

# Placing a List of Jobs (Work in Progress)

Miron Livny

John P. Morgridge Professor of Computer Science

Vials Research Professor

Director of CHTC

Technical Director of OSG



- In your working directory/folder you have three images – **ImOne**, **ImTwo** and **ImThree**
- In your working directory/folder you have an application named **IsItACat** that:
  - Requires **one core**, **2 GB** of Memory and **2 GB** of Disk Space to execute
  - Expects the first command line variable to be the name of a file in its execution directory - **Image**
  - Predicts (performs and inference) whether the image in the file is of a cat
  - Creates a file named **IsIt.Image** and writes the results of the predication to this file
  - Reports errors to the “**Standard Error**” file

- You have a (to-do) **Job List** –
  - Job # 0 – `Islt.ImOne = IsltACat(ImOne)`
  - Job # 1 – `Islt.ImTwo = IsltACat(ImTwo)`
  - Job # 2 – `Islt.ImThre = IsltACat(ImThre)`
- ❖ All Jobs in the list use one core and require 2GB of Memory and 2GB of Disk Storage

# Your JobList as a table

JobN	Image	Predication	Error	App	Core	Memory	DiskSpace
0	ImOne	Islt.ImOne	0.Err	IsltACat	1	2GB	2GB
1	ImTwo	Islt.ImTwo	1.Err	IsitACat	1	2GB	2GB
2	ImThre e	Islt.ImThree	2.Err	IsltACat	1	2GB	2GB

# Step I – Place the **Job List** at your\* HTCondor **Access Point**

- ❖ From this point on I assume that “your working environment” is configured to work with one specific HTCondor Access Point. This Access Point can Read/Write to your working Directory.

**You must trust the Access Point!**

Create a **Job Table** File (named **Images.tbl**) with the Columns in the To Do table excluding columns that are the same for all jobs (Prediction, App, Core, Memory, DiskSpace) or are controlled by the application (Predication)

<b>JOB</b>	<b>IMAGE</b>	<b>STERR</b>
<b>0</b>	<b>ImOne</b>	<b>0.Err</b>
<b>1.</b>	<b>ImTwo</b>	<b>1.Err</b>
<b>2</b>	<b>ImThree</b>	<b>2.Err</b>

# Create the following HTCondor Job Template (**template.sub**)

```
Request_cpus      = 1
Request_memory    = 2 GB
Request_disk      = 2 GB

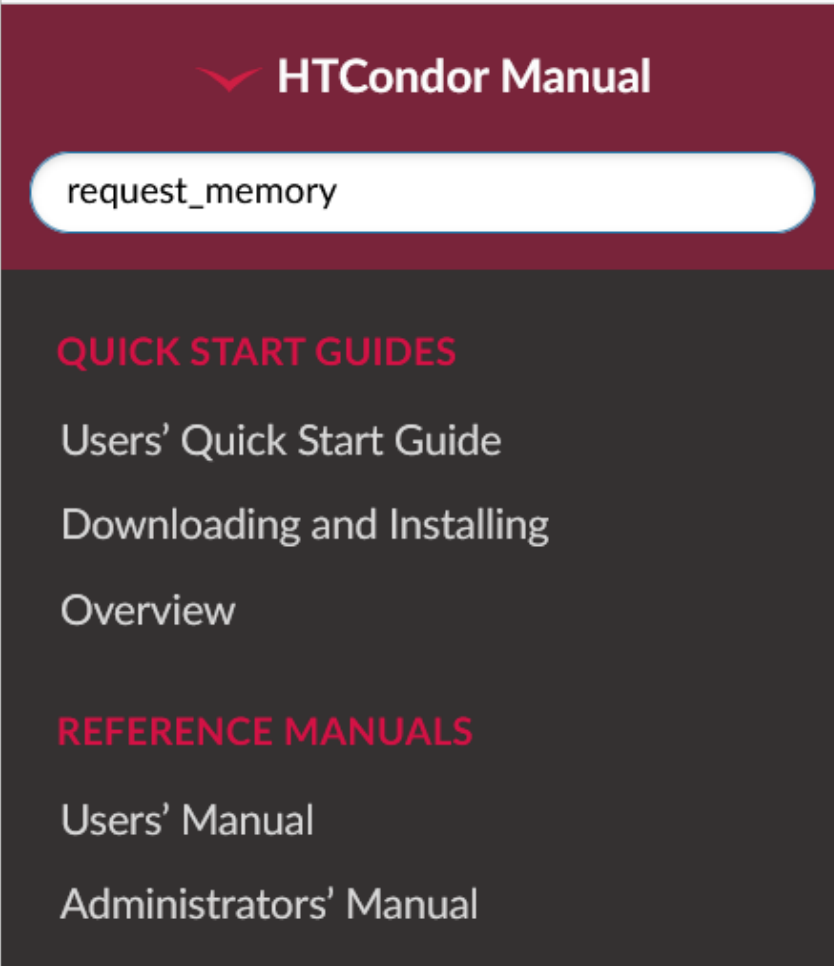
Executable = IsItACat
Arguments   = $(IMAGE)

Transfer_input_files = $(IMAGE)

Error = $(STERR)

Log = $(JOBNAME).log
```

