



A Proper Introduction to the OSPool

Tim Cartwright

University of Wisconsin–Madison

OSG School Director

OSG Campus Coordinator



Don't let computing be a barrier to research!

**Thus, you need lots of capacity,
which the OSPool offers,
but often you need to know
what capacity may be available & *how it works*
to benefit the most**





The Ideal



Submit locally, run globally





What You Learned Yesterday

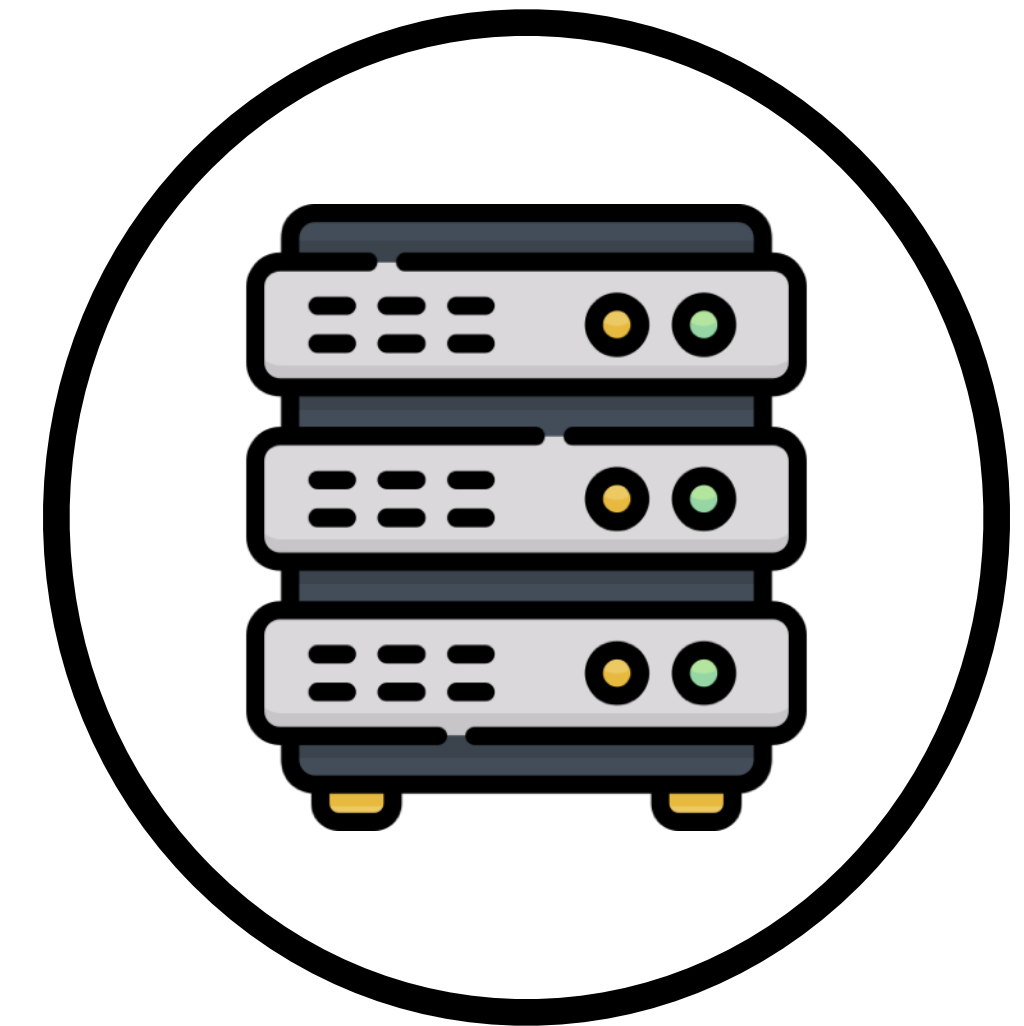


Access Point

```
$ ls  
science.sub  
input.dat  
$ condor_q
```



HTCondor (OS)Pool



<https://sweetclipart.com/>
<https://www.flaticon.com/free-icons/server>



This Lecture



Access Point

```
$ ls  
science.sub  
input.dat  
$ condor_q
```



OSPool





A Local HTCondor Pool



Common characteristics of one local HTCondor pool:

- One owner, one admin (team)
- If all machines from 1 purchase, then they may be identical
- Racked together
- Networked together
- Same OS, software, access, ...

