



Getting Help: Getting the Most out of User Support

Tuesday, August 8, 2023

Agenda

- Workflow
- Available Resources
- Identify the most appropriate resource
- Helpdesk
- Common Issues

Workflow

- Understand you problem
 - Identify the issue.
 - Personal issue:
 - General issues: access issue, permissions issue, code issue, scheduler, etc
 - System Issue
 - file system not available, resource not available
 - Is the specific error message
 - Ex. If the running jobs is returning an error
 - Is it code dependent
 - Is it system dependent
- Identify the correct support

Available Resources

- What are the available resources
 - Yourself
 - Colleagues
 - Web (community forums, user guides, git repositories)
 - Helpdesk

Identify the most appropriate resource

- Internal
 - Yourself, Colleagues
 - Sanity check, Legacy Lab Code
 - The Web
 - Generic errors
- External
 - The Software Provider
 - Software specific error or bug
 - Github documentation
 - The Resource Provider
 - Resource specific question (file systems, nodes)
 - Performance
 - At the end of your rope 😊

Help Desk

- Help Desk
 - While help desk staff are exceptional, they should be considered general practitioners
 - Provide adequate information for helpdesk to start investigating
 - Username, Account, System, Jobid, specific error message if available, etc.
 - sacct, file paths, submit scripts
 - Include nodes in output file name
 - The user with the problem should submit the help ticket
 - Always be nice to the support desk! 😊

Help Desk: Useful Information

- Resource, UserID/Username, Account/Project/Allocation
- JobId:
 - List of Node(s) that job ran on
 - Working directory (submit script name)
 - Location of .err and .out files
 - Project
 - Start time
 - End time
 - Resources requested

Slurm tools for job information:

- SLURM: Cluster management and job scheduling system
- Slurm Tools available for job information
 - Squeue
 - Used to view information about queued or running jobs
 - `>squeue -j 23646362`
 - scontrol
 - Used is used to view Slurm configuration for queued or running jobs
 - `>scontrol show job 2425622`
 - sacct for completed jobs
 - Can be used to displays accounting data for jobs and job steps
 - `>sacct -j 24256222 -all`

For additional assistance with tools use the
`>man <<command>>`

Common Issues: Resource Access

- Access
 - Password and username issues (ACCESS, UCSD)
 - Indicator message: Enter verification code
 - <https://allocations.access-ci.org/profile> (if username is not available for the resource then the account has not been created yet) Generally it takes about 1 business day for accounts to be fully functional.
 - ssh keys
 - Indicator message: Enter password
- Unable to access system
 - Check specific error message
 - User News
 - Stay Subscribed to be notified
 - Check on User Portal -- <https://support.access-ci.org/announcements>

Common Issues: Job Submissions

- Allocations
- Job Scripts
 - Error message
 - sbatch: error: Project balance is not enough to run the job
 - sbatch: error: QOSMaxNodePerJobLimit
 - sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)
 - Queues/Partitions
 - Time Limits
 - OOM
 - New Software stack
 - Module spider
 - Expanse-client tool

Common Issues

- Expanse-client tool

```
[nickel@login01 ~]$ expanse-client
Allows querying the user statistics.
```

Usage:

```
expanse-client [command]
```

Available Commands:

```
completion  Generate the autocompletion script for
help        Help about any command
project     Get project information
resource    Get resources
user        Get user information
```

Flags:

```
-a, --auth      authenticate the request
-h, --help      help for expanse-client
-p, --plain     plain no graphics output
-v, --verbose   verbose output
```

Use "expanse-client [command] --help" for more information about a command.

```
[nickel@login01 ~]$ expanse-client user nickel -r expanse_gpu -p
```

Resource		expanse_gpu					
NAME	STATE	PROJECT	TG PROJECT	USED	AVAILABLE	USED BY PROJECT	
nickel	allow	ddp324		0	5000	21	
nickel	allow	ddp386		0	2500	74	
nickel	allow	sds154	TG-ASC150024	0	100	517	
nickel	allow	sds166	TG-STAl60003	0	2500	3	
nickel	allow	use300		9	269000	63638	

```
[nickel@login01 ~]$ ^C
```

```
[nickel@login01 ~]$ expanse-client user nickel -p
```

