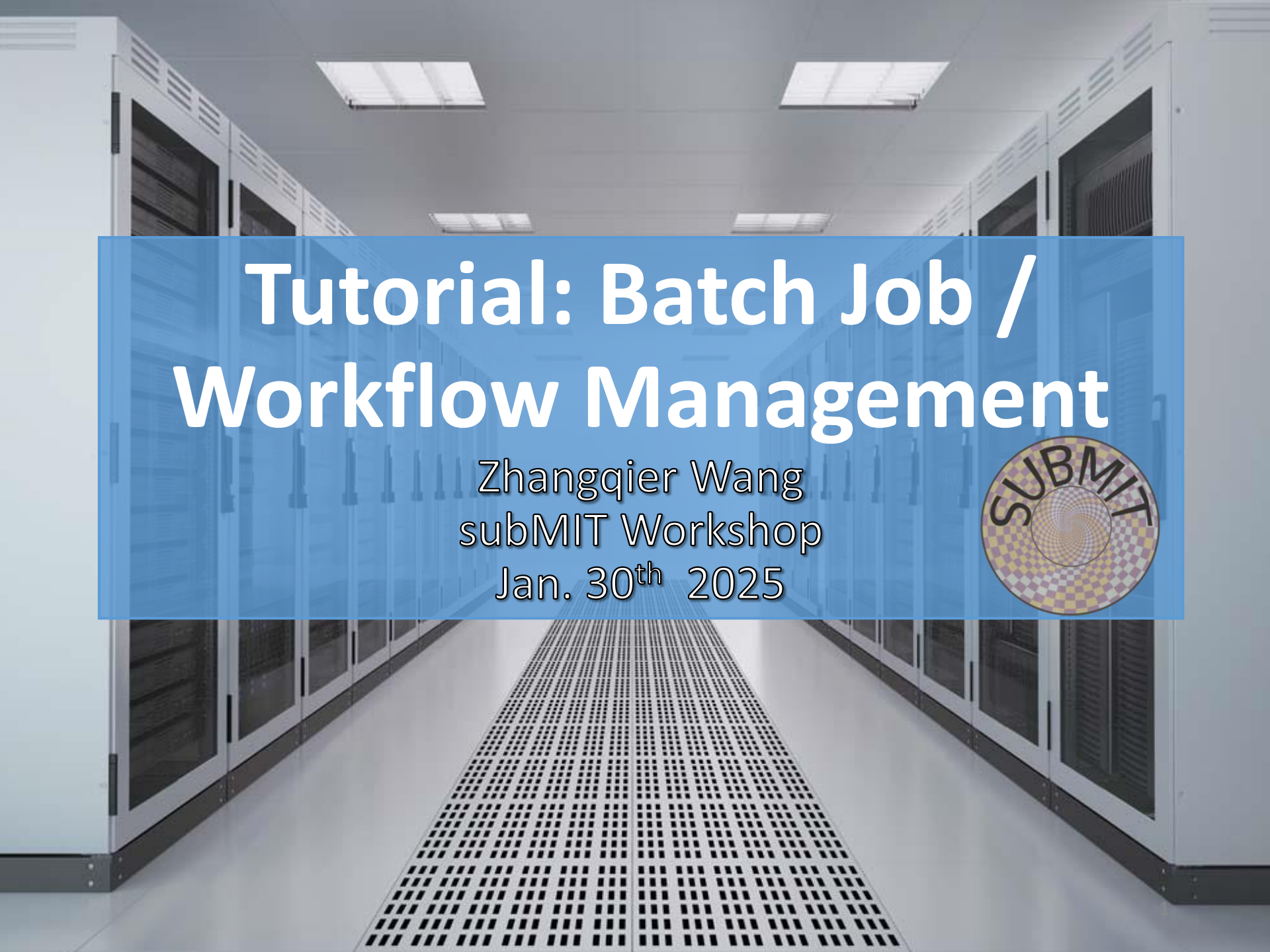


Tutorial: Batch Job / Workflow Management

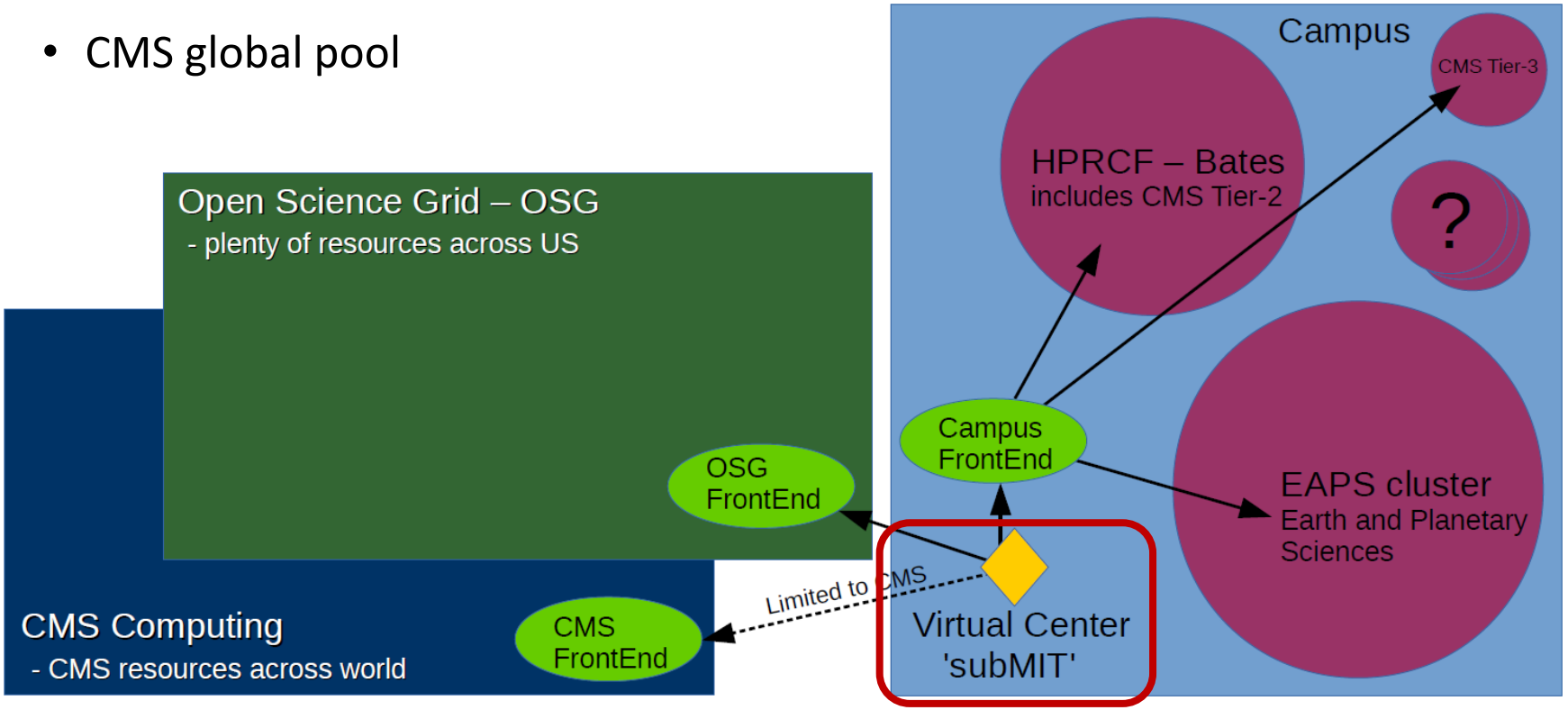
Zhangqier Wang

subMIT Workshop

Jan. 30th 2025

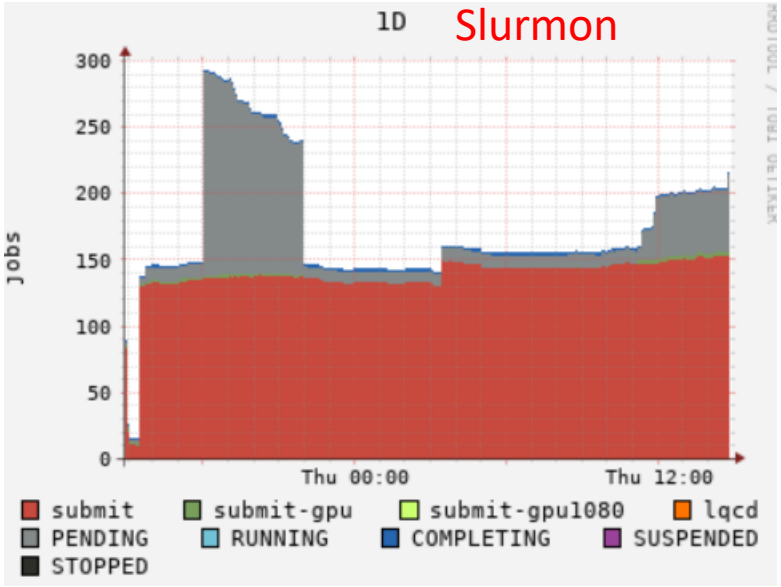
Computing Resources

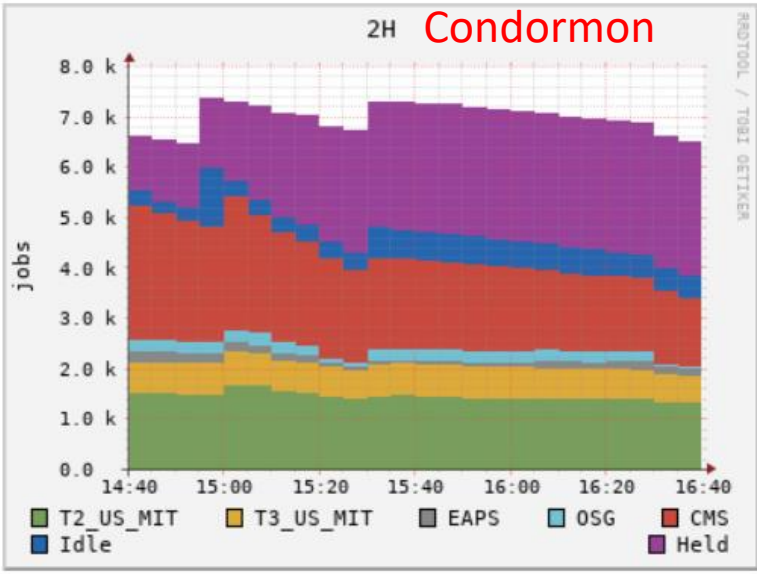
- The SubMIT connects to different external resources
- Campus FrontEnd: GlideinWMS
  - Tier2, Tier3, EAPS Clusters (not available yet)
- OSG: open science pool
- CMS global pool



Batch Jobs

- Local resources via “slurm”
  - 1000+ cores, 50+ GPUs
- External resources: access via “htcondor”
  - Campus: T3 (1000 cores), T2 (>10000 cores) , EAPS
  - OSG pool: dozens of campus, national labs, and non-profit organizations
  - CMS global pool: CMS sites worldwide
- Could reach 10k+ jobs running at once





Slurm

Slurm

- SLURM is an open-source workload manager and job scheduler for allocating resources on large compute clusters.
- It runs on subMIT machines, where /home, /work, /ceph are mounted
- The job is owned by user
- Basic Job Management Commands
  - **Submit a Job:** sbatch job_script.slurm
  - **Cancel a Job:** scancel <job_id>
  - **Hold a Job:** scontrol hold <job_id>
  - **Release a Held Job:** scontrol release <job_id>
- Monitoring Jobs
  - **Check All Job Statuses:** squeue
  - **Check Your Jobs Only:** squeue -u $USER
  - **Show Detailed Job Info:** scontrol show job <job_id>

Slurm Job Script

- More examples can be found in user guide link.
- Example script: submit.sh

```
#!/bin/bash
#
#SBATCH --job-name=test
#SBATCH --output=res_%j.txt
#SBATCH --error=err_%j.txt
#
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=100
```