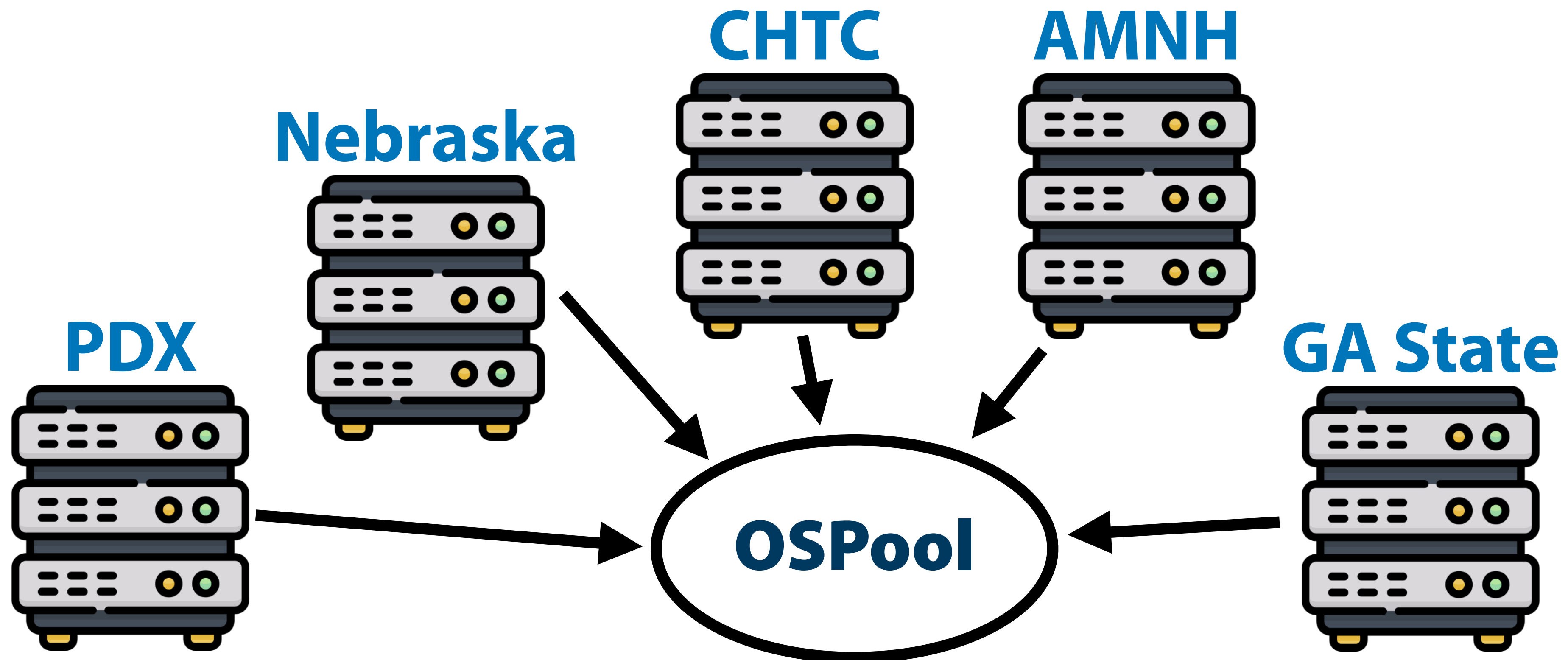




Open Science Pool (OSPool)