Resource		expanse					
NAME	STATE	PROJECT	TG PROJECT	USED	AVAILABLE	USED BY PROJECT	
nickel	allow	ddp386		2	110000	9163	
nickel	allow	sds154	TG-ASC150024	0	1000	16572	
nickel	allow	sds166	TG-STAl60003	0	100000	56528	
nickel	allow	use300		5856	5050000	3457273	

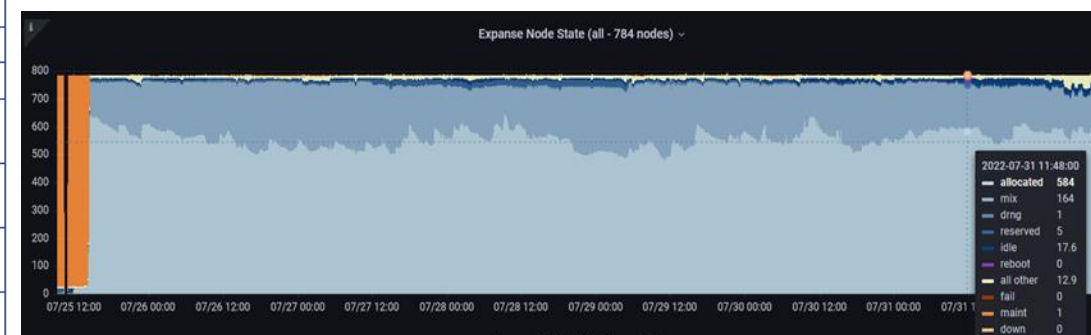
```
[nickel@login01 ~]$ expanse-client resource
```

Available resources:

```
expanse
expanse_gpu
expanse_industry
expanse_industry_gpu
```

Common Issues: Queue and Time Limits

Partition Name	Max Walltime	Max Nodes/Job	Max Running Jobs	Max Running + Queued Jobs	Charge Factor	Notes
compute	48 hrs	32	32	64	1	Exclusive access to regular compute nodes; <i>limit applies per group</i>
ind-compute	48 hrs	32	32	64	1	Exclusive access to Industry compute nodes; <i>limit applies per group</i>
shared	48 hrs	1	4096	4096	1	Single-node jobs using fewer than 128 cores
ind-shared	48 hrs	1	32	64	1	Single-node Industry jobs using fewer than 128 cores
gpu	48 hrs	4	4	8 (32 Tres GPU)	1	Used for exclusive access to the GPU nodes
ind-gpu	48 hrs	4	4	8 (32 Tres GPU)	1	Exclusive access to the Industry GPU nodes
gpu-shared	48 hrs	1	24	24 (24 Tres GPU)	1	Single-node job using fewer than 4 GPUs
ind-gpu-shared	48 hrs	1	24	24 (24 Tres GPU)	1	Single-node job using fewer than 4 Industry GPUs
large-shared	48 hrs	1	1	4	1	Single-node jobs using large memory up to 2 TB (minimum memory required 256G)
debug	30 min	2	1	2	1	Priority access to shared nodes set aside for testing of jobs with short walltime and limited resources
gpu-debug	30 min	2	1	2	1	Priority access to gpu-shared nodes set aside for testing of jobs with short walltime and limited resources; <i>max two gpus per job</i>
preempt	7 days	32		128	.8	Non-refundable discounted jobs to run on free nodes that can be pre-empted by jobs submitted to any other queue
gpu-preempt	7 days	1		24 (24 Tres GPU)	.8	Non-refundable discounted jobs to run on unallocated nodes that can be pre-empted by higher priority queues



https://www.sdsc.edu/support/user_guides/expansive.html#running

Common Issues: Charging

- Charging is based on what is requested, not how resources are used
- Charging is based on the Maximum of memory and CPU core fraction
- Example for CPU
 - $\text{Max} [3600 * \text{\#CPU cores}, 1800 * \text{\#Mem in GB}] / 3600 * (\text{wallclock time in secs} / 3600)$
- Minimum charge for any job is 1SU