Job set up

```
srun hostname
srun ls -hrlt
```

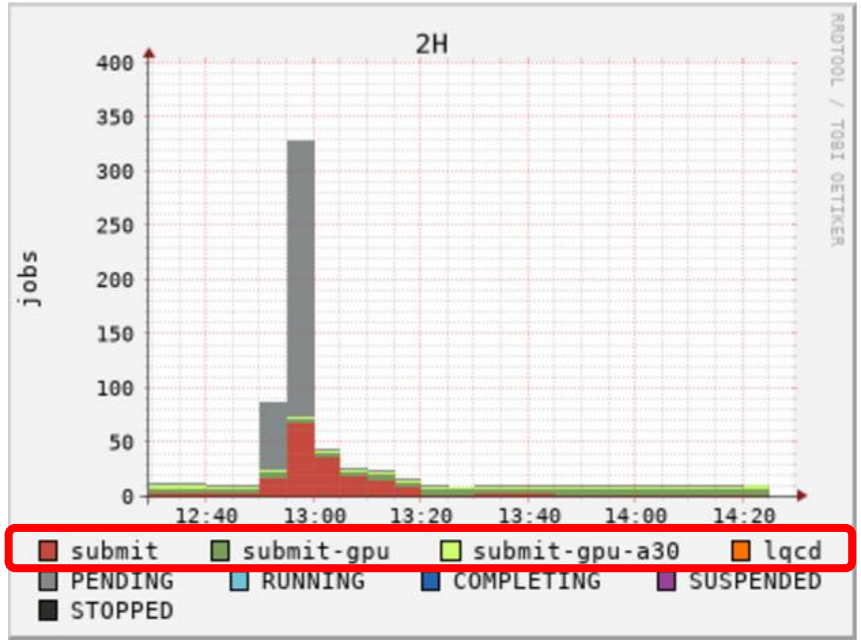
What you want to run

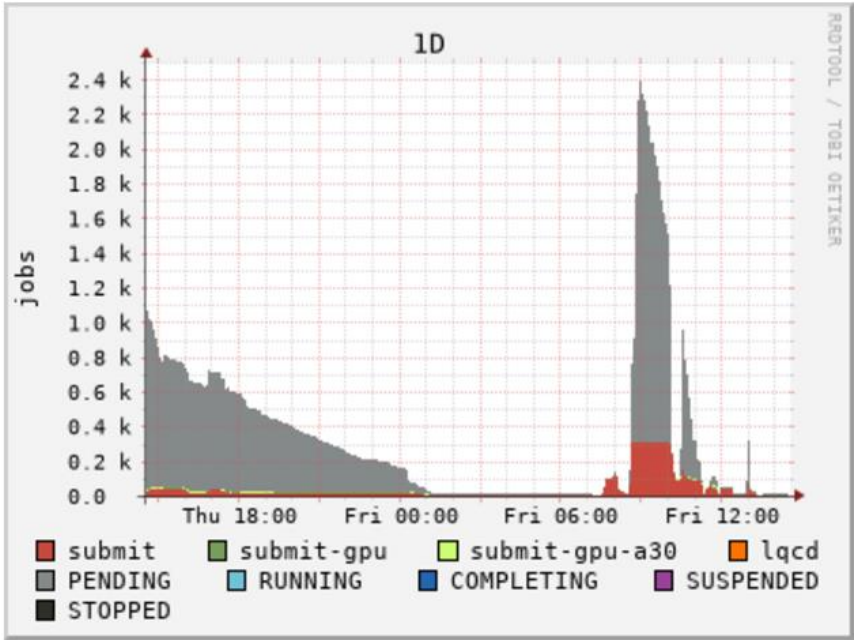
- Submit your job

```
sbatch submit.sh
```

```
[wangzqe@submit01 slurm]$ sbatch submit.sh
Submitted batch job 2007385
[wangzqe@submit01 slurm]$ squeue -u wangzqe
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2007385    submit     test  wangzqe  R       0:03      1 submit07
```

Slurm Partitions





- There are several slurm partitions holding different types of resources

Submit:

```
#SBATCH --partition=submit
```

submit-gpu:

```
#SBATCH --partition=submit-gpu
```
(1080 GPUs)

```
#SBATCH --partition=submit-gpu-a30
```
(A30 GPUs)

Customized Script

```
#!/bin/bash
#
#SBATCH --job-name=test
#SBATCH --output=res_%j.txt
#SBATCH --error=err_%j.txt
#
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=100

srun hostname
srun ls -hrlt
```

Job set up

Replace with your job content

- Run command, run your own executable scripts, etc.