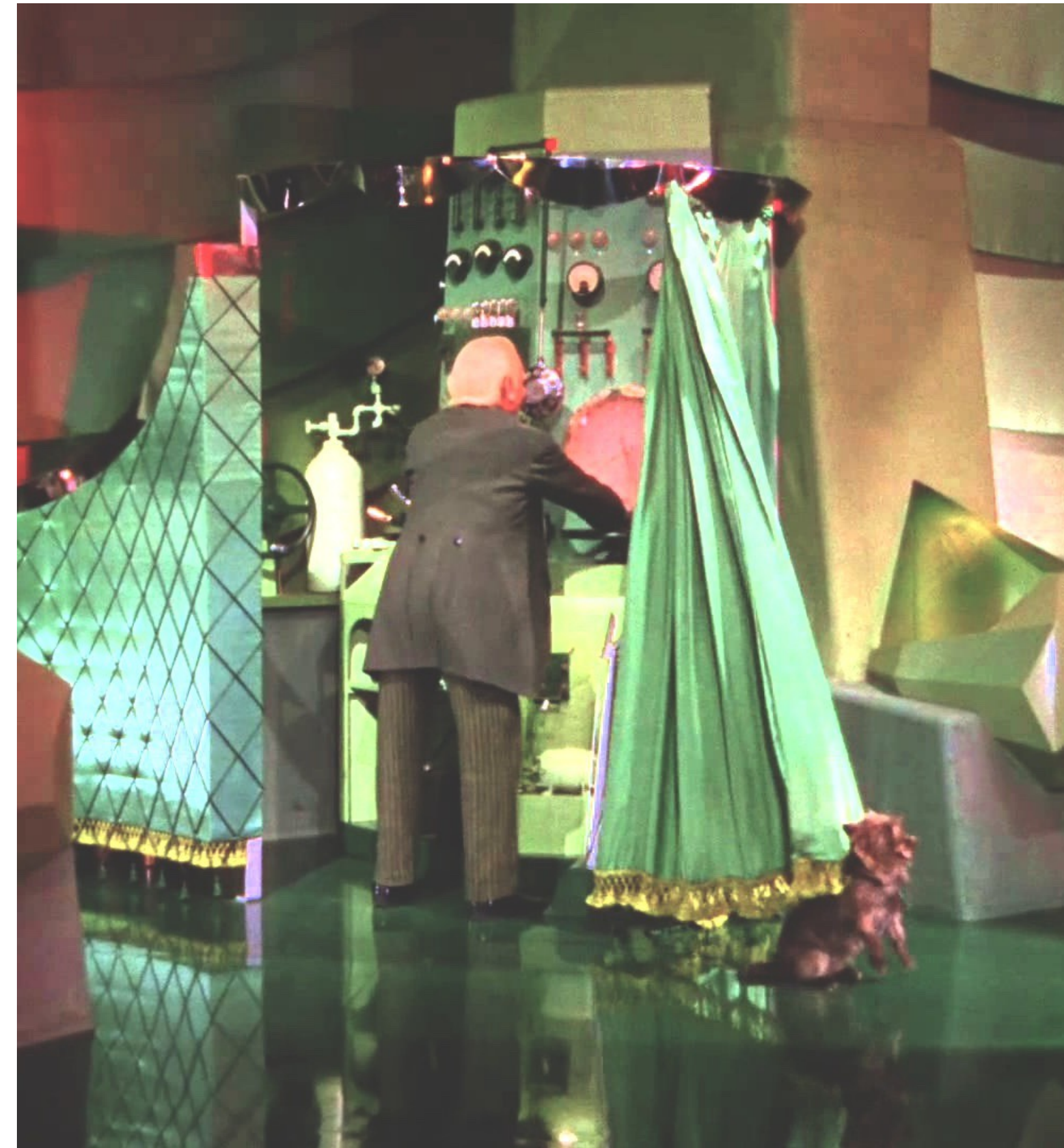
Still a single HTCondor pool, BUT:
All capacity is contributed on a volunteer basis!



