HTC From a Users Perspective

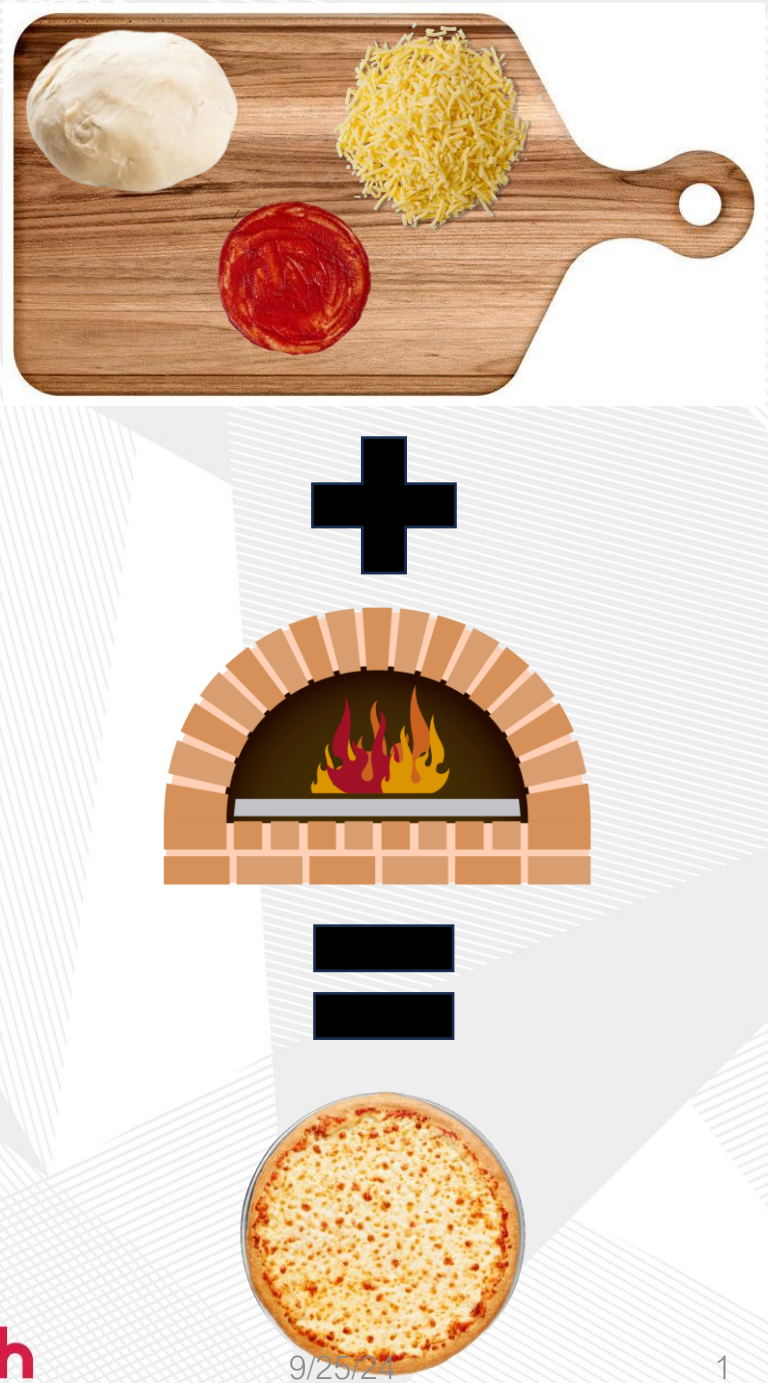
By: Cole Bollig

Software Developer for CHTC

European HTCondor Workshop 2024

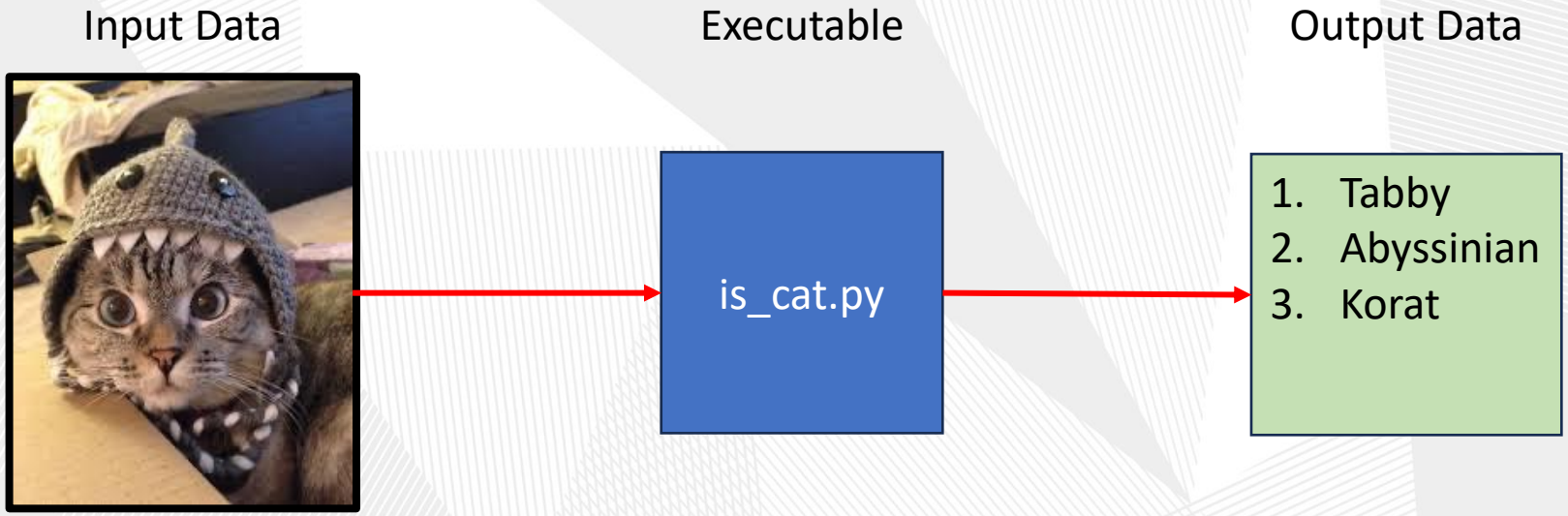
HTCondor Jobs

- HTCondor schedules and executes Jobs
- What is a Job?
  - A ‘Job’ is a single computational task
    1. Input Data
    2. Executable (program)
    3. Output Data
- Executable must be runnable from the command line without any interactive input



Example Job

- I wrote a program to check if a picture is of a cat. If the picture is a cat, then the program will create a list of possible breeds. Otherwise, the program fails.



Describing an HTCondor Job

```
executable = is_cat.py
arguments = cat.img

transfer_input_files = cat.img
transfer_output_files = potential-breeds.txt
should_transfer_files = Yes
when_to_transfer_output = ON_SUCCESS

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 50MB
request_memory = 20MB

queue 1
```

CAT-Analysis.sub

Describe an HTCondor job with the Job Description Language (JDL)

Describing an HTCondor Job

```
executable = is_cat.py
arguments = cat.img

transfer_input_files = cat.img
transfer_output_files = potential-breeds.txt
should_transfer_files = Yes
when_to_transfer_output = ON_SUCCESS

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 50MB
request_memory = 20MB

queue 1
```

CAT-Analysis.sub

List the executable and arguments for HTCondor to run at the EP like how you would run the executable by hand

