

Getting Started with

Dalma

High Performance Computing NYUAD



- A partition is a collection of nodes, they may share some attributes (CPU type, GPU, etc)
- Compute nodes may belong to multiple partitions to ensure maximum use of the system
- Partitions may have different priorities and limits of execution and may limit who can use them
- Dalma's partition (as seen by users)
 - **serial** run single core and multi-threaded jobs (eg single node)
 - **parallel** run MPI jobs (eg multi-node)
 - **bigmem** run jobs on large memory systems only
 - **butinah** run jobs on SSE2 (Westmere) processors only (for serial-integer apps)
 - **visual** run jobs on systems with a graphics card only
- There are other partitions, but these are reserved for specific groups and research projects
- For those who are experts in SLURM we use partitions to request GPUs, large memory, and visual instead of "constraints" as this approach gives us more flexibility for priorities and resource limits.

- To submit a job first you write a "job script"

```
#!/bin/bash
#SBATCH -p serial
#SBATCH -n 1
./myprogram
```

- Then you submit the script in any of the following manner

```
> sbatch job.sh
```

OR

```
> sbatch < job.sh
```

OR

```
> sbatch << EOF
#!/bin/bash
#SBATCH -p serial
#SBATCH -n 1
./myprogram
EOF
```

- Arguments to "sbatch" can be put on the command line or embedded in the job script
- Putting them in the job script is a better option as then it "documents" how to rerun your job

```
{  
#!/bin/bash  
#SBATCH -p serial  
#SBATCH -n 1  
./myprogram  
}
```

 job1.sh

```
> sbatch job1.sh
```

OR

```
{  
#!/bin/bash  
./myprogram  
}
```

 job2.sh

```
> sbatch -p serial -n 1 job2.sh
```

Common Job submission arguments:

-n	Select number of tasks to run (default 1 core per task)
-C	Select required system feature (eg avx2, sse, gpu)
-N	Select number of nodes on which to run
-t	Wallclock in hours:minutes:seconds (ex 4:00:00)
-p	Select partition (serial, parallel, gpu, bigmem)
-o	Output file (with no <code>-e</code> option, err and out are merged to the Outfile)
-e	Keep a separate error File
-d	Dependency with prior job (ex don't start this job before job XXX terminates)
-A	Select account (ex physics_ser, faculty_ser)
-c	Number of cores required per task (default 1)
--tasks-per-node	Number of tasks on each node
--mail-type=type	Notify on state change: BEGIN, END, FAIL or ALL
--mail-user=user	Who to send email notification
--mem	Maximum amount of memory per job (default is in MB, but can use GB suffix) (Note: not all memory is available to jobs, 8GB is reserved on each node for the OS) (So a 128GB node can allocate up to 120GB for jobs)

- **Submitting with dependencies:** Useful to create workflows
 - Any specific job may have to wait until any of the specified conditions are met
 - These conditions are set with `-d type:jobid` where type can be:
 - **after** run after <jobid> has terminated
 - **afterany** if <jobid> is a job array run after any job in the job array has terminated
 - **afterok** run after <jobid> if it finished successfully
 - **afternotok** run after <jobid> if it failed to finish successfully

```
# Wait for specific job array elements
sbatch --depend=after:123_4 my.job
sbatch --depend=afterok:123_4:123_8 my.job2

# Wait for entire job array to complete
sbatch --depend=afterany:123 my.job

# Wait for entire job array to complete successfully
sbatch --depend=afterok:123 my.job

# Wait for entire job array to complete and at least one task fails
sbatch --depend=afternotok:123 my.job
```

- Each submitted job is given a unique number
- You can list your jobs to see which ones are waiting (pending), running
- As well as how long a job has been running and on which node(s)

```
> squeue
      JOBID PARTITION    NAME    USER ST      TIME  NODES NODELIST(REASON)
435251_[1-50]  ser_std 151215_F u123 PD      0:00      1 (Priority)
435252_[1-50]  ser_std 151215_F u123 PD      0:00      1 (Priority)
  435294      ser_std Merge5.s u123 PD      0:00      1 (Priority)
435235_[20-50] ser_std 151215_F u123 PD      0:00      1 (Priority)
  435235_19    ser_std 151215_F u123 R      12:55      1 compute-21-8
  435235_17    ser_std 151215_F u123 R      47:34      1 compute-21-12
  435235_15    ser_std 151215_F u123 R      49:04      1 compute-21-7
  435235_13    ser_std 151215_F u123 R      50:34      1 compute-21-4
  435235_11    ser_std 151215_F u123 R      54:35      1 compute-21-9
  435235_9     ser_std 151215_F u123 R      56:35      1 compute-21-6
  435235_7     ser_std 151215_F u123 R      58:35      1 compute-21-5
  435235_5     ser_std 151215_F u123 R      59:36      1 compute-21-1
  435235_3     ser_std 151215_F u123 R     1:00:36      1 compute-21-11
  435235_1     ser_std 151215_F u123 R     1:04:37      1 compute-21-3
```

