

# Troubleshooting Jobs on OSPool

Showmic Islam

Research Computing Facilitator@ OSG

HPC Applications Specialist

Holland Computing Center

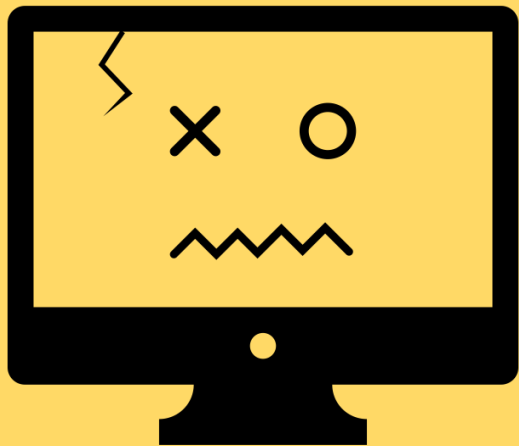
University of Nebraska-Lincoln



# Outline

## Job Failures

- Why a job may fail?
- What can go wrong?
- Reviewing failed jobs
  - Job holds



## Diagnosis

- Tips for troubleshooting
- Diagnosing Holds



## Common Issues

- Examples of typos
  - Badput



# HTCondor Workflow

1. Log in to an OSG  
Access Point\*  
and upload  
data/software



SSH

OSPool Access Point



/home/user

Job Components

- Software
- Scripts
- Input Data

HTCondor Submit File

Job specifications

**HTCondor**

Output  
transferred  
back  
Executable  
Input Files

```
$ condor_submit SubmitFile.submit  
Error/Log Files  
Number of Jobs
```



