Job scheduler and job submission

Bootcamp on using Paramshakti, IIT Kharagpur, September 18, 2021

Dr. Sandeep Kumar Reddy, CCDS, IIT Kharagpur

Outline

- Job scheduler - Slurm
- Useful Slurm commands
- Serial and parallel computing
- Submitting jobs
- Monitoring jobs
- Slurm account coordinator

Participants

- 67% participants never used HPC facility before
- 23% participants not aware of basic Linux commands
- Part-II of Bootcamp: training for beginners

Job scheduler - SLURM

- Open-source resource management and job scheduling software for Linux computing clusters
- Resource management: manage and represent resources like CPU-cores, memory and GPU card to the users in a simplest way
- Job scheduling: decides which user jobs to run for optimal utilization of the cluster
- Website: https://slurm.schedmd.com/
- It can be used on desktop PC, workstations, servers and clusters as well

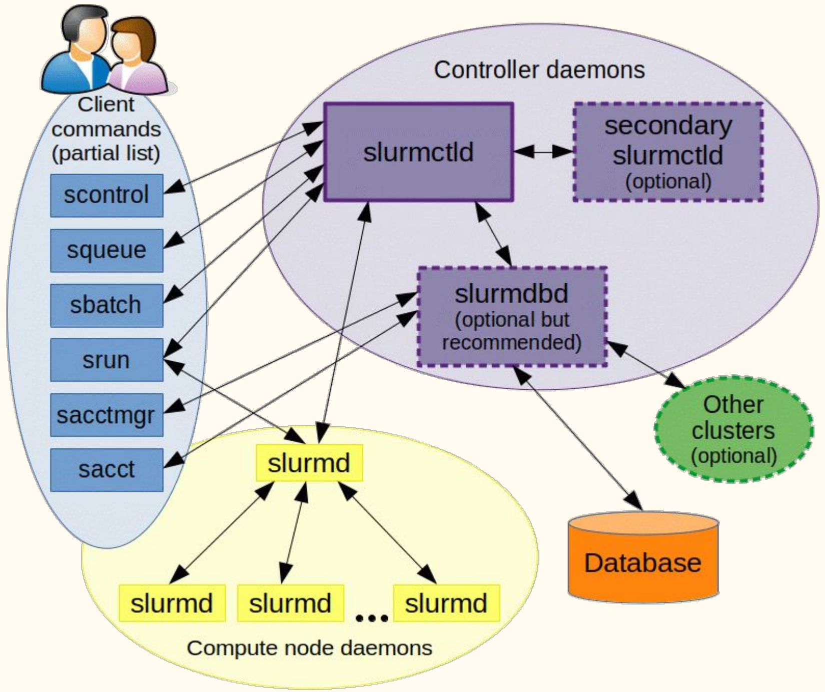
Do you need to learn any programming language to use SLURM?

NO

To submit your jobs, learning a few commands and associated flags is sufficient

Slurm components

https://slurm.schedmd.com/quickstart.html



Client commands (partial list)
scontrol
squeue
sbatch
srun
sacctmgr
sacct
Controller daemons
slurmctld
secondary slurmctld (optional)
slurmdbd (optional but recommended)
Other clusters (optional)
Database
slurmd
slurmd
slurmd
...
slurmd
Compute node daemons

Useful Slurm commands

https://slurm.schedmd.com/pdfs/summary.pdf

Command	**Used for**
sinfo	view information about Slurm nodes and partitions/queues.
sbatch	Submit a job for remote execution
squeue	Display the list of all jobs
scancel	Kill pending or running jobs
scontrol	Display or modify Slurm configuration and state eg. job, node, partition, reservation, and overall system configuration
seff	Display job efficiency information of completed jobs

Name of compute nodes on PS

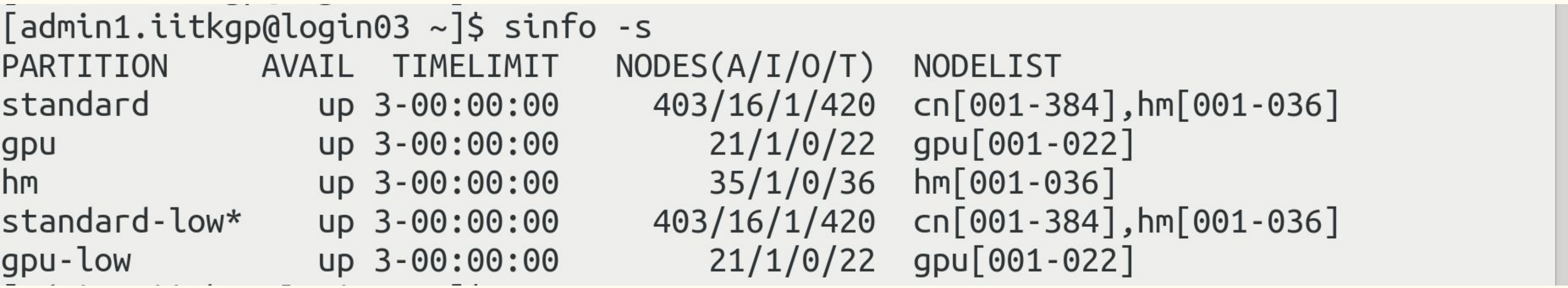
- CPU only nodes: (memory - 4.8 GB per core)
  - cn001, cn002, cn003, cn004, ... , cn384
- Nodes with high memory: (memory - 19.2 GB per core)
  - hm001, hm002, hm003, ..., hm036
- Nodes with GPUs:
  - gpu001, gpu002, gpu003, ..., gpu022