Command you run interactively

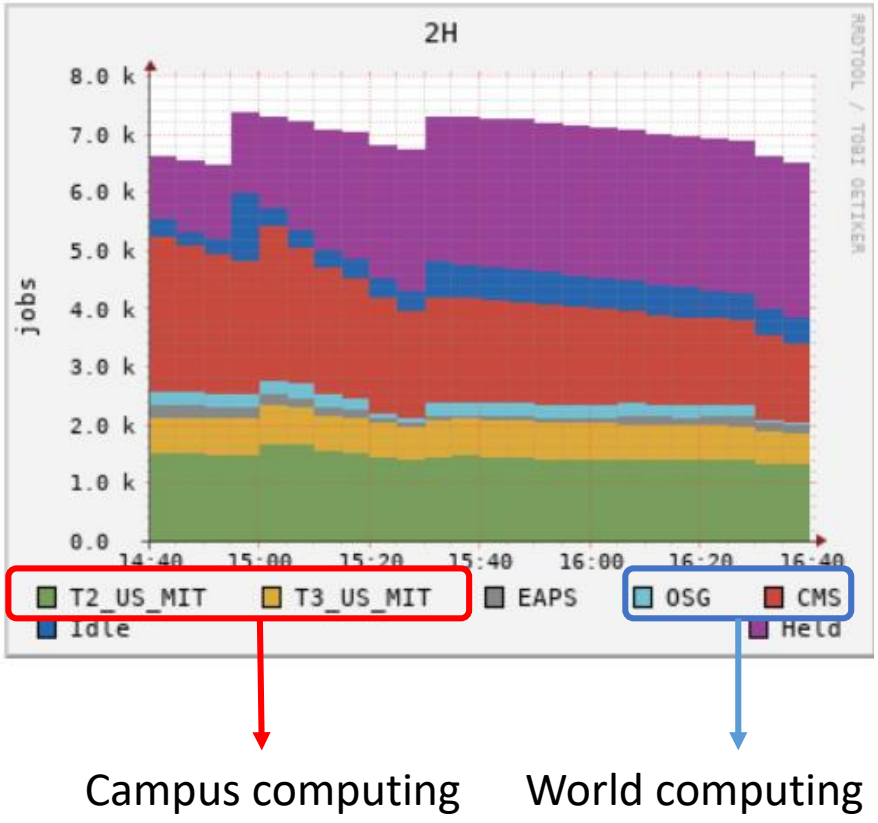
Slurm pool is local on subMIT, relative easier to use
Scale level ~100 jobs, short time < 2 days

HTCondor

Condor

- HTCondor, AKA condor, is a powerful workload management system. Connect to the external computing resources

- It runs outside the subMIT
  - No user account
  - No /home, /work, /ceph
  - More computing available
  - Job owned by condor service account
- /cvmfs mounted
  - container/singularity
- Job self contained
  - Input/output transfered with the job



Condor pool is external
Scale level >1k jobs
Less busy compared to Slurm pool

Condor Submission

- Basic condor script, example “condor.sub”:

```
universe              = vanilla
request_disk          = 1024
executable            = script.sh
arguments             = $(ProcId)
output                = $(ClusterId).$(ProcId).out
error                 = $(ClusterId).$(ProcId).err
log                   = $(ClusterId).$(ProcId).log
+DESIRED_Sites        = "mit_tier3"
queue 1
```

- Executable script “script.sh”:

```
#!/bin/bash

echo "I am a HTCondor job!"
echo "I have landed in $hostname"
echo "I have recieved parameter $1"
echo "That's all!"
```

Two components
- Condor submit file
- Your job files

- Submit the job:

```
condor_submit condor.sub
```

Campus Computing

- Connect via glideinWMS, can set up in condor submit file.

Glidein submission to T2/T3

Submit jobs to the T2 cluster by adding following to the HTCondor submission script:

```
+DESIRED_Sites = "mit_tier2"
```

If instead you want to run on the T3 machines you can replace the "+DESIRED_Sites" to:

```
+DESIRED_Sites = "mit_tier3"
```

If you want to submit to both T2 and T3, do:

```
+DESIRED_Sites = "mit_tier2,mit_tier3"
```

To submit GPU jobs, you need to add:

```
RequestGPus=1
```

To submit multi-core jobs, you need to add (4-core job for example, maximum 8):

```
RequestCpus=4
```

Global Computing

Jobs submission to CMS global pool

If you are a CMS member you can also go through the US CMS global pool. Here is an example sample list of sites you can use,

```
+DESIRED_Sites = "T2_AT_Vienna,T2_BE_IIHE,T2_BE_UCL,T2_BR_SPRACE,T2_BR_UERJ,T2_CH_CERN,T2_CH_CERN_AI,T2_CH_CERN
```