OSPool Execute Point

/condor/scratch

Software  
Scripts  
Input Data

**Output Data  
Log/Error/Out**



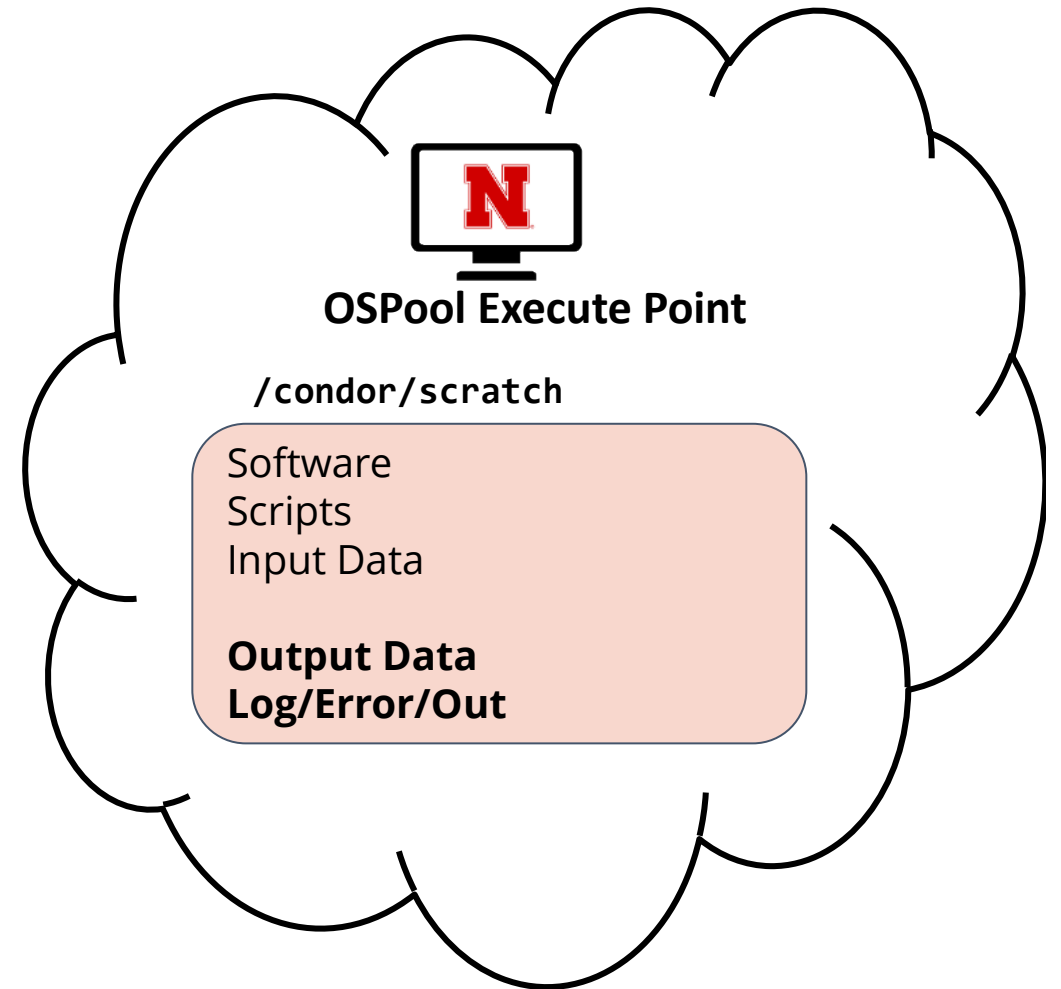
# Job Failures



# What can go wrong?

Let's start with the issues that we are all familiar with

- ❑ The code tries to run and fails
  - Script has typos (e.g. : misspelled input arguments)
  - **Path** names to a file or data are misspelled/wrong
  - Software does not have the **required libraries** (e.g. : mismatch between libraries in Window/linux)



# What can go wrong?

## ❑ OSPool is heterogeneous

- Different varieties of Unix Operating Systems

## ❑ Never assume any required library packages to be present at the execute point

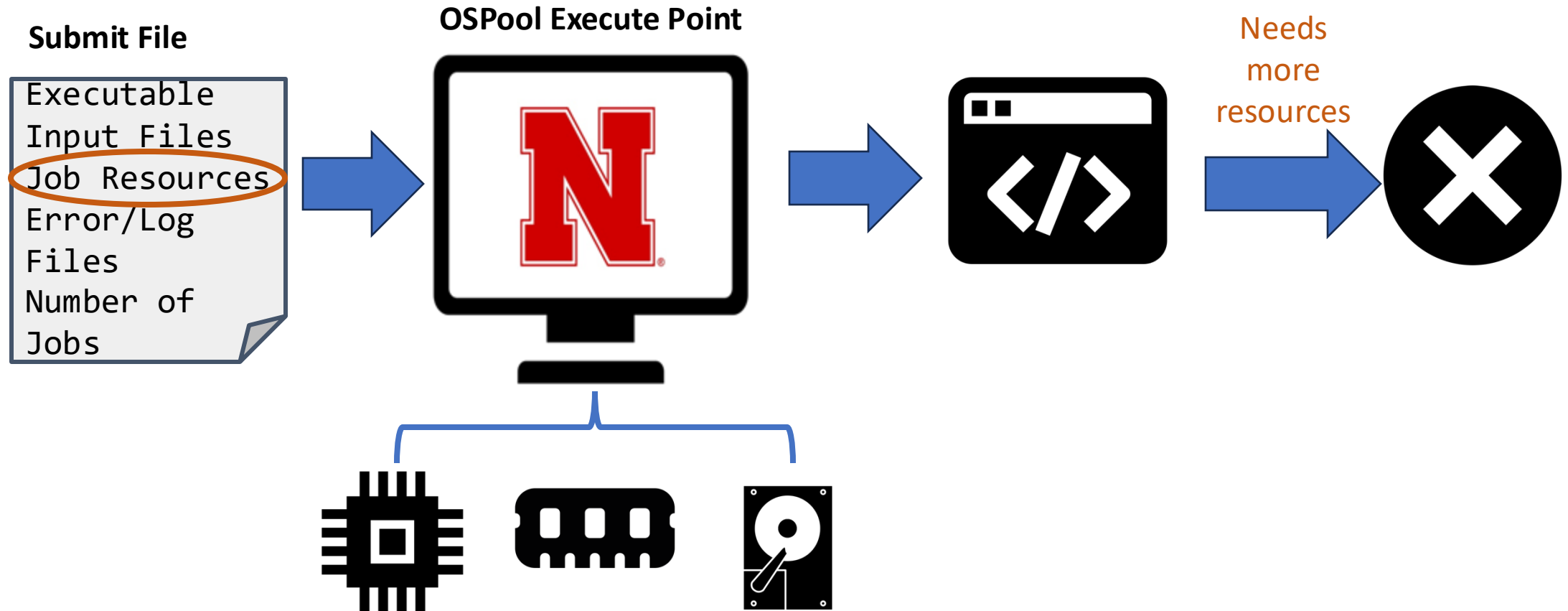
## Sample Software Error Messages

*/srv//cp: /lib64/libc.so.6: version `GLIBC\_2.33' not found (required by /srv//cp) /srv//cp: /lib64/libc.so.6: version `GLIBC\_2.34' not found (required by /srv//cp) /srv//cp: /lib64/libselinux.so.1: no version information available (required by /srv//cp)*