<https://www.flaticon.com/free-icons/server>



Behind the Curtain





Reasons for Continuing



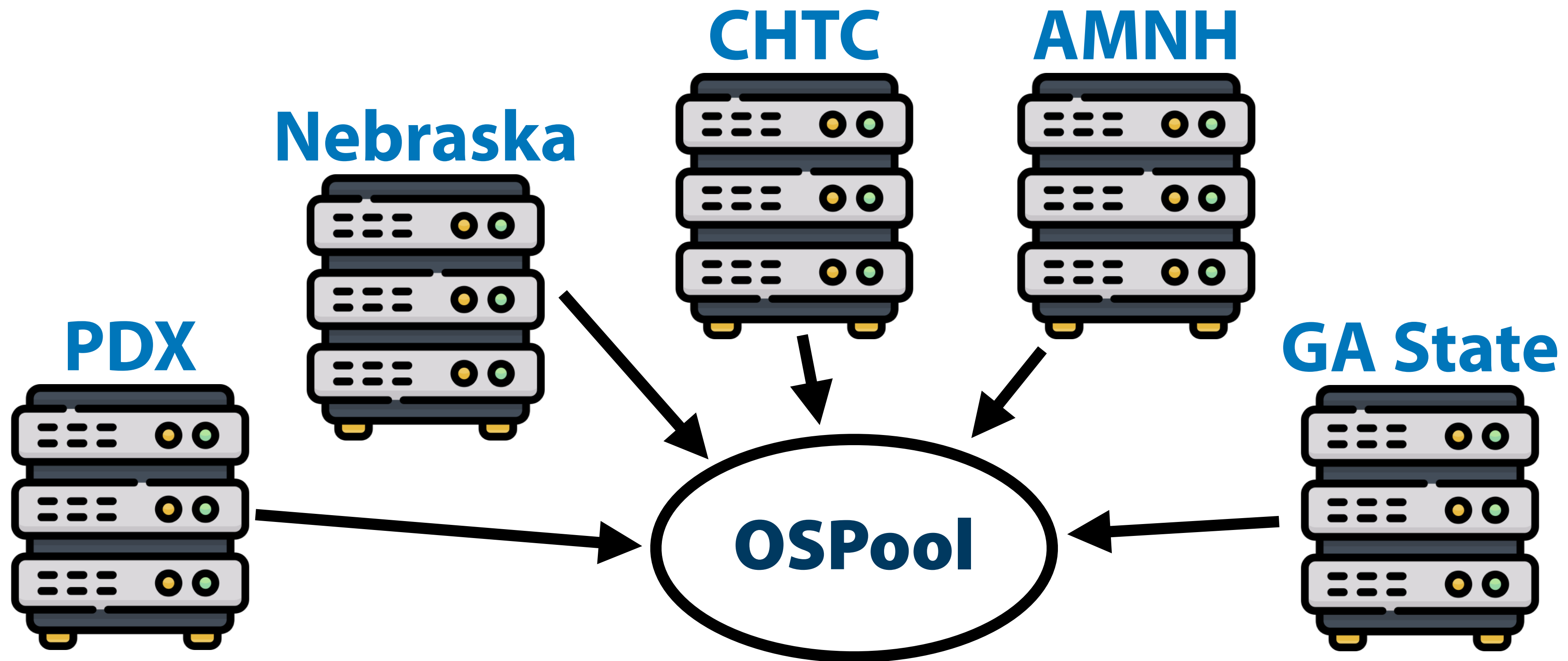
- So why learn more about how the OSPool works?
 - May change how you plan to run computing there
 - May change the way you use the Access Point
 - May change how you handle issues that arise
- What is there to learn? (outline of rest of talk)
 - How the OSPool gets capacity
 - Important characteristics of that capacity
 - Why did I cross out ~~OSG~~?
(don't even get me started on "Open Science Grid")