<https://htcondor.readthedocs.io/en/latest/>  
(<https://htcondor.org/htcondor/documentation/>)



HTCondor Manual

request\_memory

**QUICK START GUIDES**

- Users' Quick Start Guide
- Downloading and Installing
- Overview

**REFERENCE MANUALS**

- Users' Manual
- Administrators' Manual



## Place\* the Job List to your Access Point

```
> htcondor jobs place template.sub -table  
images.tbl
```

## HTCondor Access Point responds

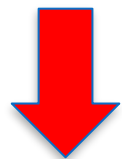
Access Point named **Nikhef2** recorded a  
joblist containing 3 jobs as **Placement  
416638**.

**\* Disclaimer – Work in progress. Do not use it at  
home (yet).**

Images.tbl

Template.sub

Place  
List  
of Jobs



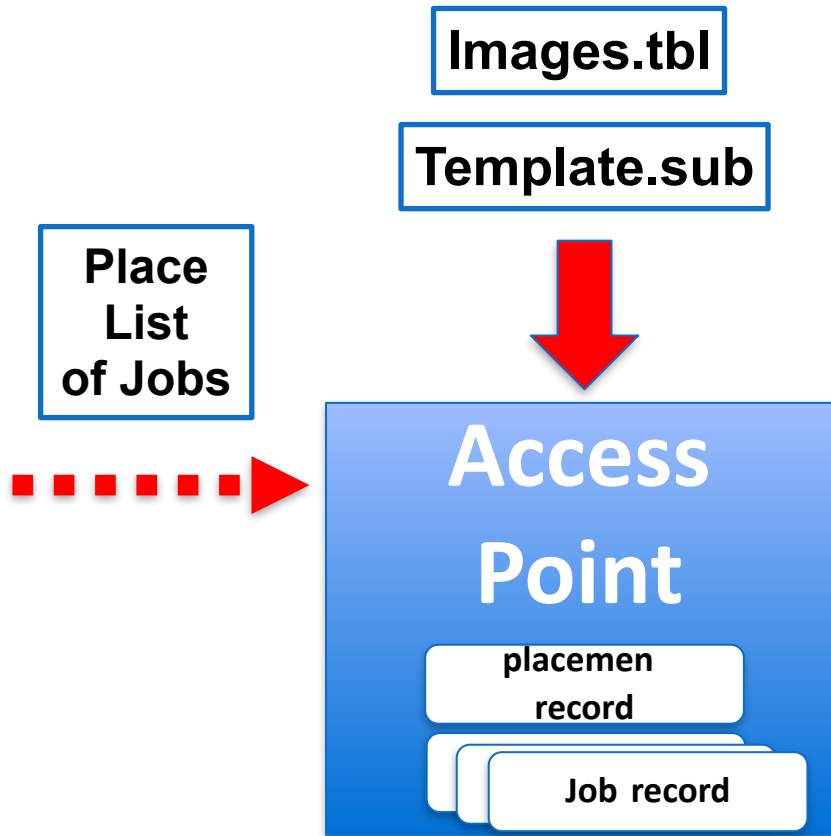
Access  
Point

placemen  
record

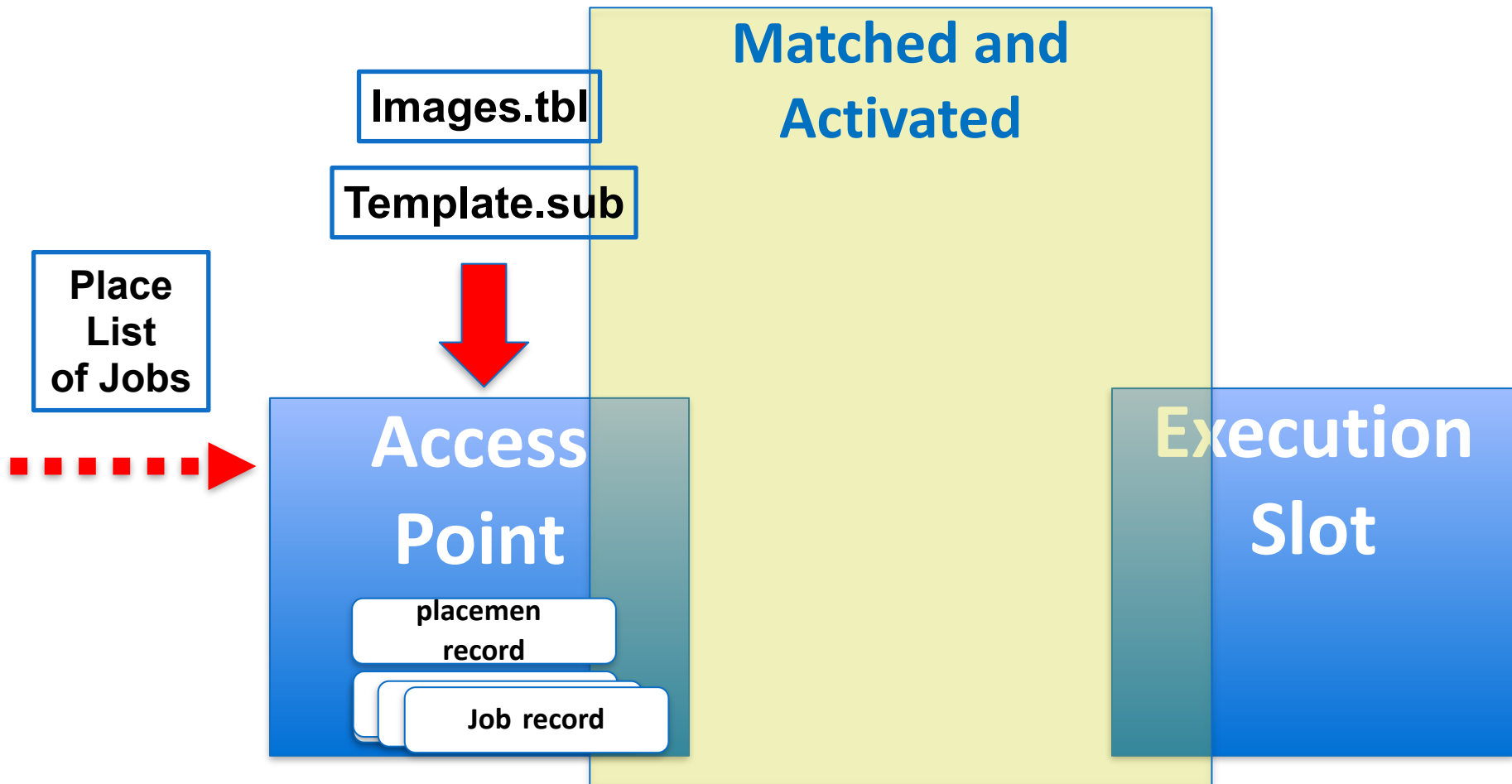
**Step II – Access Point uses Job  
Template and the Job Table  
from the Job List Record to  
Materialize each job into a Job  
Record**

- HTCondor **Access Point** manages several **Persistent Databases**
- One of these Databases stores **Active Placement Records**
- Another Database stores **Active Job Records**
- When records transition from **Active** to **History** they are moved to one of the **Archival Database**\*

\* Due to Disk Space constraints, Access Point may delete “Archival Records”