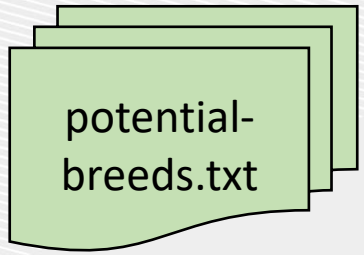
```
$ is_cat.py cat.img
```

Describing an HTCondor Job

```
executable = is_cat.py
arguments = cat.img

transfer_input_files = cat.img
transfer_output_files = potential-breeds.txt
should_transfer_files = Yes
when_to_transfer_output = ON_SUCCESS

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 50MB
request_memory = 20MB

queue 1
```

CAT-Analysis.sub

Specify input data that the job requires and output data we are expecting to be created (and transferred back)



Describing an HTCondor Job

```
executable = is_cat.py
arguments = cat.img

transfer_input_files = cat.img
transfer_output_files = potential-breeds.txt
should_transfer_files = Yes
when_to_transfer_output = ON_SUCCESS

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 50MB
request_memory = 20MB

queue 1
```

CAT-Analysis.sub

- **log -** Generated by HTCondor to track job progress from key events in the jobs lifetime
- **output -** File that captures the jobs STDOUT
- **error -** File that captures the jobs STDERR

Describing an HTCondor Job

```
executable = is_cat.py
arguments = cat.img

transfer_input_files = cat.img
transfer_output_files = potential-breeds.txt
should_transfer_files = Yes
when_to_transfer_output = ON_SUCCESS

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_disk = 50MB
request_memory = 20MB

queue 1
```

CAT-Analysis.sub

- Request the appropriate resources for your job to run.
- **queue -** keyword indicating "create a job."

Resource Requirements

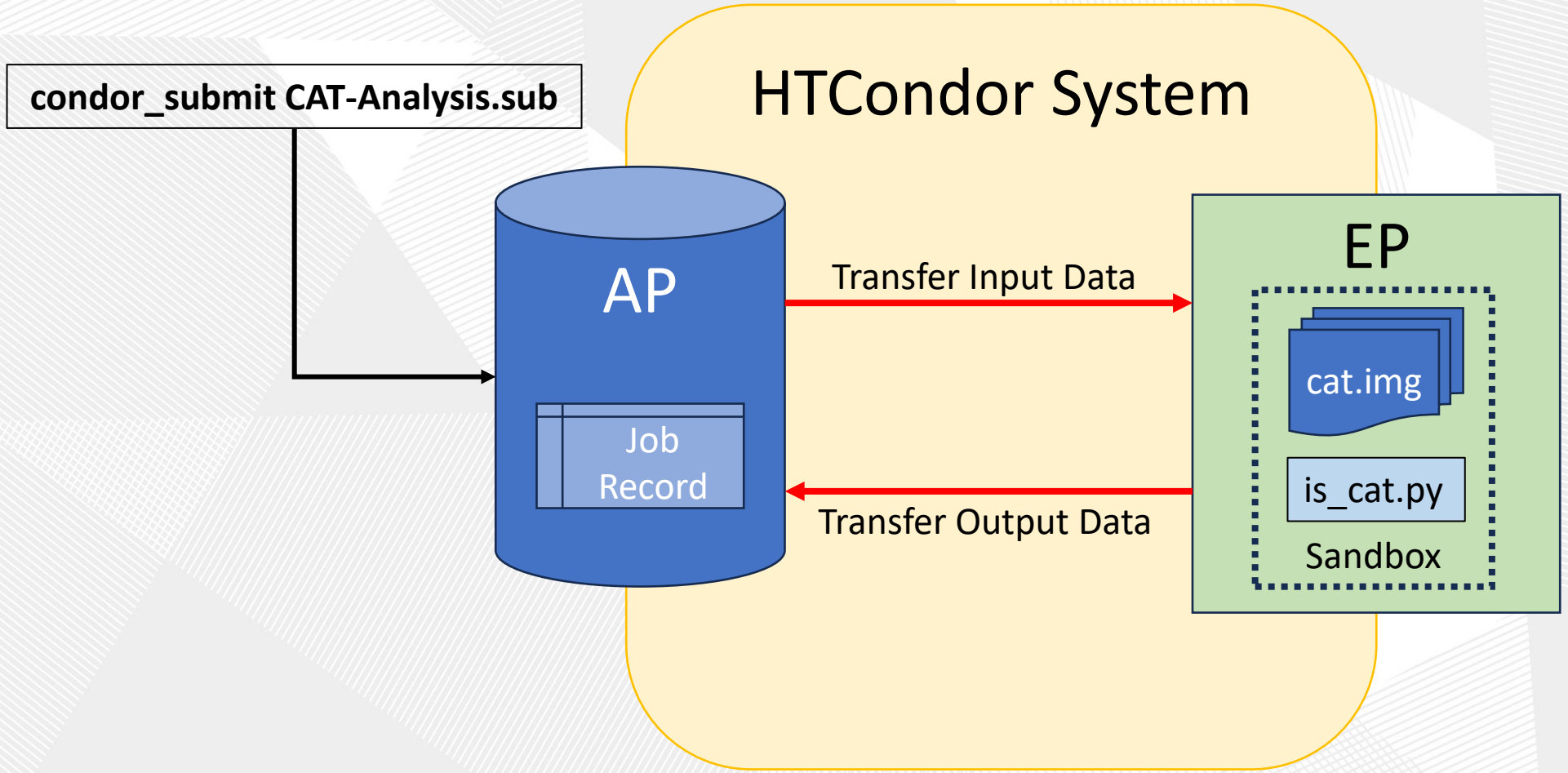
- Jobs are nearly always using a part of a computer, not the whole thing. EP divides worker node into execute "Slots".
- Very important to request appropriate resources (memory, cpus, disk) for a job
- Job will likely be enforced to only use what it requests