In order to use the CMS global pool, you will need to add a few additional lines to your submission script. The lines below with the proper ID and username (uid and id from subMIT) are necessary in order to get into the global pool:

```
use_x509userproxy      = True
x509userproxy          = /<path>/x509up_u<uid>
+AccountingGroup       = "analysis.<username>"
```

If you wish to submit jobs to GPU machines, you need to add additional lines in the script:

```
RequestGPus=1
+RequiresGPU=1
```

Jobs submission to OSG pool

Finally, you can also use OSG,

```
+ProjectName             = "MIT_submit"
```

General Tips

- More details in tutorial link

General Tips for HTCondor Jobs

Transferring Scripts and Data

via the submission script

File size < 200 MB

```
transfer_input_files    = <your comma-separated list of files>
transfer_output_remaps = "out.out = /work/submit/$USER/out.out"
```

via XRootD

Need x509 certificate

```
xrdcp root://submit50.mit.edu//data/user/w/wangzqe/test.txt .
xrdcp <your output> root://submit50.mit.edu//data/user/w/wangzqe/
```

size could be GB scale

- Software from CVMFS. Get singularity for example

```
+SingularityImage       = "/cvmfs/singularity.opensciencegrid.org/htc/rocky:9"
```

Condor Example

- An script that includes all the components mentioned
  - All files: https://github.com/mit-submit/submit-examples/tree/main/htcondor/test-basic
- "test.sub"

```
universe = vanilla
executable = first_test.sh
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
transfer_input_files = first_test_inputs.tar.gz
transfer_output_files = first_test_output_$(Process).txt
output = log/$(Process).stdout
error = log/$(Process).stderr
log = log/$(Process).log
+REQUIRED_OS = "rhel7"
+DESIRED_Sites = "mit_tier3"
+SingularityImage = "/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest"
rank = Mips
arguments = "$(Process)"
queue 10
```

Input, output transfer

Logs for debug/detail check

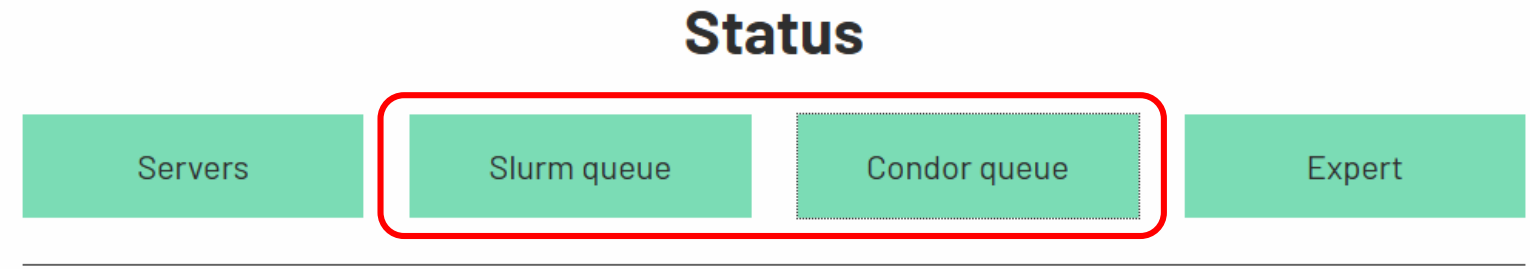
The osg version is in test_osg.sub

use centos 7 singularity

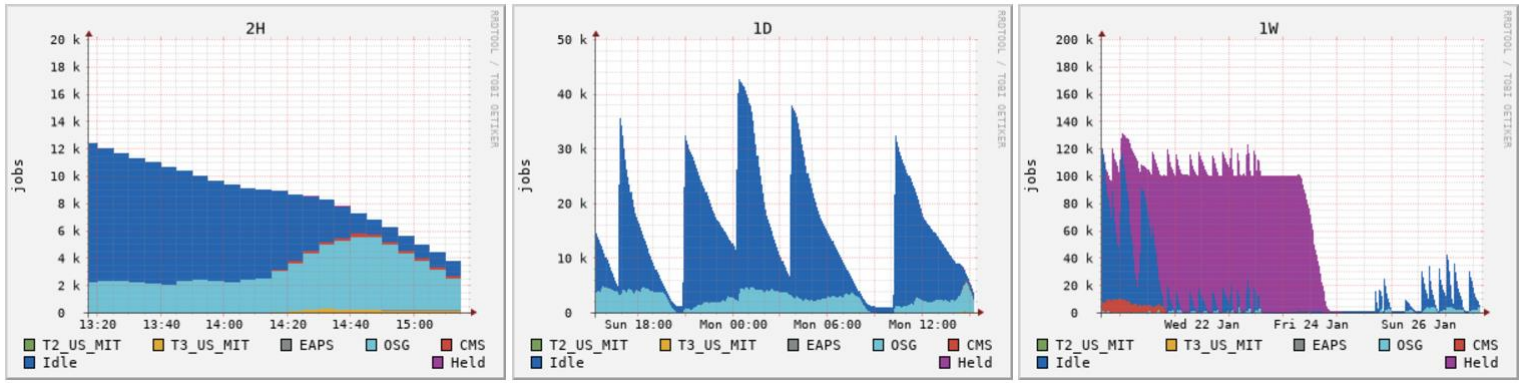
condor_submit test.sub = Submit 10 jobs, in each job, it runs: source first_test.sh $(Process)