- You can look at completed jobs using the "sacct" command
- To look at jobs you ran since July 1, 2017

```
> squeue --starttime=2017-07-01
```

- You can retrieve the following informations about a job after it terminates:

JobID	JobIDRaw	JobName	Partition	MaxVMSize	MaxVMSizeNode
MaxVMSizeTask	AveVMSize	MaxRSS	MaxRSSNode	MaxRSSTask	AveRSS
MaxPagesNode	MaxPagesTask	AvePages	MinCPU	MinCPUNode	MinCPUTask
NTasks	AllocCPUS	Elapsed	State	ExitCode	AveCPUFreq
MaxPages	AveCPU	ReqCPUFreqMin		ReqCPUFreqMax	
ReqCPUFreqGov	ReqMem	ConsumedEnergy		MaxDiskRead	
MaxDiskReadNode	MaxDiskReadTask	AveDiskRead		MaxDiskWrite	
MaxDiskWriteNode	MaxDiskWriteTask	AveDiskWrite		AllocGRES	
ReqGRES	ReqTRES	AllocTRES			

- To retrieve specific informations about a job

```
> sacct -j 466281 -format=partition,alloccpus,elapsed,state,exitcode
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
466281	job3.sh	par_std	cpcm_par	56	COMPLETED	0:0
466281.batch	batch		cpcm_par	28	COMPLETED	0:0
466281.0	env		cpcm_par	56	COMPLETED	0:0

- You can see your job's progress by looking at the output and error files
- By default output and error files are named "slurm-XXX.out" and "slurm-XXX.err" where XXX is the job id
- "tail -f" allows you to track new output as it is produced

```
> cat slurm-435563.out
```

```
> more slurm-435563.out
```

```
> tail -f slurm-435563.out
```

- Sometimes you need to kill your job when you realise it is not working as expected
- Note that your job can be killed automatically when it reaches its maximum time/memory allocation

```
> scancel 435563
```

SLURM "tasks"

```
#SBATCH -n 2 VS #PBS -l nodes=1:ppn=2
```

In SLURM users specify how many tasks - not cores! - they need (-n). Each task by default uses 1 core. But this can be redefined by users using the "-c" option.

For example #SBATCH -n 2 is requesting 2 cores, while #SBATCH -c 3 #SBATCH -n 2 is requesting 6 cores.

On Dalma/SLURM we implement an exclusive policy on nodes being used to run parallel jobs - eg no other jobs may run on nodes allocated for running parallel jobs.

When submitting parallel jobs on Dalma you need not specify the number of nodes. The number of tasks and cpus-per-task is sufficient for SLURM to determine how many nodes to reserve.

Sometimes applications require a list of nodes where they are to run in parallel to start.

SLURM keeps the list of nodes within the environment variable "\$SLURM_JOB_NODELIST".

To retrieve the list of nodes in a PBS format file use "generate_pbs_nodefile".

For example:

```
export NODEFILE=$(generate_pbs_nodefile)
```

Then \$NODEFILE contains the name of a temporary file containing the list of nodes used following the PBS hostfile format.

SLURM maintains user associations which include user, account, qos, and partition. Users may have several associations. Moreover, accounts are hierarchical. For example, account "physics" maybe be a sub-account of "faculty", which may be a sub-account of "institute", etc.

When submitting jobs users with multiple associations must explicitly list the account, qos, partition details they wish to use.