Getting Capacity for OSPool



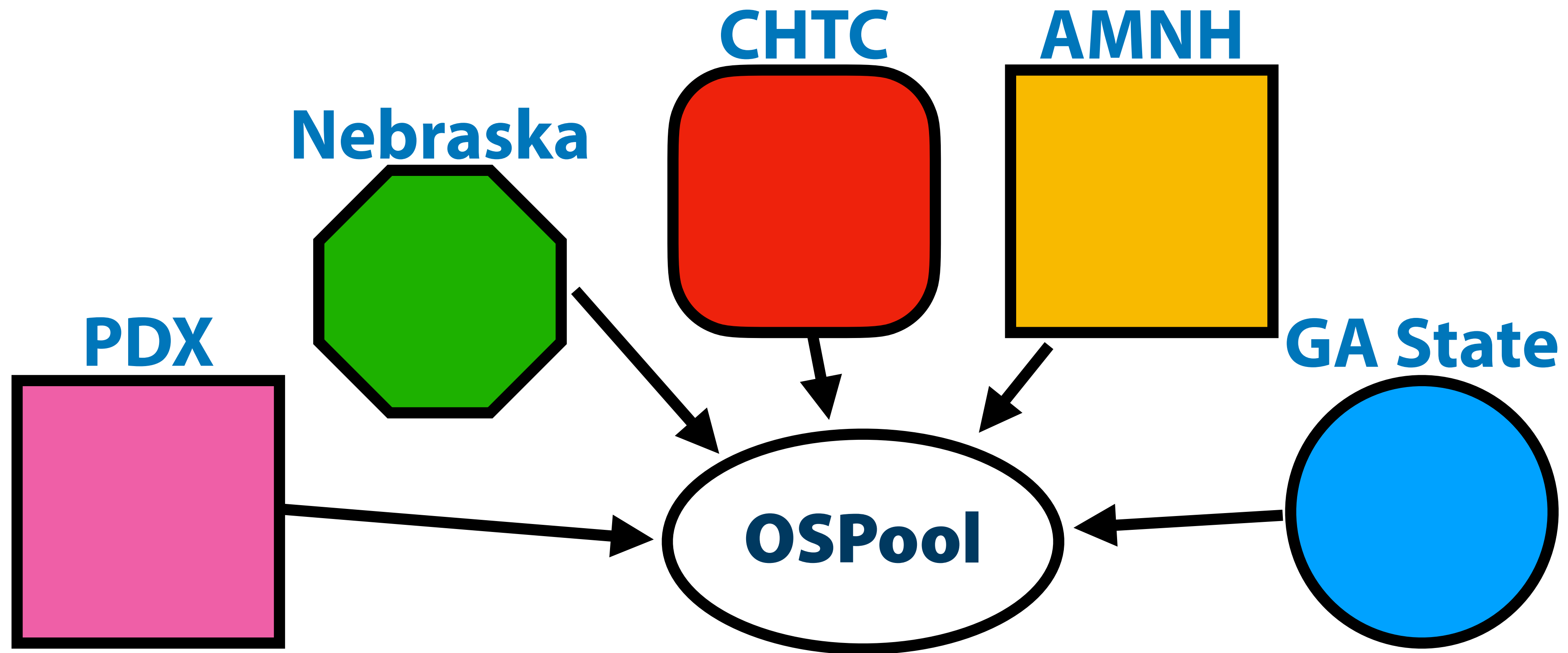
This Diagram Is *Still* Pretty Vague



<https://www.flaticon.com/free-icons/server>



This Diagram Is *Still* Pretty Vague



<https://www.flaticon.com/free-icons/server>



1. Before OSPool

Nothing available at Wisc. 😭

Access Point

Job1.0
Job1.1
Job1.2
Job1.3
...
Job1.1999

CHTC

<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>



2. Add capacity contributors!

Access Point

Job1.0
Job1.1
Job1.2
Job1.3
...
Job1.1999

Nebraska

<i>Busy</i>
<i>Busy</i>

Portland St.

<i>Busy</i>
<i>Busy</i>

CHTC

<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>

GA State

<i>Busy</i>
<i>Busy</i>
<i>Busy</i>

AMNH

<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>
<i>Busy</i>



3. Request capacity (method #1)

Start Execution Points on clusters

Access Point

Job1.0
Job1.1
Job1.2
Job1.3
...
Job1.1999

Nebraska

OSPoo1 EP NU1
OSPoo1 EP NU2
Busy
Busy

Portland St.

OSPoo1 EP PS2
Busy
OSPoo1 EP PS1
OSPoo1 EP PS3
Busy

CHTC

Busy
Busy
Busy
Busy
Busy

GA State

OSPoo1 EP GA2
OSPoo1 EP GA1
Busy
Busy
Busy

AMNH

Busy
OSPoo1 EP AM1
Busy
Busy
Busy
OSPoo1 EP AM4
OSPoo1 EP AM2
OSPoo1 EP AM3
Busy
Busy
Busy



4. EPs add capacity to OSP Pool

(I am not explaining how yet)

AP

Job1.0
Job1.1
Job1.2
Job1.3
...
Job1.1999

OSP Pool

OSPoo1 EP NU1	idle
OSPoo1 EP NU2	idle
OSPoo1 EP PS1	idle
OSPoo1 EP PS2	idle
OSPoo1 EP PS3	idle
OSPoo1 EP GA1	idle
OSPoo1 EP GA2	idle
OSPoo1 EP AM1	idle
OSPoo1 EP AM2	idle
OSPoo1 EP AM3	idle
OSPoo1 EP AM4	idle

Nebraska

OSPoo1 EP NU1
OSPoo1 EP NU2
Busy
Busy

Portland St.

OSPoo1 EP PS2
Busy
OSPoo1 EP PS1
OSPoo1 EP PS3
Busy

CHTC

Busy
Busy
Busy
Busy
Busy

GA State

OSPoo1 EP GA2
OSPoo1 EP GA1
Busy
Busy
Busy

AMNH

Busy
OSPoo1 EP AM1
Busy
Busy
Busy
OSPoo1 EP AM4
OSPoo1 EP AM2
OSPoo1 EP AM3
Busy
Busy
Busy



5. Run jobs

Normal HTCondor OSPool

AP

Job1.0
Job1.1
Job1.2
Job1.3
...
Job1.1999



OSP Pool

OSPoo1	EP	NU1	Job1.4
OSPoo1	EP	NU2	idle
OSPoo1	EP	PS1	Job1.0
OSPoo1	EP	PS2	Job1.3
OSPoo1	EP	PS3	idle
OSPoo1	EP	GA1	Job1.2
OSPoo1	EP	GA2	Job1.6
OSPoo1	EP	AM1	Job1.8
OSPoo1	EP	AM2	Job1.12
OSPoo1	EP	AM3	Job1.10
OSPoo1	EP	AM4	idle

Nebraska

NU1 > Job1.4
NU2 > idle
Busy
Busy

Portland St.