Monitoring

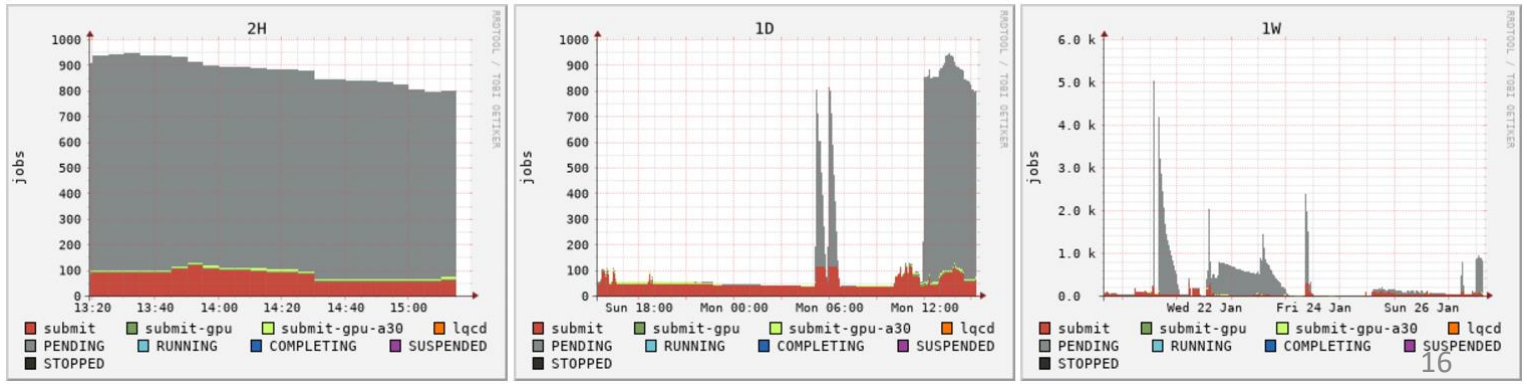
- https://submit.mit.edu/



condor



slurm

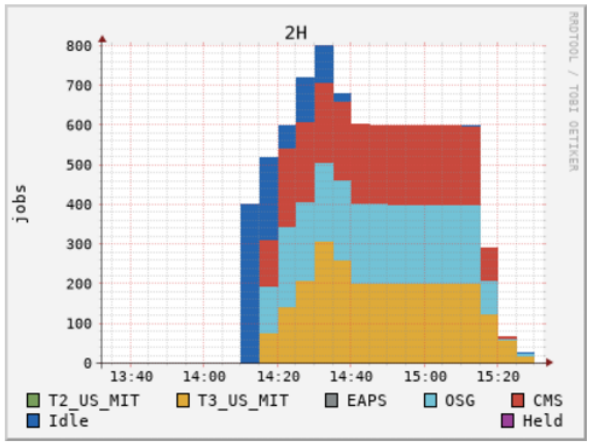


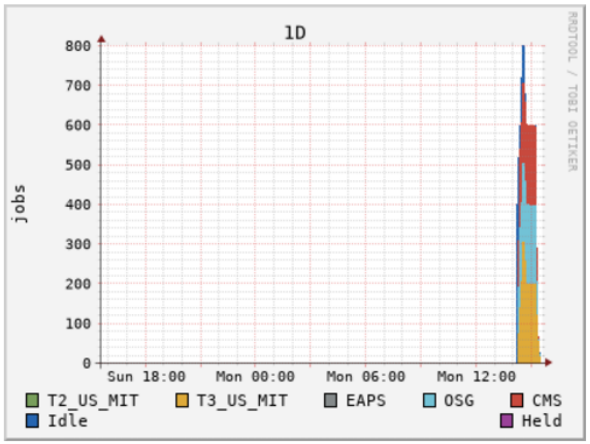
Monitoring Example

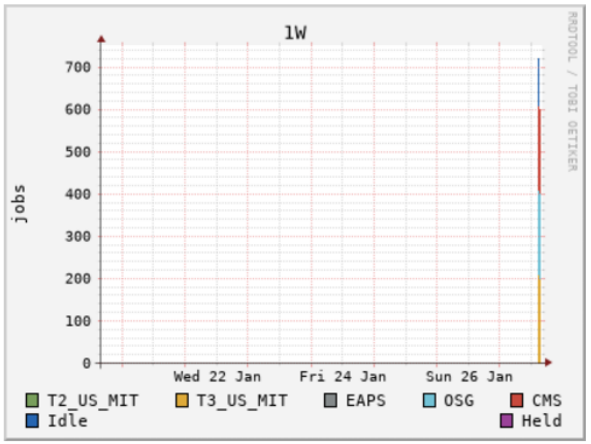
- Example of submitting ~20 jobs to MIT Tier 3, OSG pool, and CMS global pool.

User	Idle	Held	Running	MIT			OSG	CMS	Total
				T2_US_MIT	T3_US_MIT	EAPS			
wangzqe	0	0	23	0	17	0	5	1	23
sahughes	0	0	912	0	0	2	910	0	912