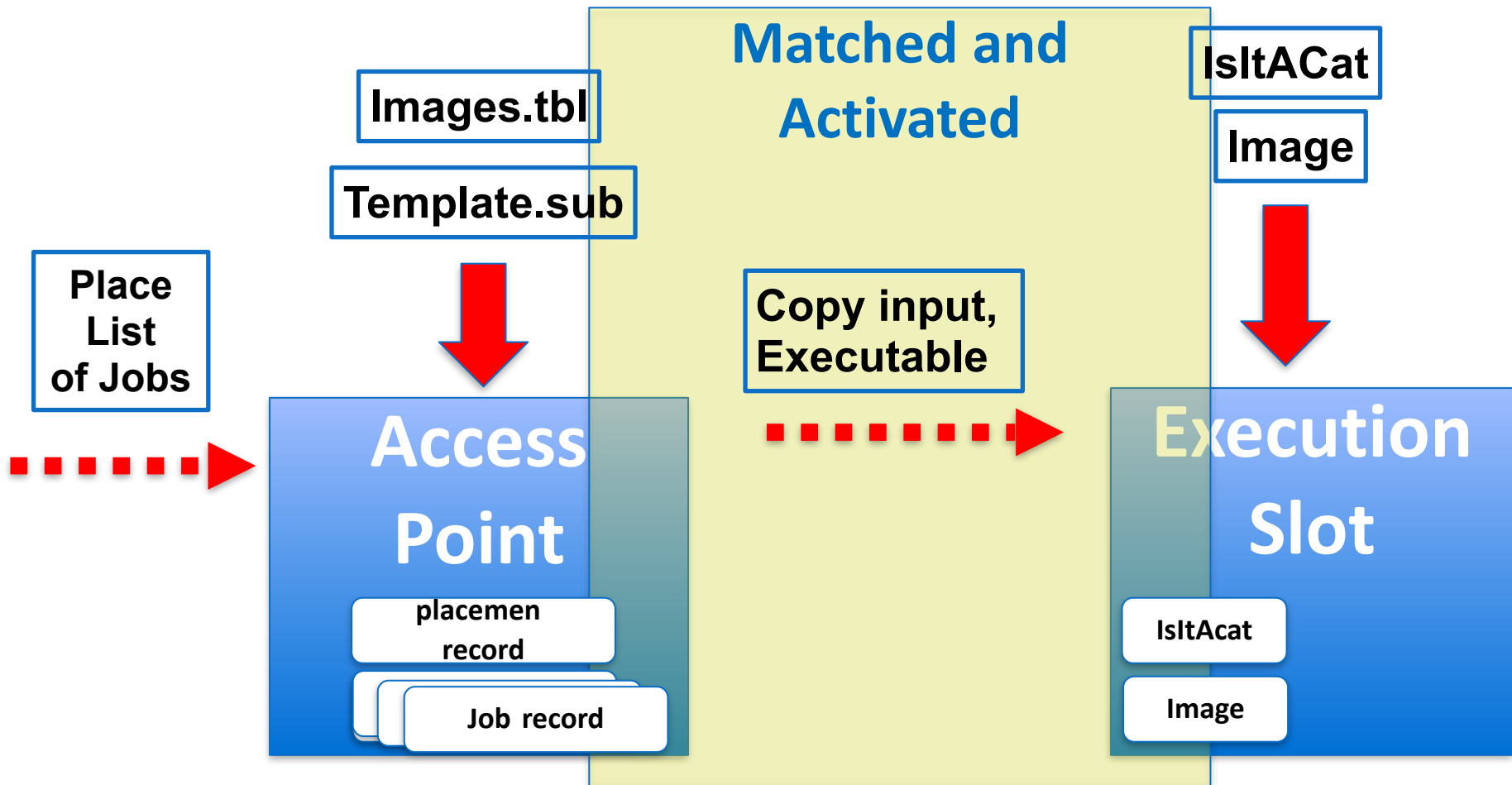


**Step III – Access Point Matches**  
**one of the Job Records with an**  
**HTCondor Execution Slot and**  
**activates the matched**  
**Execution Slot on behalf