Place the Job to an Access Point

- Simply run *condor_submit <Submit File>*

```
colebollig@Coles-MacBook-Pro sleep % condor_submit CAT-Analysis.sub
Submitting job(s).
1 job(s) submitted to cluster 470.
colebollig@Coles-MacBook-Pro sleep %
```

What happens?



Monitor Jobs

- Use *condor_q* to get job record information from the AP
- Use *condor_watch_q* to actively track a jobs state

```
colebollig@Coles-MacBook-Pro sleep % condor_q
-- Schedd: COLES_AP@ : <127.0.0.1:49876?... @ 09/25/24 00:51:57
OWNER      BATCH_NAME    SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
colebollig ID: 470      9/25 00:51      _      1      _      1 470.0

Total for query: 1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
Total for all users: 1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended


colebollig@Coles-MacBook-Pro sleep % condor_watch_q
BATCH     IDLE  RUN  DONE  TOTAL  JOB_IDS
ID: 470      -    1    -    1   37.0   [==============================================]

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

Total: 6 jobs; 5 completed, 1 running

Updated at 2024-09-25 00:54:15
Input ^C to exit
```

Monitor Jobs

- Get human readable job status via *htcondor job status*

```
$ htcondor job status 123.45

Job 123.45 is currently running on host exec221.chtc.wisc.edu.
It started running again 2.1 hours ago.
It was submitted 3.6 hours ago.
Its current memory usage is 2.5 GB out of 4.0 GB requested.
Its current disk usage is 3.8 GB out of 5.5 GB requested.
It has restarted 2 times.
Goodput is 80% (0.5 hours badput, 2.1 hours goodput).
```

```
[cabollig@ap2002 ~]$ htcondor ap status
        Access Point          Health
ap2002.chtc.wisc.edu          Good
ap2004.chtc.wisc.edu          Good
```