```
sbatch -p serial -a physics -q normal -u benoit job
```

Some specific job submission tools extend SLURM's associations to define a "default" association. So you only need to specify accounts is, for example, you belong to multiple accounts - ex faculty and research-lab - and you want to execute using your non-default account. So at most you'll need to specify:

```
sbatch -p <partition> -a <account> job
```

Moreover, accounts, partitions, qos and users may each be configured with resource usage limits. Thus the administrators can impose limits to the number of jobs queued, jobs running, cores usage, and run time.

To see you SLURM associations (and their parents) as well as your resource usage limits use the following Dalma specific tool:

```
> slurm-show-my-limits.sh
```

User	Account	Partition	QOS	GrpSubmit	GrpJobs	GrpTRES	MaxTRES	MaxJobs	MaxSubmit	MaxWall	Par Name
benoit	qaad_par	par_std	par_std			cpu=700	cpu=700	100	200	12:00:00	
benoit	qaad_ser	visual	visual					100	200	12:00:00	
benoit	qaad_ser	butinah	butinah					100	200	12:00:00	
benoit	qaad_ser	bigmem	bigmem			cpu=10		100	200	12:00:00	
benoit	qaad_ser	preempt_std	preempt_std					100	200	12:00:00	
benoit	qaad_ser	ser_std	ser_std			cpu=70		100	200	12:00:00	
	qaad_par		normal			cpu=112		100	200	12:00:00	qaad
	qaad		normal			cpu=252		100	200		nyuad
	nyuad		normal	20000	1000	cpu=6608		100	200		root
	root		normal								
	qaad_ser		normal			cpu=140		100	200	12:00:00	qaad
	qaad		normal			cpu=252		100	200		nyuad
	nyuad		normal	20000	1000	cpu=6608		100	200		root
	root		normal								

```
> slurm-show-my-limits.sh
```

User	Account	Partition	GrpSubmit	GrpJobs	GrpTRES	MaxTRES	MaxJobs	MaxSubmit	MaxWall	Par	Name
benoit	qaad_par	par_std			cpu=700	cpu=350	100	200	12:00:00		
benoit	qaad_ser	visual					100	200	12:00:00		
benoit	qaad_ser	butinah					100	200	12:00:00		
benoit	qaad_ser	bigmem			cpu=10		100	200	12:00:00		
benoit	qaad_ser	preempt_std					100	200	12:00:00		
benoit	qaad_ser	ser_std			cpu=70		50	75	6:00:00		
	qaad_par				cpu=1000		100	200	12:00:00		qaad
	qaad				cpu=252		100	200			nyuad
	nyuad		20000	1000	cpu=6608		100	200			root
	root										
	qaad_ser				cpu=140		100	200	12:00:00		qaad
	qaad				cpu=252		100	200			nyuad
	nyuad		20000	1000	cpu=6608		100	200			root
	root										

In this output we see:

- user "benoit" can submit up to 200 jobs on "par_std" (parallel) partition, but have at most 100 jobs running consuming a maximum of 700 cores total where each jobs is limited to a maximum of 350 cores for 12 hours
- user "benoit" can submit up to 75 jobs on "ser_std" (serial) partition, with at most 50 jobs running using a total of up to 70 cores for up to 6 hours
- account "qaad_par" is shared with other users and together they have a limit of 1000 cores, 200 jobs queued, and 100 jobs running (eg the sum of all cores used by running jobs using account "qaad_par" can't exceed 1000 cores)
- account "qaad_ser" is shared with other users and together they have a limit of 140 cores, 200 jobs queued, and 100 jobs running
- account "qaad" is a sub-account of "nyuad" and the sum of all parallel and serial jobs can't exceed 200 jobs queued, 100 jobs running and 252 cores

This next Dalma specific tool allows you to see how much resources you are using. This is useful when your job can't run because of "group resource limit" having been reached.

Here user "u123" has two accounts, "cpcm_par" and "cpcm_ser". On the "cpcm_par" (parallel partition) his limit is 1400 cores, and he's currently using 0 cores. However, other users from the same account are already using 336 cores out of the account maximum 5600.

The "cpcm_par" account is a sub-account of "cpcm", which currently is using 336 cores out of the 5600 permitted.

The "cpcm" account is also a sub-account of "institute". All "institute" users are presently using 846 cores out of the 6608 cores account limit.

Finally "institute" is a sub-account of "nyuad" where 5876 cores are being used - nearly at the 6608 limit.

```

slurm_show_usage V1.0 - 2017 NYUAD Proprietary Software

User line show the usage and limit for user=u123 within each account it belong.
Account lines show the usage and limit for all users in that account.
The indentation on Account lines represent the level of sub-accounts
up to the present account.