of the**  
**matched Job**

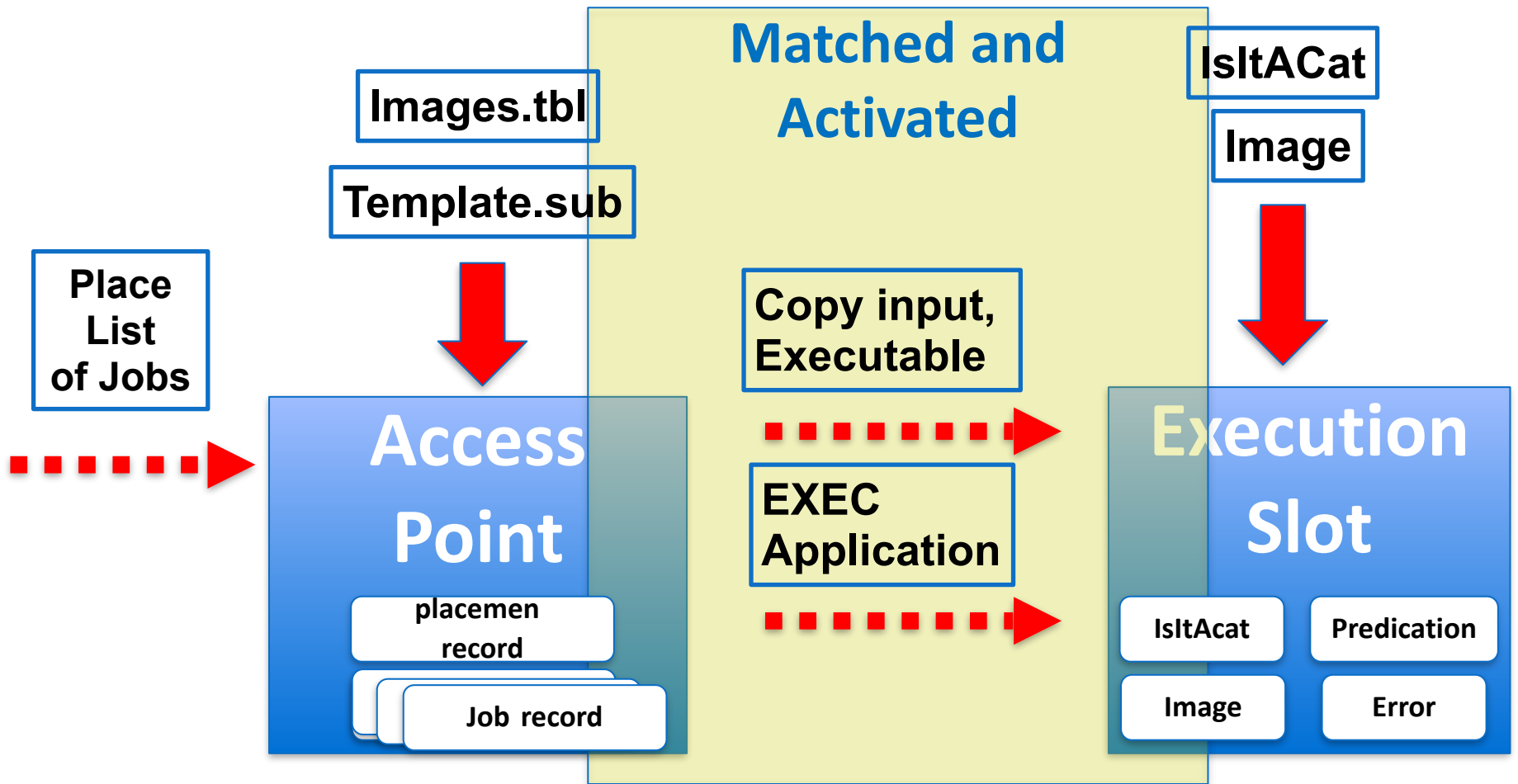


**Step IV – Access Point** copies  
Image file and executable file  
from your working directory/  
folder to  
**Execution Slot**