Also, checkout AP health via *htcondor ap status*

Monitor Jobs

- Use *condor_history* to view completed job records from the AP

```
colebollig@Coles-MacBook-Pro sleep % condor_history -limit 10
 ID     OWNER          SUBMITTED   RUN_TIME     ST COMPLETED   CMD
37.0    colebollig     9/25 00:54   0+00:01:00 C   9/25 00:55 /bin/sleep 60
34.0    colebollig     9/18 19:25   0+00:01:38 X              /Users/colebollig/Desktop/...
36.0    colebollig     9/18 19:25   0+00:01:16 X              /bin/sleep 1000
35.3    colebollig     9/18 19:25              X              /bin/sleep 1000
35.2    colebollig     9/18 19:25              X              /bin/sleep 1000
35.1    colebollig     9/18 19:25              X              /bin/sleep 1000
35.0    colebollig     9/18 19:25              X              /bin/sleep 1000
31.0    colebollig     9/18 19:23   0+00:00:56 X              /Users/colebollig/Desktop/...
33.0    colebollig     9/18 19:24   0+00:00:13 X              /bin/sleep 1000
32.1    colebollig     9/18 19:23   0+00:00:15 X              /bin/sleep 1000
```

Other Useful Tools

• condor_status	View Ads in the Collector (e.g. EP Slots)
• condor_q	View Jobs at an AP
• condor_q -analyze	Why job/machines fail to match?
• condor_ssh_to_job	Create ssh session to active job
• condor_submit -i	Submit interactive job
• condor_hold / release	Hold a job, or release a held job
• condor_run	Submit and block
• condor_rm	Remove Jobs
• condor_prio	Intra-User Job Prios
• condor_history	Completed Job Info
• condor_submit_dag	Submit new DAG workflow
• condor_chirp	Access files/ad from active job

Details of the Job Event Log

```
000 (128.000.000) 05/09 11:09:08 Job submitted from host:
<128.104.101.92&sock=6423_b881_3>
...
001 (128.000.000) 05/09 11:10:46 Job executing on host:
<128.104.101.128:9618&sock=5053_3126_3>
...
006 (128.000.000) 05/09 11:10:54 Image size of job updated: 220
    1  -  MemoryUsage of job (MB)
    220  -  ResidentSetSize of job (KB)
...
005 (128.000.000) 05/09 11:12:48 Job terminated.
    (1) Normal termination (return value 0)
        Usr 0 00:00:00, Sys 0 00:00:00  -  Run Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00  -  Run Local Usage
        Usr 0 00:00:00, Sys 0 00:00:00  -  Total Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00  -  Total Local Usage
    0  -  Run Bytes Sent By Job
    33  -  Run Bytes Received By Job
    0  -  Total Bytes Sent By Job
    33  -  Total Bytes Received By Job
    Partitionable Resources :    Usage  Request Allocated
       Cpus                 :        1        1         1
       Disk (KB)            :       14    20480  17203728
       Memory (MB)          :        1       20        20
```

Submitting Lists of Jobs

```
executable = is_cat.py
arguments = $(image)

transfer_input_files = $(image)
transfer_output_files = $(image).breeds
should_transfer_files = Yes
when_to_transfer_output = ON_SUCCESS

log = job.log
output = out/job-$(Process).out
error = error/job-$(Process). err

request_cpus = 1
request_disk = 50MB
request_memory = 20MB

queue image matching images/*.img
```

CAT-Analysis.sub

- ”Nobody is in the business of submitting one job” – Miron

Working Directory/

CAT-Analysis.sub	**images/**	**out/**	**error/**
is_cat.py	cat.img	job-0.out	job-0.err
job.log	dog.img	job-1.out	job-1.err
cat.img.breeds	zoo.img	job-2.out	job-2.err
	…	job-N.out	job-N.err

Possible Queue Statements

matching ... pattern	`queue image matching images/*.img`
in ... list	`queue image in (cat.img dog.img zoo.img)`
from ... file	`queue image from picture_list.txt`

```
cat.img
dog.img
zoo.img
```

picture_list.txt

Late Materialization

- Submit millions of jobs without taking down the AP
- AP creates a job factory to materialize jobs as needed

```
executable = is_cat.py
arguments = $(image)

…

max_materialize = 500
max_idle = 100

queue 1000000
```