-----
Usage  Limit  TYPE   Account
-----
  0     1400   User   cpcm_par
336     5600   Account cpcm_par
336     5600   Account cpcm
846     6608   Account institute
5873    6608   Account nyuad
-----

  0     140   User   cpcm_ser
  0     280   Account cpcm_ser
336     5600   Account cpcm
846     6608   Account institute
5873    6608   Account nyuad
-----
Login-0-3.local 7: █

```


SLURM: account usage

The "slurm_show_usage" tool has an option to show you which account level would prevent you to run a job.

The "-n 800" option will show which account(s) would exceed the user or account core limit if you were to submit a job requiring 800 cores.

```

benoit — ssh -Y benoit@dalma.abudhabi.nyu.edu — 85x29
[ui23@login-0-1 ~]$ slurm_show_usage -n 800
-----
slurm_show_usage V1.0 - 2017 NYUAD Proprietary Software

Checking for 800 cores availability
Highlighted numbers show user / account limits that would
prevent a 800 core job to run based on current usage.

User line show the usage and limit for user=ui23 within each account it belong.
Account lines show the usage and limit for all users in that account.
The indentation on Account lines represent the level of sub-accounts
up to the present account.

-----
Usage  Limit  TYPE  Account
-----
  0      1400  User   cpcm_par
 336     5600  Account cpcm_par
 337     5600  Account cpcm
 819     6608  Account institute
5863    6608  Account nyuad
-----

  0      140  User   cpcm_ser
  1      280  Account cpcm_ser
 337     5600  Account cpcm
 819     6608  Account institute
5863    6608  Account nyuad
-----
login-0-3.local 17:
    
```

SLURM: account usage

The "-a" option will show all accounts usage and limit on Dalma, as well as their current usage.

The usage limits are defined by the academic steering committee in order to meet each group's computational needs, while allowing fairness to all groups.

The account limits are periodically revised based on prior usage statistics and inputs from the research groups about new project requirements.

Thus, the HPC support team role is limited to implementing the recommendations from the steering committee and to provide the steering committee with statistics and other key informations that help them define fair resource usage rules.

```

benoit — ssh -Y benoit@dalma.abudhabi.nyu.edu — 82x62
[ul23@login-0-1 ~]$ slurm_show_usage -a
-----
slurm_show_usage V1.0 - 2017 NYUAD Proprietary Software

Account lines show the usage and limit for all users in that account.
The indentation on Account lines represent the level of sub-accounts
up to the present account.
-----
Usage  Limit  TYPE  Account
-----
 868 UNLIMITED  Account  condo_serdal
-----
2688   2800  Account  physics_par
2688   2800  Account  physics
4784   6608  Account  faculty
5891   6608  Account  nyuad
-----
 240    560  Account  spacescience_ser
 240   2800  Account  spacescience
 848   6608  Account  institute
5891   6608  Account  nyuad
-----
 28     280  Account  chemistry_ser
 672   2800  Account  chemistry
4784   6608  Account  faculty
5891   6608  Account  nyuad
-----
 336   5600  Account  cpcm_par
 338   5600  Account  cpcm
 848   6608  Account  institute
5891   6608  Account  nyuad
-----
 140   1400  Account  cgsb_par
 242   1400  Account  cgsb
 848   6608  Account  institute
5891   6608  Account  nyuad
-----
1400   2800  Account  engineering_par
1400   2800  Account  engineering
4784   6608  Account  faculty
5891   6608  Account  nyuad
-----
 644   2800  Account  chemistry_par
 672   2800  Account  chemistry
4784   6608  Account  faculty
5891   6608  Account  nyuad
-----
 102   1400  Account  cgsb_ser
 242   1400  Account  cgsb
 848   6608  Account  institute
5891   6608  Account  nyuad

```

The "dmap" tool (Dalma specific) will show you the utilization of each compute node on the cluster. The first numbers is a shorthand for the compute node name, so "12-3" actually means "compute-12-3". The second numbers represent the number of cores used and total number of cores in the system.

"white" highlight shows nodes that are down for maintenance.

"green" means a node is busy.

No highlight means a node is free.