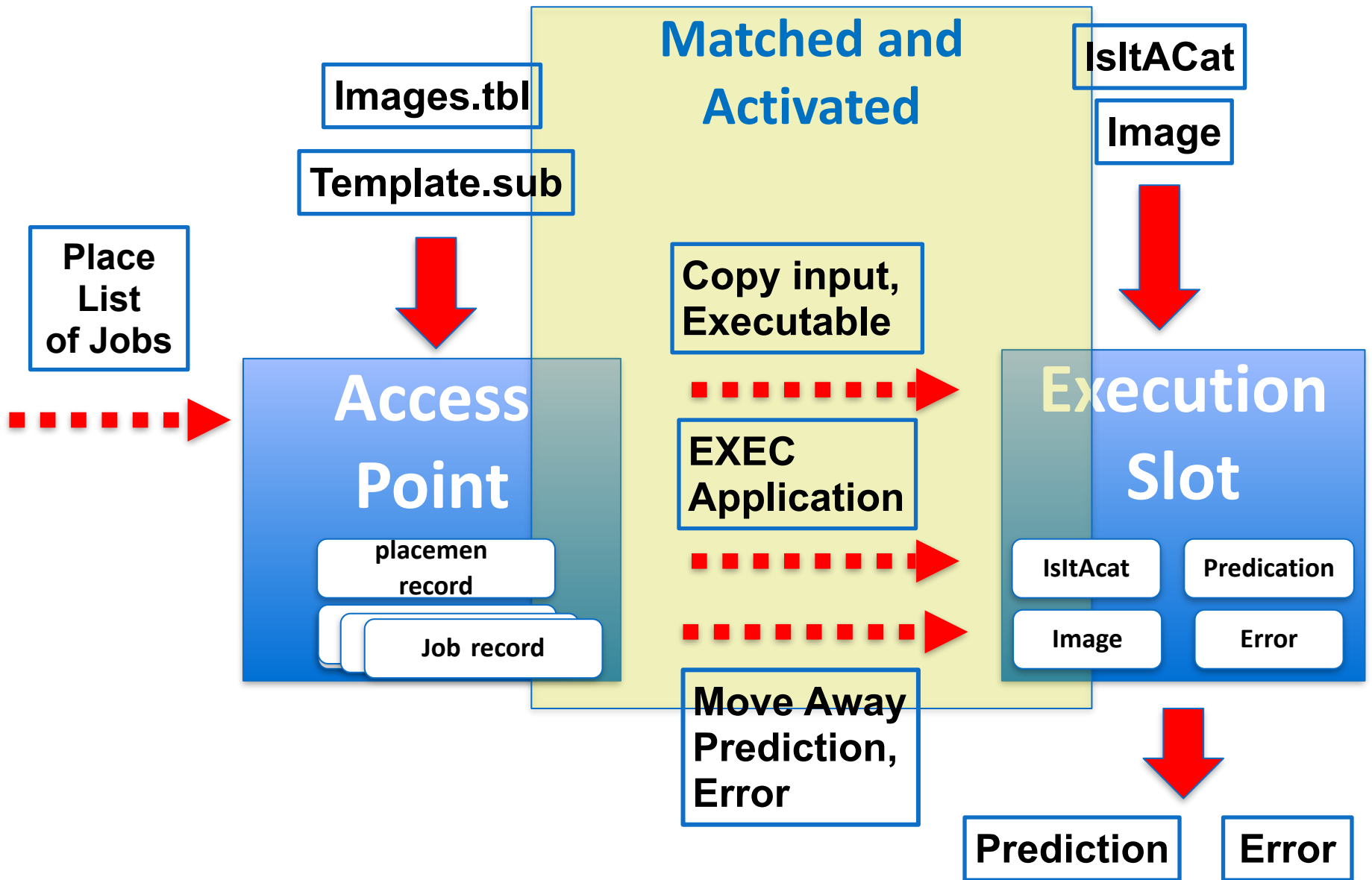




**Step V – Access Point triggers  
the execution of the Application  
at the  
Execution Slot**



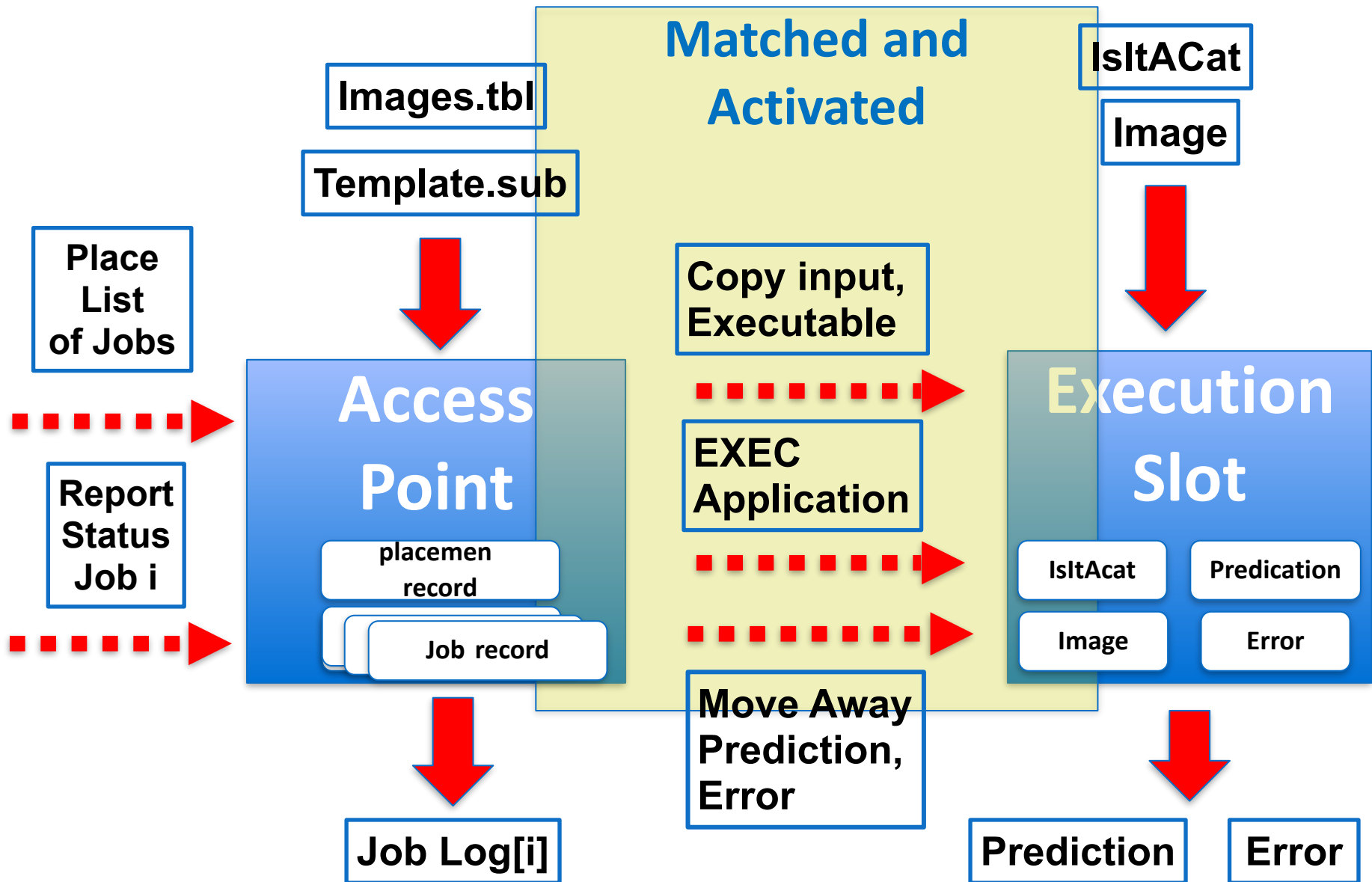
**Step VI – Access Point** moves  
the files created by the  
Application (Prediction file and  
Error File) away from the  
**Execution Slot** to  
your working directory/folder



**Step VII– Access Point moves  
Job Record from Active to  
History. Does the same for  
Placement Record if Job is last  
to complete for  
Job List Placement**