CAT-Analysis.sub

condor_submit –factory <Submit File>

Python API

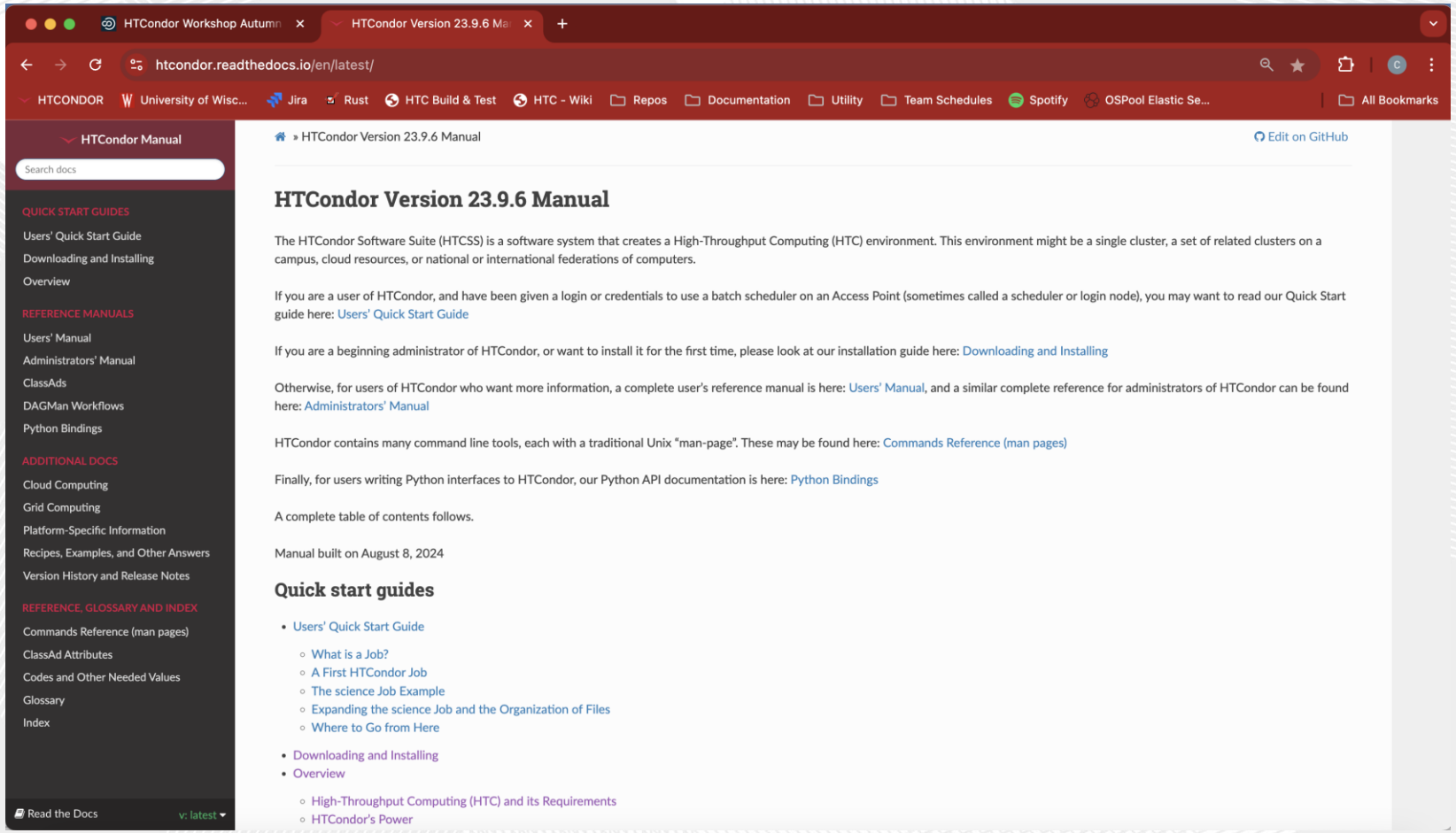
- HTCondor has a built in Python API
  - Simply import the htcondor python module
- Can do what most of the tools can do from the comfort of Python
  - Place jobs
  - Query an AP
  - Query for available capacity (Slot Ads)
  - And more…
- HTCondor Python API