PS2 > Job1.3
Busy
PS1 > Job1.0
PS3 > idle
Busy

CHTC

Busy
Busy
Busy
Busy
Busy

GA State

GA2 > Job1.6
GA1 > Job1.2
Busy
Busy
Busy

AMNH

Busy
AM1 > Job1.8
Busy
Busy
Busy
AM4 > idle
AM2 > Job1.12
AM3 > Job1.10
Busy
Busy
Busy



6. Get resources (method #2)

Direct contributions

AP

Job1.0
Job1.1
Job1.2
Job1.3
...
Job1.1999



OSP Pool

OSPoo1	EP	NU1	Job1.4
OSPoo1	EP	NU2	idle
OSPoo1	EP	PS1	Job1.0
OSPoo1	EP	PS2	Job1.3
OSPoo1	EP	PS3	idle
OSPoo1	EP	GA1	Job1.2
OSPoo1	EP	GA2	Job1.6
OSPoo1	EP	AM1	Job1.8
OSPoo1	EP	AM2	Job1.12
OSPoo1	EP	AM3	Job1.10
OSPoo1	EP	AM4	idle
AM Contrb	EP1		idle

Nebraska

NU1 > Job1.4
NU2 > idle
Busy
Busy

Portland St.

PS2 > Job1.3
Busy
PS1 > Job1.0
PS3 > idle
Busy

CHTC

Busy
Busy
Busy
Busy
Busy

GA State

GA2 > Job1.6
GA1 > Job1.2
Busy
Busy
Busy

AMNH

Busy
AM1 > Job1.8
Busy
Busy
Busy
AM4 > idle
AM2 > Job1.12
AM3 > Job1.10
Busy
AM Contrb EP1
Busy
Busy