```

benoit — ssh -Y benoit@dalma.abudhabi.nyu.edu — 156x53
*****
'dmap' showing:
*****
          cpus alloc, for all users, with all jobs
*****
[1-1 : 28/28] [1-2 : 28/28] [1-3 : 01/28] [1-4 : 00/28] [1-5 : 28/28] [1-6 : 28/28] [1-7 : 28/28] [1-8 : 28/28] [1-9 : 28/28]
[1-10 : 28/28] [1-11 : 28/28] [1-12 : 28/28] [1-13 : 28/28] [1-14 : 28/28] [1-15 : 28/28] [1-16 : 28/28] [1-17 : 28/28] [1-18 : 28/28]
[2-1 : 28/28] [2-2 : 28/28] [2-3 : 28/28] [2-4 : 28/28] [2-5 : 28/28] [2-6 : 28/28] [2-7 : 28/28] [2-8 : 28/28] [2-9 : 28/28]
[2-10 : 28/28] [2-11 : 28/28] [2-12 : 28/28] [2-13 : 28/28] [2-14 : 28/28] [2-15 : 00/28] [2-16 : 00/28] [2-17 : 28/28] [2-18 : 28/28]
[3-1 : 28/28] [3-2 : 28/28] [3-3 : 28/28] [3-4 : 28/28] [3-5 : 28/28] [3-6 : 28/28] [3-7 : 28/28] [3-8 : 28/28] [3-9 : 28/28]
[3-10 : 28/28] [3-11 : 28/28] [3-12 : 28/28] [3-13 : 28/28] [3-14 : 28/28] [3-15 : 28/28] [3-16 : 28/28] [3-17 : 28/28] [3-18 : 28/28]
[4-1 : 28/28] [4-2 : 28/28] [4-3 : 28/28] [4-4 : 28/28] [4-5 : 28/28] [4-6 : 28/28] [4-7 : 28/28] [4-8 : 28/28] [4-9 : 28/28]
[4-10 : 00/28] [4-11 : 28/28] [4-12 : 28/28] [4-13 : 28/28] [4-14 : 28/28] [4-15 : 28/28] [4-16 : 28/28] [4-17 : 28/28] [4-18 : 28/28]
[5-1 : 28/28] [5-2 : 28/28] [5-3 : 28/28] [5-4 : 28/28] [5-5 : 28/28] [5-6 : 28/28] [5-7 : 28/28] [5-8 : 28/28] [5-9 : 28/28]
[5-10 : 28/28] [5-11 : 28/28] [5-12 : 28/28] [5-13 : 28/28] [5-14 : 28/28] [5-15 : 28/28] [5-16 : 28/28] [5-17 : 28/28] [5-18 : 28/28]
[6-1 : 28/28] [6-2 : 28/28] [6-3 : 28/28] [6-4 : 28/28] [6-5 : 28/28] [6-6 : 28/28] [6-7 : 28/28] [6-8 : 28/28] [6-9 : 28/28]
[6-10 : 28/28] [6-11 : 28/28] [6-12 : 28/28] [6-13 : 28/28] [6-14 : 28/28] [6-15 : 28/28] [6-16 : 28/28] [6-17 : 28/28] [6-18 : 28/28]
[7-1 : 28/28] [7-2 : 28/28] [7-3 : 28/28] [7-4 : 28/28] [7-5 : 28/28] [7-6 : 28/28] [7-7 : 28/28] [7-8 : 28/28] [7-9 : 28/28]
[7-10 : 28/28] [7-11 : 28/28] [7-12 : 28/28] [7-13 : 28/28] [7-14 : 28/28] [7-15 : 28/28] [7-16 : 28/28] [7-17 : 28/28] [7-18 : 28/28]
[8-1 : 00/28] [8-2 : 28/28] [8-3 : 28/28] [8-4 : 28/28] [8-5 : 28/28] [8-6 : 00/28] [8-7 : 00/28] [8-8 : 00/28] [8-9 : 28/28]
[8-10 : 28/28] [8-11 : 28/28] [8-12 : 28/28] [8-13 : 00/28] [8-14 : 00/28] [8-15 : 00/28] [8-16 : 28/28] [8-17 : 00/28] [8-18 : 00/28]
[9-1 : 28/28] [9-2 : 28/28] [9-3 : 28/28] [9-4 : 28/28] [9-5 : 26/28] [9-6 : 28/28] [9-7 : 28/28] [9-8 : 00/28] [9-9 : 28/28]
[9-10 : 28/28] [9-11 : 28/28] [9-12 : 28/28] [9-13 : 28/28] [9-14 : 28/28] [9-15 : 28/28] [9-16 : 28/28] [9-17 : 24/28] [9-18 : 28/28]