Common Issues: Job Status

- `squeue --` reports status and reason codes
 - Queued Jobs

```
[nickel@login01 ~]$ squeue | more
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST (REASON)
13574113	compute	80dgree_	yweng3	PD	0:00	2	(MaxMemPerLimit)
12668967	compute	0-xtensi	kavousi	PD	0:00	1	(MaxMemPerLimit)
14756880	compute	job001_p	amysai	PD	0:00	10	(Reservation)
14800161	compute	namd-com	sasadian	PD	0:00	6	(QOSMaxCpuPerUserLimit)
14800218	compute	namd-com	sasadian	PD	0:00	6	(QOSMaxCpuPerUserLimit)
14789098	compute	bl_8JHNp	uscms	PD	0:00	1	(MaxJobsPerAccount)

- Running jobs

14813206	compute	post0110	lpegolot	R	16:30:28	1	exp-9-35
14800090	compute	namd-com	sasadian	R	16:13:01	6	exp-2-29,exp-3-23,exp-4-33,exp-7-20,exp-9-[03,26]
14764467	compute	V1WTReRU	aminkvh	R	16:08:56	1	exp-2-54
14773832	compute	V4R1639Q	aminkvh	R	15:55:58	1	exp-8-14
14800092	compute	namd-com	sasadian	R	15:29:28	6	exp-4-29,exp-7-[07,39-40],exp-9-[28,41]
14812166	compute	scratch	mlaskow2	R	15:53:59	1	exp-10-20
14812167	compute	scratch	mlaskow2	R	15:39:34	1	exp-8-48
14800158	compute	namd-com	sasadian	R	15:17:18	6	exp-2-[26,50],exp-4-[52-53],exp-7-[42-43]
14812168	compute	scratch	mlaskow2	R	15:20:01	1	exp-10-37

Common Issues: Reasons

- **queue** – Common “reasons” for pending jobs
 - MaxMemPerLimit
 - Max. mem per Node = 243G
 - QOSMaxNodePerUserLimit
 - Priority
 - ReqNodeNotAvail, Unavailable nodes: exp-x-xx
- **File system not available**
 - We have added a slurm directive `#SBATCH --constraint = “lustre”` to indicate if your job is using the lustre file system. If this is provided, the scheduler will not launch the job on a node that is missing lustre.
- **System Maintenance**
 - <https://support.access-ci.org/outages>

Common Issues: Software

- Available Software
 - Module spider
 - Module spider <<software_package>>
- Software installs
 - Help desk
- Software specific error messages
 - Github repo issues file

Allocations

- SDSC allocates resources via three methods ACCESS-CI, HPC@UC, and Industrial program
 - ACCESS-CI: (Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support) is an NSF advanced computing and data resource. ACCESS-CI maintains an innovative, agile, integrated, robust, trustworthy and sustainable CI ecosystem including Compute, Storage and Cloud resources to support S&E research and education. (<https://access-ci.org/>)