OSPool dHTC – A Few Details

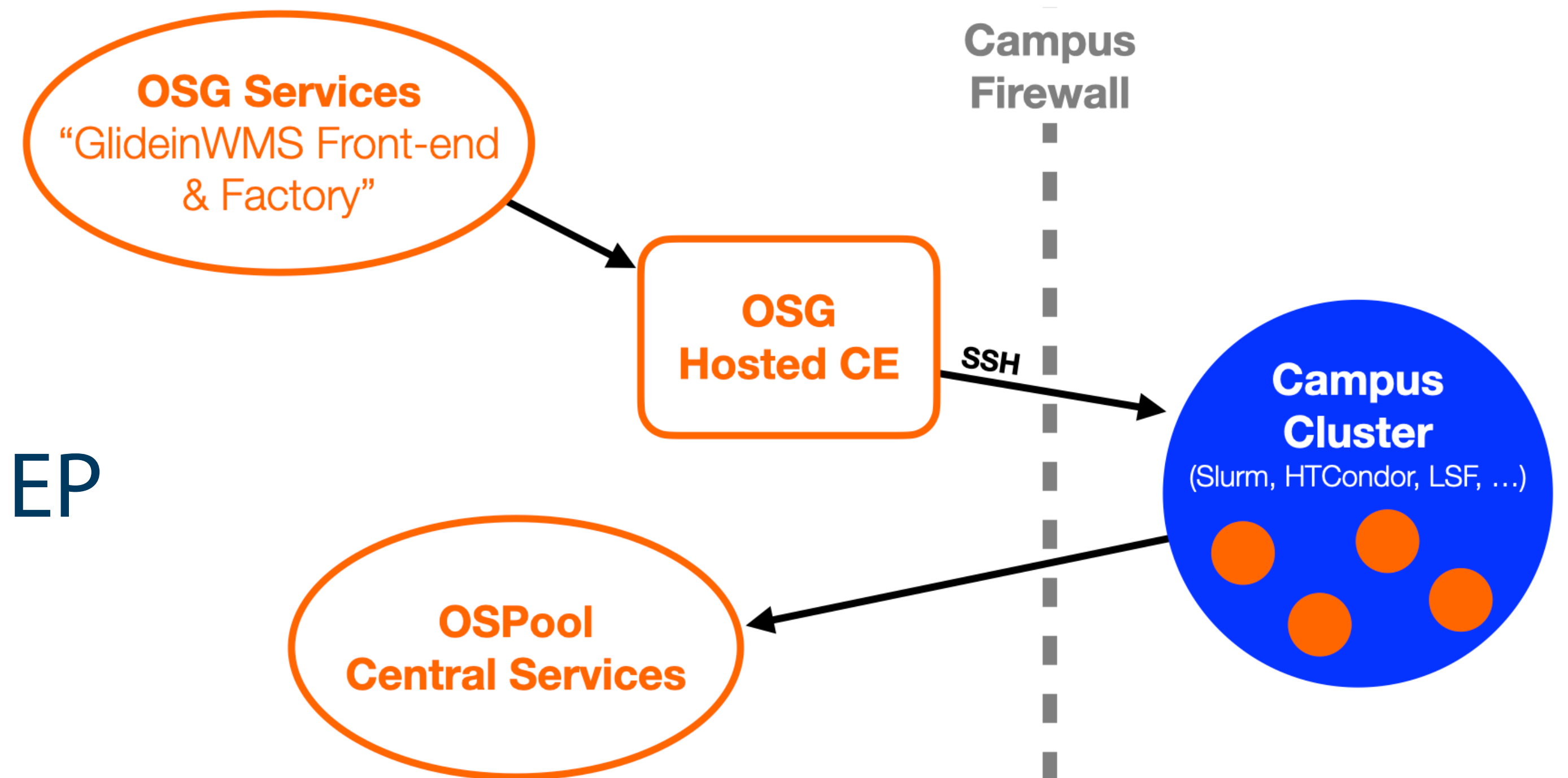


For a batch scheduler:

- CE requests capacity (as jobs) based on demand
- Scheduler may run req.s
- Our SW creates *Execution Point* & adds to OSPool
- OSPool workloads run on EP

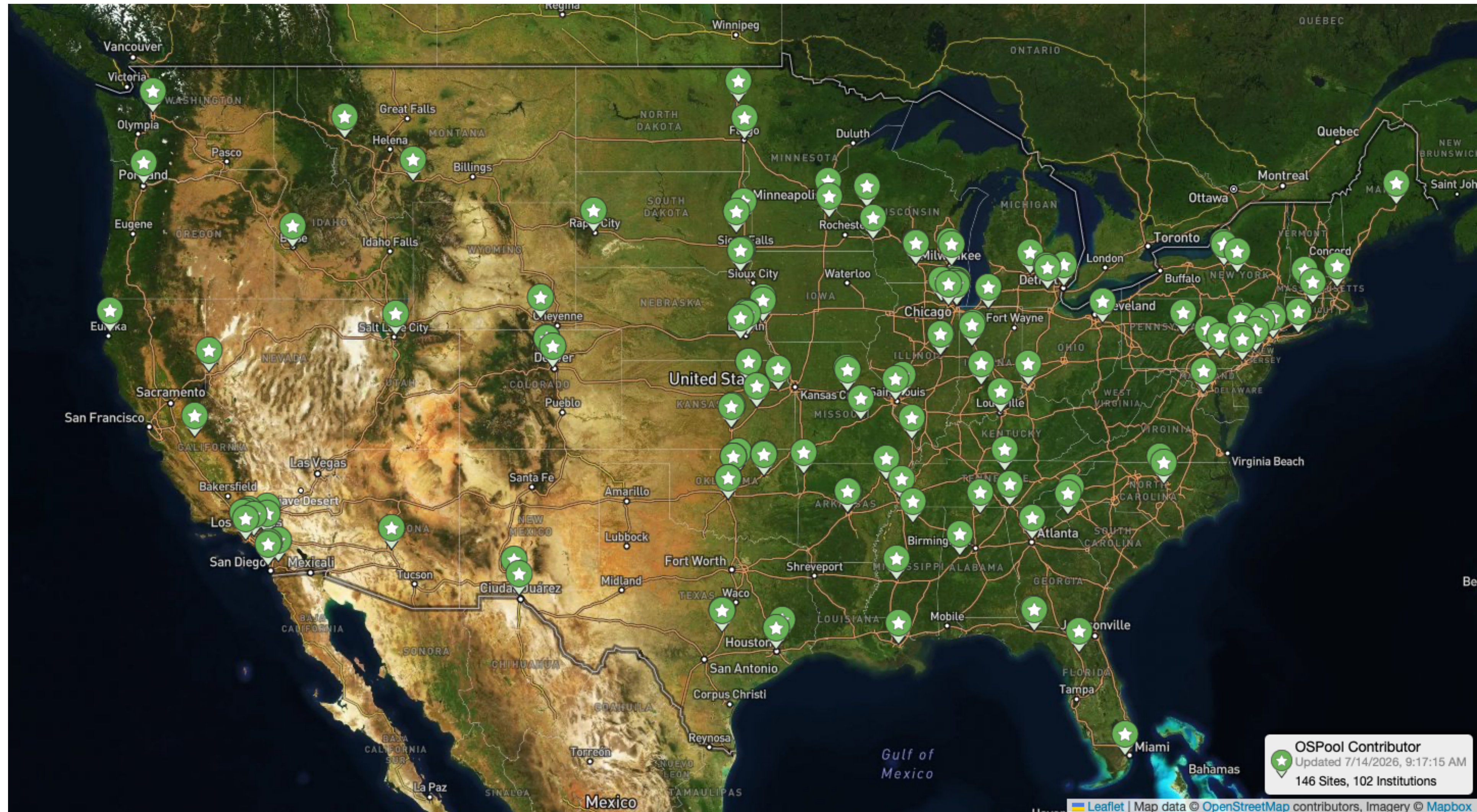
Using containers:

- Admin starts containers
- SW makes EP, etc. (same)





OSPool Contributors (*United States*)





Brief Digression...