*Traceback (most recent call last): File "/srv//mcpi.py", line 6, in <module> def mcpi(iterations: int) -> tuple[float, int]: TypeError: 'type' object is not subscriptable*



# What else can go wrong?



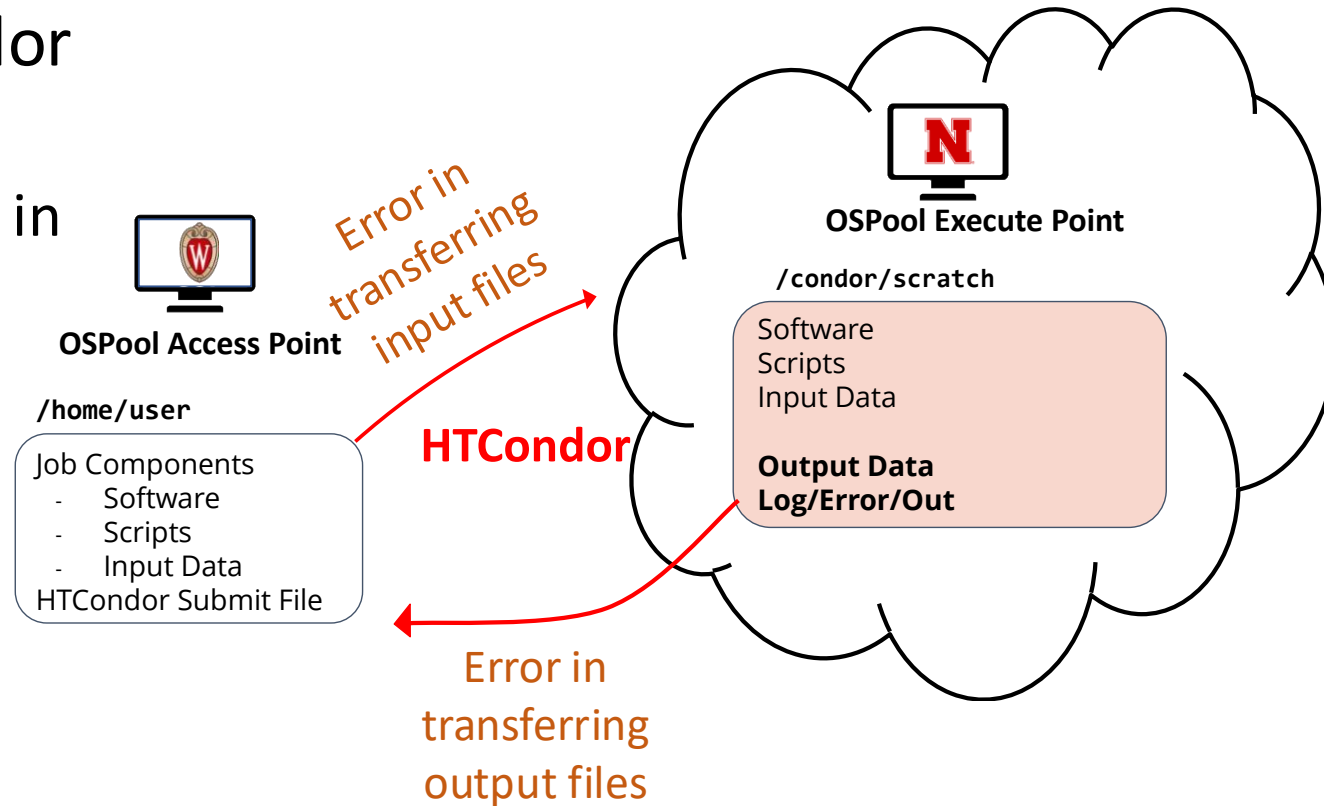
Computer by miracle from NounProject.com  
CPU by Prithvi from NounProject.com  
RAM by George Ianta from NounProject.com  
Code by Lagot Design from NounProject.com  
Hard drive by Perilisima Shoeder from NounProject.com  
Fail by Bluetip Design from NounProject.com



# What other things can go wrong?

## ❑ Jobs can go wrong in the HTCondor workflow.

- A job can't be matched (no machine in the pool to accommodate user's request)
- Files not found for transfer
- Job used too much memory
- Job used too much disk space
- Badly-formatted executable
- And many more





# Job Holds

- ❑ HTCondor will *hold* your job if there's a *logistical* issue that YOU (or maybe an admin) need to fix.
  - files not found for transfer, over memory, etc.
- ❑ A job that goes on hold is interrupted (all progress is lost) but remains in the queue in the “H” state until removed, or (fixed and) released.



```
$ condor_q
OWNER  BATCH_NAME  SUBMITTED  DONE  RUN  IDLE  HOLD  TOTALJOB_IDS
cat    ID:123456  7/11 11:23  _    _    _    1    1 123456.0
```