lavezzo	0	0	3	0	0	0	0	3	3
kudinoor	10	0	0	0	0	0	0	0	10
mori25	20	0	0	0	0	0	0	0	20
matzeni	1	0	0	0	0	0	0	0	1
akanugan	1000	0	0	0	0	0	0	0	1000
ceballos	0	0	0	0	0	0	0	0	0
Total	1031	0	938	0	17	2	915	4	1969

wangzqe

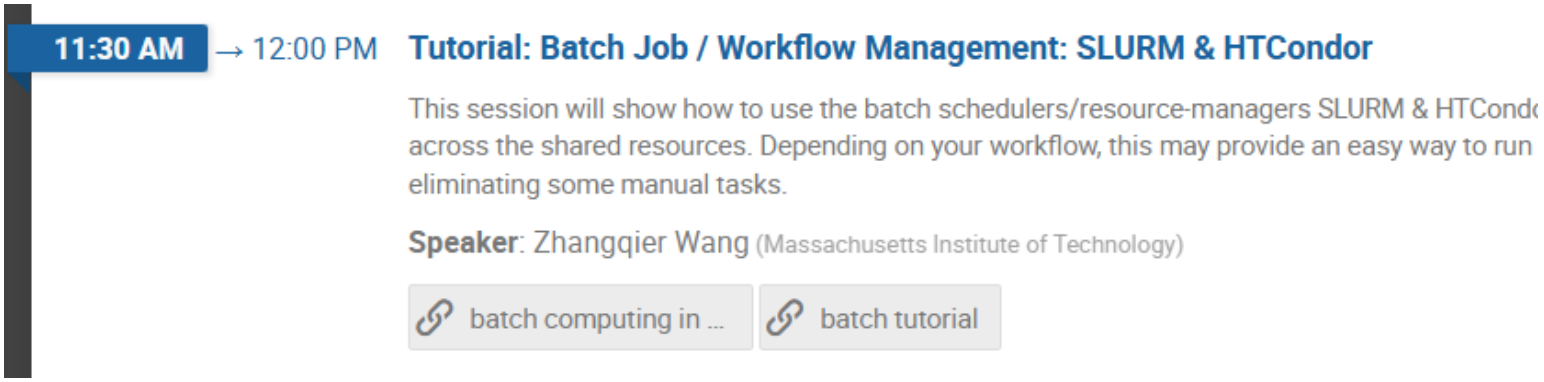






Exercise, Q&A

- Follow the tutorial instructions, have an exercise on slurm and condor submission



- Also more examples available on our github repo:

https://github.com/mit-submit/submit-examples