 - HPC@UC: Available to UC folks to gain quick access to limited compute and storage resources with goal to graduate to ACCESS proposal
 - Industrial Partners: Purchase SUs on our Industry Rack

ACCESS-CI: Expanse Allocation

- Trial allocation: 100 GPU and/or 1000 CPU hours
 - consult@sdsc.edu
- Apply for allocations through ACCESS
 - <https://allocations.access-ci.org/prepare-requests-overview>

Allocation	Credit Threshold
<u>Explore ACCESS</u>	400,000
<u>Discover ACCESS</u>	1,500,000
<u>Accelerate ACCESS</u>	3,000,000
<u>Maximize ACCESS</u>	Not awarded in credits.

Allocation Levels

Opportunity	Explore	Discover	Accelerate	Maximize
Purpose	Resource evaluation, grad student projects, small classes and training events, benchmarking, code development and porting, similar small-scale uses.	Grants with modest resource needs, Campus Champions, large classes and training events, NSF graduate fellowships, benchmarking and code testing at scale, gateway development.	Mid-scale resource needs, consolidating multi-grant programs, collaborative projects, preparation for Maximize ACCESS requests, gateways with growing communities.	Large-scale research projects.

HPC@UC

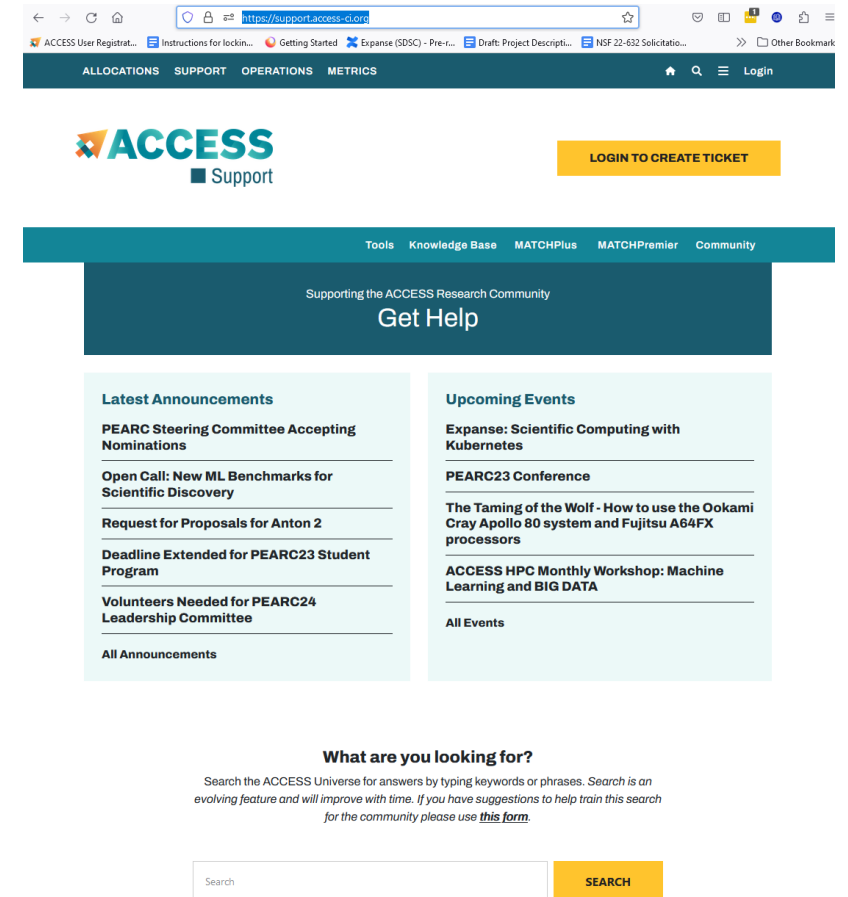
- Request HPC@UC at: https://www.sdsc.edu/support/hpc_uc_apply-exp.html
- Up to 500K core-hours of computing, associated data storage, and access to SDSC expertise to assist their research team.
- Awards are active for one year. NO supplements, renewals or Extensions
- Applicants must not have an active ACCESS award
- Developed to support onboarding to ACCESS and larger, formal allocation requests
- SDSC staff will assist in developing these allocation applications.
- Applications are reviewed on an ongoing basis. Applicants will be notified within 10 business days of the review decision.
- https://www.sdsc.edu/collaborate/hpc_at_uc.html

How much time do I need

- 1 CPU and less than 2GB of memory are charged 1 CPU Service Unit(1SU = 1 core/hour).
- 1 GPU and up to 10 CPUs and 92 GB of memory are charged 1 GPU Service Unit (SU)/hour. This job will be charged 1 GPU SU/hour.
- The minimum charge for any job is 1 SU.
- 1 Expanse SUs = 1 ACCESS Credit
- 1 Expanse GPU SU = 54 Expanse SUs (for conversion)

How to reach support

- consult@sdsc.edu
- <https://support.access-ci.org/>



Review

- Understand your problem
- Engage with appropriate support tools
- Provide relevant information to reduce iterations
- Always be nice to the support desk! 😊

Questions?