[10-1 : 28/28] [10-2 : 28/28] [10-3 : 28/28] [10-4 : 28/28] [10-5 : 28/28] [10-6 : 28/28] [10-7 : 28/28] [10-8 : 28/28] [10-9 : 28/28]
[10-10 : 28/28] [10-11 : 28/28] [10-12 : 28/28] [10-13 : 28/28] [10-14 : 28/28] [10-15 : 28/28] [10-16 : 28/28] [10-17 : 28/28] [10-18 : 28/28]
[11-1 : 00/28] [11-2 : 00/28] [11-3 : 00/28] [11-4 : 00/28] [11-5 : 00/28] [11-6 : 00/28] [11-7 : 00/28] [11-8 : 00/28] [11-9 : 28/28]
[11-10 : 00/28] [11-11 : 00/28] [11-12 : 00/28] [11-13 : 00/28] [11-14 : 22/28] [11-15 : 00/28] [11-16 : 00/28] [11-17 : 00/28] [11-18 : 00/28]
[12-1 : 00/28] [12-2 : 00/28] [12-3 : 28/28] [12-4 : 28/28] [12-5 : 00/28] [12-6 : 00/28] [12-7 : 00/28] [12-8 : 00/28] [12-9 : 00/28]
[12-10 : 28/28] [12-11 : 28/28] [12-12 : 28/28] [12-13 : 28/28] [12-14 : 00/28] [12-15 : 28/28] [12-16 : 28/28] [12-17 : 00/28] [12-18 : 28/28]
[13-1 : 28/28] [13-2 : 28/28] [13-3 : 28/28] [13-4 : 28/28] [13-5 : 28/28] [13-6 : 28/28] [13-7 : 28/28] [13-8 : 28/28] [13-9 : 28/28]
[13-10 : 28/28] [13-11 : 28/28] [13-12 : 28/28] [13-13 : 28/28] [13-14 : 00/28] [13-15 : 00/28] [13-16 : 00/28] [13-17 : 28/28] [13-18 : 28/28]
[14-1 : 00/28] [14-2 : 28/28] [14-3 : 28/28] [14-4 : 28/28] [14-5 : 28/28] [14-6 : 28/28] [14-7 : 28/28] [14-8 : 00/28] [14-9 : 28/28]
[14-10 : 28/28] [14-11 : 28/28] [14-12 : 28/28] [14-13 : 28/28] [14-14 : 28/28] [14-15 : 28/28] [14-16 : 28/28] [14-17 : 28/28] [14-18 : 28/28]
[15-5 : 24/28] [15-6 : 24/28] [15-7 : 24/28] [15-8 : 24/28] [15-9 : 24/28] [15-10 : 24/28] [15-11 : 24/28] [15-12 : 22/28] [15-13 : 28/28]
[15-14 : 28/28] [15-15 : 28/28] [15-16 : 28/28] [15-17 : 24/28] [15-18 : 21/28] [16-1 : 28/28] [16-2 : 28/28] [16-3 : 28/28] [16-4 : 28/28]
[16-5 : 28/28] [16-6 : 28/28] [16-7 : 28/28] [16-8 : 28/28] [16-9 : 24/28] [16-10 : 28/28] [16-11 : 28/28] [16-12 : 28/28] [16-13 : 28/28]
[16-14 : 28/28] [21-1 : 12/12] [21-2 : 12/12] [21-3 : 12/12] [21-4 : 12/12] [21-5 : 12/12] [21-6 : 12/12] [21-7 : 12/12] [21-8 : 12/12]
[21-9 : 11/12] [21-10 : 12/12] [21-11 : 12/12] [21-12 : 12/12] [21-13 : 12/12] [21-14 : 12/12] [21-15 : 12/12] [21-16 : 12/12] [22-1 : 12/12]
[22-2 : 12/12] [22-3 : 12/12] [22-4 : 12/12] [22-5 : 12/12] [22-6 : 12/12] [22-7 : 12/12] [22-8 : 12/12] [22-9 : 00/32] [22-10 : 00/72]
[23-3 : 24/24] [23-4 : 22/24] [ : 00/6] [ : 00/6] [ : 00/6] [ : 00/6] [ : 00/6] [ : 00/6] [ : 00/6] [ : 00/6]
[ : 00/6]
Total Cpus: 8280
Alloc Cpus: 6851
CPU Utilization: 82.7%
    
```