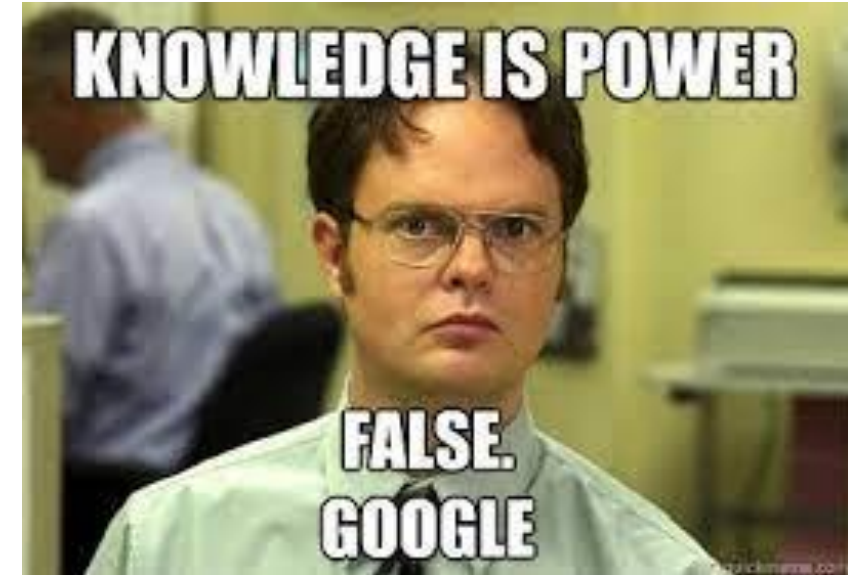


# Diagnosis



# General Troubleshooting Tips

- Comparing **expectations vs. what happened**: Either might be wrong!
- **Read messages carefully** — even if some parts make no sense, what hints can you get?
- Search **online** ... but evaluate what you find
- Collect links and other resources that help
- Ask for help! And provide key details: versions, commands, files, messages, logs, etc.
- Always keep the log, error and condor output file



# Reviewing Failed Jobs

- Job log, output and error files can provide valuable troubleshooting details:

Log	Output	Error
<ul style="list-style-type: none"><li>• when jobs were submitted, started, held, or stopped</li><li>• where job ran</li><li>• resources used</li><li>• interruption reasons</li><li>• <b>exit status</b></li></ul>	<ul style="list-style-type: none"><li>• stdout (or other output files) may contain errors from the executable</li></ul>	<ul style="list-style-type: none"><li>• stderr captures errors from the operating system, or reported by the executable, itself.</li></ul>



# Diagnosing Holds: Hold Reasons

If HTCondor puts a job on hold, it provides a hold reason, which can be viewed in the log file, with

**condor\_q -hold <Job.ID>**, or with **<username>**:

*Failed to initialize user log to **/path***

- ☐ Could not create log file, check **/path** carefully

*Error from ...: memory usage exceeded request\_memory*

*Job in status 2 put on hold by SYSTEM\_PERIODIC\_HOLD due to memory usage **BBB**.*

- ☐ Job used too much memory
- ☐ Request more memory than **BBB** megabytes

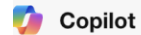
*Transfer input files failure at **access point ap40** while sending files to the execution point. Details: reading from file **/path**: (errno 2) No such file or directory*

- ☐ Job can not find the files in **/path** to transfer to execute point



# Diagnosing Holds: Hold Reasons

HTCondor error reason for the following message: "Transfer input files failure at access point ap40 while sending files to the execution point. Details: reading from file /path: (errno 2) No such file or directory"



This error message indicates that HTCondor is unable to find the specified file at the given path. The error code (errno 2) specifically means "No such file or directory." Here are a few potential reasons and solutions:

1. **Incorrect File Path:** Double-check the file path specified in your job submission script. Ensure that the file exists at the specified location on the submit machine.
2. **File Transfer Settings:** Verify that the file transfer settings in your HTCondor submit file are correctly configured. Ensure that `should_transfer_files` is set to `YES` if you need HTCondor to transfer files from the submit machine to the execute machine [1](#).
3. **File Permissions:** Ensure that the file has the correct permissions and is accessible by the user running the HTCondor job.
4. **Special Characters in File Paths:** Sometimes, special characters or incorrect separators in the file paths can cause issues. Ensure that paths are correctly formatted and use commas to separate multiple file paths [2](#).

If you continue to experience issues, providing more details about your HTCondor submit file and the environment might help in diagnosing the problem further.