hist.py — executables

```python
#!/usr/bin/env python3

import htcondor
import htcondor2


print("Schedd Job History:")
for ad in htcondor.Schedd().history(None,["ClusterId","ProcId"],match=1):
    print(ad)

print("Epoch History:")
for ad in htcondor.Schedd().jobEpochHistory(None,["ClusterId","ProcId"],match=1):
    print(ad)

print("Epoch history w/ type:")
for ad in htcondor.Schedd().jobEpochHistory(None,[],match=1,ad_type="transfer"):
    print(ad)

print("Startd history:")
for ad in htcondor.Startd().history(None,["ClusterId","ProcId"],match=1):
    print(ad)

print("============htcondor2===============")

print("Schedd Job History:")
for ad in htcondor2.Schedd().history(None,["ClusterId","ProcId"],match=1):
        print(ad)

print("Epoch History:")
for ad in htcondor2.Schedd().jobEpochHistory(None,["ClusterId","ProcId"],match=1):
        print(ad)

print("Epoch history w/ type:")
for ad in htcondor2.Schedd().jobEpochHistory(None,[],match=1,ad_type="transfer"):
        print(ad)

print("Epoch history w/ list[type]:")
for ad in htcondor2.Schedd().jobEpochHistory(None,[],match=1,ad_type=["output"]):
    print(ad)

print("Startd history:")
for ad in htcondor2.Startd().history(None,["ClusterId","ProcId"],match=1):
        print(ad)

print("Epoch None")
for ad in htcondor2.Schedd().jobEpochHistory():
    print(ad)
    break

```

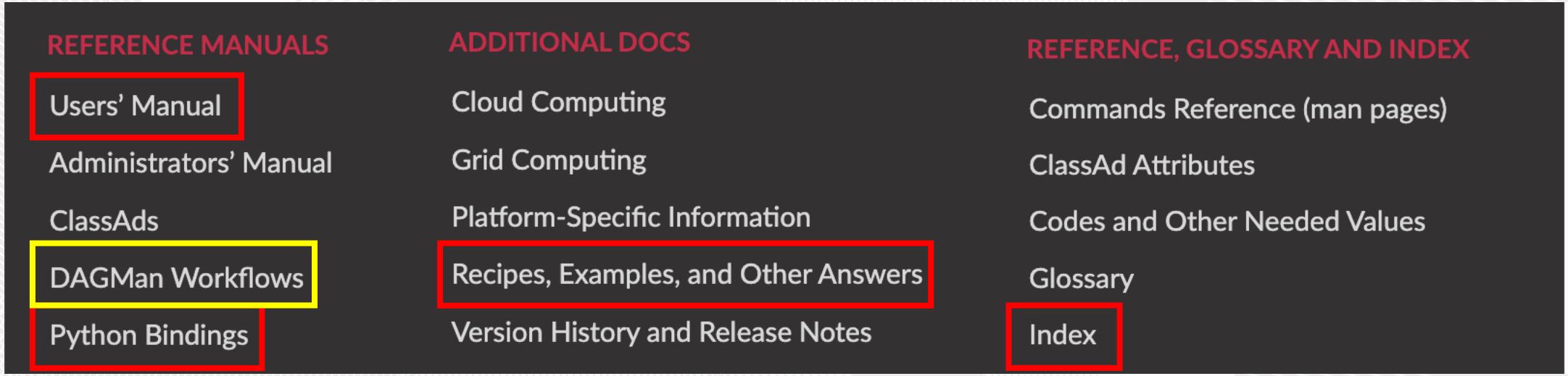
Documentation

https://htcondor.readthedocs.io/en/latest/



Documentation Navigation

https://htcondor.readthedocs.io/en/latest/



Questions?