# Step VIII – Retrieve the Job Log File

placed by the **Access Point** in  
your working directory/folder





```

000 (416638.000.000) 2024-08-08 15:35:24 Job submitted from host: <128.105.68.112:9618?addr=128.105.68.112-9618+[2607-f388-2200-100-eaeb-d3ff-
fea3-4202]-9618&alias=ap2001.chtc.wisc.edu&noUDP&sock=schedd_322984_bce1>
...
040 (416638.000.000) 2024-08-08 15:39:15 Started transferring input files
Transferring to host: <128.105.68.125:9618?addr=128.105.68.125-9618+[2607-f388-2200-100-ba3f-d2ff-fe19-86da]-9618&alias=txie-
dsigpu4000.chtc.wisc.edu&noUDP&sock=backfill13_15_1779611_b0c8_17563>
...
040 (416638.000.000) 2024-08-08 15:39:15 Finished transferring input files
...
001 (416638.000.000) 2024-08-08 15:39:16 Job executing on host: <128.105.68.125:9618?addr=128.105.68.125-9618+[2607-f388-2200-100-ba3f-d2ff-
fe19-86da]-9618&alias=txie-dsigpu4000.chtc.wisc.edu&noUDP&sock=startd_2590937_db07>
SlotName: backfill13_15@txie-dsigpu4000.chtc.wisc.edu
AvailableGPUs = { }
CondorScratchDir = "/var/lib/condor/execute/slot3/dir_3281451"
Cpus = 1
Disk = 28095
GPUs = 0
Memory = 128
...
006 (416638.000.000) 2024-08-08 15:39:16 Image size of job updated: 600
1 - MemoryUsage of job (MB)
248 - ResidentSetSize of job (KB)
...
040 (416638.000.000) 2024-08-08 15:39:16 Started transferring output files
...
040 (416638.000.000) 2024-08-08 15:39:16 Finished transferring output files
...
005 (416638.000.000) 2024-08-08 15:39:16 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
67 - Run Bytes Sent By Job
233 - Run Bytes Received By Job
67 - Total Bytes Sent By Job
233 - Total Bytes Received By Job
Partitionable Resources : Usage Request Allocated
Cpus : 0 1 1
Disk (KB) : 28 1024 28095
GPUs : 0
Memory (MB) : 1 1 128
TimeExecute (s) : 0
TimeSlotBusy (s) : 1

Job terminated of its own accord at 2024-08-08T20:39:16Z with exit-code 0.
...

```

```
000 (416638.000.000) 2024-08-08 15:35:24 Job submitted from host:
<128.105.68.112:9618?addrs=128.105.68.112-9618+[2607-f388-2200-100-eaeb-
d3ff-
fea3-4202]-9618&alias=ap2001.chtc.wisc.edu&noUDP&sock=schedd_322984_bce1>
...
040 (416638.000.000) 2024-08-08 15:39:15 Started transferring input files
Transferring to host: <128.105.68.125:9618?
addrs=128.105.68.125-9618+[2607-f388-2200-100-ba3f-d2ff-
fe19-86da]-9618&alias=txie-
dsigpu4000.chtc.wisc.edu&noUDP&sock=backfill13_15_1779611_b0c8_17563>
...
040 (416638.000.000) 2024-08-08 15:39:15 Finished transferring input files
...
001 (416638.000.000) 2024-08-08 15:39:16 Job executing on host:
<128.105.68.125:9618?addrs=128.105.68.125-9618+[2607-f388-2200-100-ba3f-
d2ff-fe19-86da]-9618&alias=txie-
dsigpu4000.chtc.wisc.edu&noUDP&sock=startd_2590937_db07>
SlotName: backfill13_15@txie-dsigpu4000.chtc.wisc.edu
AvailableGPUs = { }
CondorScratchDir = "/var/lib/condor/execute/slot3/dir_3281451"
Cpus = 1
Disk = 28095
GPUs = 0
Memory = 128
```

```
...
006 (416638.000.000) 2024-08-08 15:39:16 Image size of job
updated: 600
    1 - MemoryUsage of job (MB)
    248 - ResidentSetSize of job (KB)
...
040 (416638.000.000) 2024-08-08 15:39:16 Started
transferring output files
...
040 (416638.000.000) 2024-08-08 15:39:16 Finished
transferring output files
...
...
```

005 (416638.000.000) 2024-08-08 15:39:16 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

67 - Run Bytes Sent By Job  
233 - Run Bytes Received By Job  
67 - Total Bytes Sent By Job  
233 - Total Bytes Received By Job

Partitionable Resources	:	Usage	Request	Allocated
Cpus	:	0	1	1
Disk (KB)	:	28	1024	28095
GPUs	:			0
Memory (MB)	:	1	1	128
TimeExecute (s)	:	0		
TimeSlotBusy (s)	:	1		

Job terminated of its own accord at 2024-08-08T20:39:16Z  
with exit-code 0.

...

- You can make your **Job Log File** more readable by adding to your job template

**submit\_event\_user\_notes = \$(JOBID) \$**

**(IMAGE)**

- You can switch to a **Placement Log File** by changing the value of the **Log Attribute** in the **Job Template** to

**log = \$(CLUSTERID).log**

[HTCondor](#)[HTCondor-CE](#)

## HTCondor Documentation

### Feature Channel

This guide provides enough guidance to submit and observe the successful completion of a first job. It then suggests extensions that you can apply to your particular jobs.

[User Quick Start Guide](#)[Admin Quick Start Guide](#)

For more details and a full reference to HTCondor's capabilities and configuration, see the HTCondor Manual. The HTCondor Manual may be viewed online or downloaded to your site.

[HTCondor Manual](#)

### Long Term Support (LTS) Channel

This guide provides enough guidance to submit and observe the successful completion of a first job. It then suggests extensions that you can apply to your particular jobs.

[User Quick Start Guide](#)[Admin Quick Start Guide](#)

For more details and a full reference to HTCondor's capabilities and configuration, see the HTCondor Manual. The HTCondor Manual may be viewed online or downloaded to your site.