- This is my day job! *OSG Campus Coordinator*
- I work with the campuses that contribute to the OSPool (and OSDF – see tomorrow)
- We have a couple campus admins here this week!



13 July 2026 (Mon)

593K jobs completed (6.8 Hz)

Placed by 157 researchers (incl. you!)

Using 3.4M file transfers

Consuming 840K core hours

Across 75 sites at 81 institutions



Using the OSPool



The OSPool is the potluck of HTCondor pools



It is highly varied and always changing (“dynamic”)



And So?



Given the OSPool's varied and dynamic nature:

Danny's Decree:

Put some extra thought into defining your jobs

E.g., requested CPUs, GPUs, memory, disk; how to transfer input and output; how to handle software; special hardware, environmental, or other needs

This is a big part of why you are here!



Dimensions of Variation



- CPUs (type, speed)
- GPUs (type, GPU memory, availability $\Rightarrow$ *Thu PM*)
- Memory
- Data (storage capacity + transfer in/out $\Rightarrow$ *Wed*)
- Networking (consider a small, rural college)
- Operating system (all Linux, but...)
- Software environment ($\Rightarrow$ *today AM*)
- Policies: Owner/Local + OSPool per-site



Three Key Guidelines



- Bring what you can (esp. software)
- Declare what you need
- Try to make “OSPool-sized” jobs (next)



What Makes a Good OSPool Job?



	Ideal Jobs! (1,000s concurrent jobs)	Still Advantageous! (100s concurrent jobs)	Maybe not, but talk to us!
Cores (GPUs)	1 (0)	< 8 (1)	> 8 (or MPI) (> 1)
Walltime	< 10 hrs* *or checkpointable	< 20 hrs* *or checkpointable	> 20 hrs (not a good fit)
RAM	< few GB	< 40 GB	> 40 GB
Input	< 500 MB	< 10 GB	> 10 GB (per job)
Output	< 1 GB	< 10 GB	> 10 GB (per job)
Software	<i>pre-compiled binaries, containers</i>	<i>most other than →</i>	<i>Licensed software; non-Linux</i>

See also: https://portal.osg-htc.org/documentation/overview/account_setup/is-it-for-you/#computational-fit-on-the-ospool



The OSPool (and APs, etc.) are a ***shared*** resource

- **Basic security tips**
 - Don't share account + use keys or great passwords
 - Avoid sensitive software and data (no HIPAA!)
 - Do not try to work around security barriers;
contact us to reach your goals safely for all
- **Acceptable Use Policy (AUP)**
tl;dr Use OSPool only for academic, non-profit research



What is “OSG”?





OSG Defined, Version 1



OSG Consortium – in this view, OSG *is* people:

- Users: individual PIs/students to collaborations of 1000s
- Capacity owners/contributors
- Team: provides infrastructure, support, features, ...





Pools of capacity

- **Capacity:** compute, storage, and other systems that can be used for research workflows
- **Services:** software infrastructure that manages capacity and makes features available



<https://www.pngall.com/wp-content/uploads/5/Server-Rack-PNG-Free-Image.png>



Your Access Point (OSPool or otherwise)

- Where you go to do computing
- Has access to capacity
- Provides means for accessing data

```
[tim.cartwright@ap40 ~]$ condor_version  
$CondorVersion: 10.7.0 2023-07-10 BuildID: 659788 PackageID: 10.7.0-0.659788 RC $  
$CondorPlatform: x86_64_AlmaLinux8 $  
[tim.cartwright@ap40 ~]$ █
```



Acknowledgements



You Can Acknowledge OSG!



If you publish or present results that benefitted from using OSG services, please acknowledge us!

<https://osg-htc.org/acknowledging>



Acknowledgements



OSG team, especially Brian Lin, Christina Koch,
Danny Morales, Mats Rynge, Jason Patton

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.



A Few Suggestions



- **Exercises**

- Today, some exercises will specify less, so try to use what you learned yesterday — first from memory, if possible, then look things up
- Use Slack! There are staff online who can help, too

- **Consultations**

- If you want a consultation, put your name up on the whiteboard, and we will figure out a time



Live Demo