Is there anything specific you'd like to dive deeper into?

Learn more [1](#) [htcondor.readthedocs.io](#) [2](#) [github.com](#) [3](#) [stackoverflow.com](#)

🔍 📄 📁 📌 📎 📧 📧 📧



False  
ChatGPT



# Example of common hold reasons

- ❑ Job specified **transfer\_output\_files** but */path* on execute point was not found

*The job exceeded allowed execute duration of 20:00:00*

- ❑ Job ran for too long

*Error from ....: execute point failed to upload checkpoint*

- ❑ Job failed to checkpoint (*more on Friday*)

*Transfer output files failure at access point... while receiving files from the execution point. Details: Error from ....execute point ... failed to send file(s) at apxx; failed to create directory */path* Disk quota exceeded*

- ❑ File **transfer error** due to exceeding disk space



# What To Do About Held Jobs

1. If the situation can be fixed while job is held (e.g., you forgot to create directory for output):

- a. Fix the situation: `condor_qedit`
- b. Release the job(s): `condor_release JOB_IDs`  
`condor_release <username>`

2. Otherwise (and this is common):

- a. Remove the held jobs: `condor_rm JOB_IDs`
- b. Fix the problems
- c. Re-submit





# DEMO



# Common Issues



# Issue: Failed to Parse

```
$ condor_submit job.sh
```

```
Submitting job(s)
```

```
ERROR: on Line 6 of submit file:
```

```
ERROR: Failed to parse command file (line 6).
```

- Completely failed to submit!
- **Notice:** Failed to parse
- **Why:** You tried to submit your executable (or other file), not an HTCondor submit file
- **Fix:** Submit an HTCondor submit file (e.g., **.sub**)



# Issue: Typos in Submit File

```
$ condor_submit sleep.sub
```

Submitting job(s)

- ERROR: No 'executable' parameter was provided
- ERROR: Parse error in expression:  
RequestMemory = 1BG
- ERROR: Executable file /bin/slep does not exist

- Also failed to submit (missing **job(s) submitted**)
- **Why:** Typos in your submit file (e.g., **BG** for **GB**)
- **Fix:** Correct typos!



# Issue: Jobs Idle for a Long Time

```
$ condor_q
```

```
OWNER  BATCH_NAME  SUBMITTED  DONE  RUN  IDLE  TOTAL JOB_IDS
cat    ID: 123456  6/30 12:34  _    _    1    1 123456.0
```

Jobs are **idle** for a **long** time – *can be hard to judge!*

***condor\_q -analyze <JobId>***

***condor\_q -better-analyze <JobId>***

```
$ condor_q -better-analyze 123456.0
```

```
...
```

```
Slots
```

```
Step  Matched Condition
```

```
-----
```

```
[0]    13033 TARGET.PoolName == "OSPool"
```

```
[9]    13656 TARGET.Disk >= RequestDisk
```

```
[11]    0 TARGET.Memory >= RequestMemory
```



# Issue: Missing or Unexpected Results

- ❑ Job runs ... but something does not seem right
  - Short or zero-length output file(s)
  - Very short runtime (almost instant)
- ❑ May be problems with app, inputs, arguments, ...
  - Check log files for **unexpected exit codes**, etc.
  - Check output and error files for messages from app
  - Can't find anything? Add more debugging output



# Issue: Badput

- What is *badput*?
  - Basically, wasted computing
    - Job runs for *97 minutes*, gets kicked off, starts over on another server
    - Job runs for *97 minutes*, is removed
  - Not jobs that **must be re-run** due to code changes! (that's just part of science, right?)
- Badput uses resources that others could have used
- If contacted, help us help you and others!



# Tips for Avoiding Badput

- ❑ Always test with a **small set of jobs** before scaling up. (This practice applies to any modifications made to a **tried and tested** code as well )
- ❑ Monitor your jobs memory and disk usage
  - condor\_q <jobid> -af RequestMemory MemoryUsage |sort |uniq -c*
  - condor\_q <jobid> -af RequestDisk DiskUsage |sort |uniq -c*
- ❑ Have an idea/expectation about the software/code's limit- e.g. **Segfault.**
- ❑ Have a general idea about the inner workings of the software and libraries.





# DEMO 2



# More Troubleshooting Resources

- Tim Cartwright's OSG User School 2021 talk
- OSPool Documentation-<https://portal.osg-htc.org/documentation/>
- Can't solve issues-email us: support@osg-htc.org



# Acknowledgements

Christina Koch; Tim Cartwright

This work was supported by NSF grants MPS-1148698, OAC-1836650, and OAC-2030508