[HTCondor Manual](#)

### General

[Job Submission Examples](#)[How To Recipes](#)

# Some thoughts on Job Lists:

- Current **Condor\_submit** command and the HTCondor Job Descriptions Language support most of the concepts/and functionality I presented
- You ,may implement a script to place a **Job List** as individual jobs
- Today we refer to **Job Records** as **job ClassAds**
- **Placements** are captured by (unique) **ClusterId** and are part of the (unique) **JobId**
- We did not cover how to manage the output files in case you **place** the **Job List** more than once

## Submitting a Job

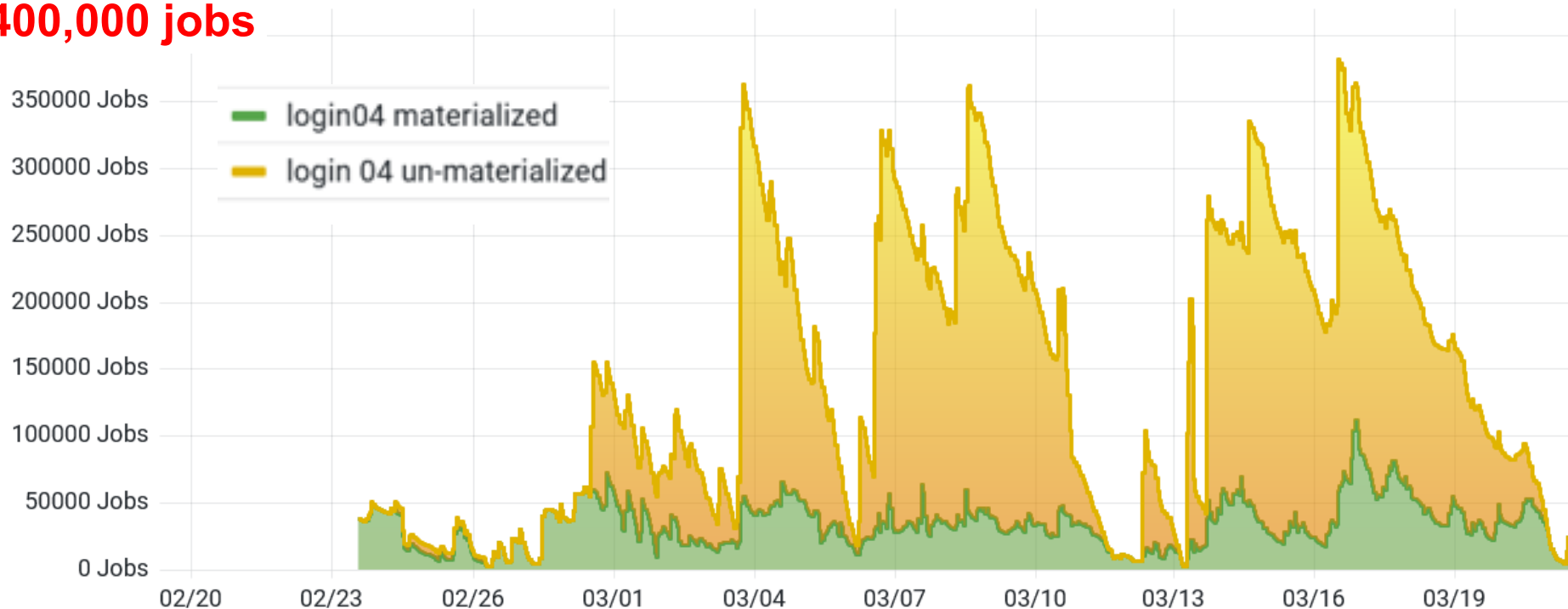
The `condor_submit` command takes a job description file as input and submits the job to HTCondor. In the submit description file, HTCondor finds everything it needs to know about the job. Items such as the name of the executable to run, the initial working directory, and command-line arguments to the program all go into the submit description file. `condor_submit` creates a job ClassAd based upon the information, and HTCondor works toward running the job.



**An Access Point can delay  
the materialization of  
jobs in a Job List into  
active job records**

# One week in the Life of an Access Point # of jobs managed

400,000 jobs





<https://path-cc.io/contact/>



## Contact

PATH is a unique partnership between the Center for High Throughput Computing (CHTC) and the OSG Consortium.

- For enquiries about the *PATH project*, please contact the [PATH leadership](#).
- For help with *CHTC technologies* such as the HTCondor Software Suite (HTCSS), contact [chtc@cs.wisc.edu](mailto:chtc@cs.wisc.edu).
- *Campuses* interested in providing resources to the [Open Science Pool \(OSPool\)](#) can contact [support@osg-htc.org](mailto:support@osg-htc.org)
- *Users* interested in using an Access Point to leverage resource like the OSPool can contact [support@osgconnect.net](mailto:support@osgconnect.net).
- *PIs* interested in getting credit accounts on PATH-managed hardware should visit the [dedicated page](#).

**This work is supported by the National Science Foundation under Cooperative Agreements OAC-2030508, OAC-2331480. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.**