E.g. "htcondor job status"
- "htcondor eventlog read"

- We're looking for feedback – how can we improve tools?

Thank you and questions

Thank you – Questions?

This work is supported by the NSF under Cooperative Agreement OAC-2030508. Any options, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.