Storage

- For keeping the software, codes, scripts, etc use /home
  - By default, you are on */home/$USER* path after login
- For running jobs, always copy files to /scratch
  - After login, type *cd /scratch/$USER* to go the scratch directory

sinfo

- view information about Slurm nodes and partitions/queues.
- *sinfo --help* to display all possible options



```
[admin1.iitkgp@login03 ~]$ sinfo -s
PARTITION      AVAIL  TIMELIMIT   NODES(A/I/O/T)  NODELIST
standard          up 3-00:00:00     403/16/1/420  cn[001-384],hm[001-036]
gpu               up 3-00:00:00        21/1/0/22  gpu[001-022]
hm                up 3-00:00:00        35/1/0/36  hm[001-036]
standard-low*     up 3-00:00:00     403/16/1/420  cn[001-384],hm[001-036]
gpu-low           up 3-00:00:00        21/1/0/22  gpu[001-022]
```

partitions/queue

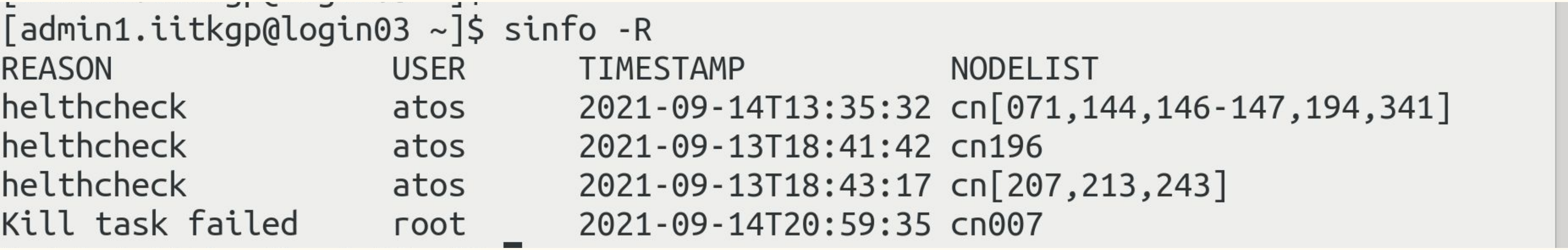
Following partitions/queues have been defined for different requirements.

standard:	CPU and High memory jobs (*Chargeable)
gpu:	CPU and GPU jobs (*Chargeable)
hm:	CPU and High memory intensive jobs (*Chargeable)
standard-low:	Default Non-chargeable Queue for CPU and High memory jobs*
gpu-low:	Non-chargeable CPU and GPU jobs *

(* Less priority)

sinfo

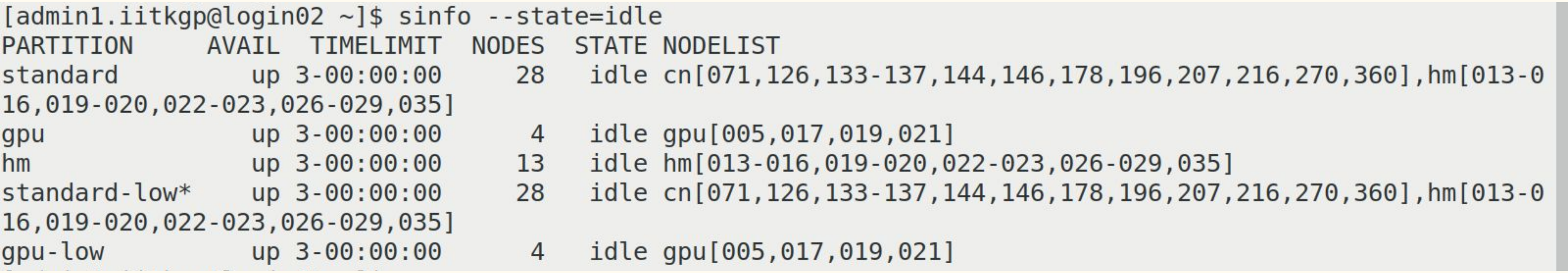
- view information about Slurm nodes and partitions/queues.
- *sinfo --help* to display all possible options



```
[admin1.iitkgp@login03 ~]$ sinfo -R
REASON               USER      TIMESTAMP           NODELIST
helthcheck           atos      2021-09-14T13:35:32 cn[071,144,146-147,194,341]
helthcheck           atos      2021-09-13T18:41:42 cn196
helthcheck           atos      2021-09-13T18:43:17 cn[207,213,243]
Kill task failed     root      2021-09-14T20:59:35 cn007
```

sinfo

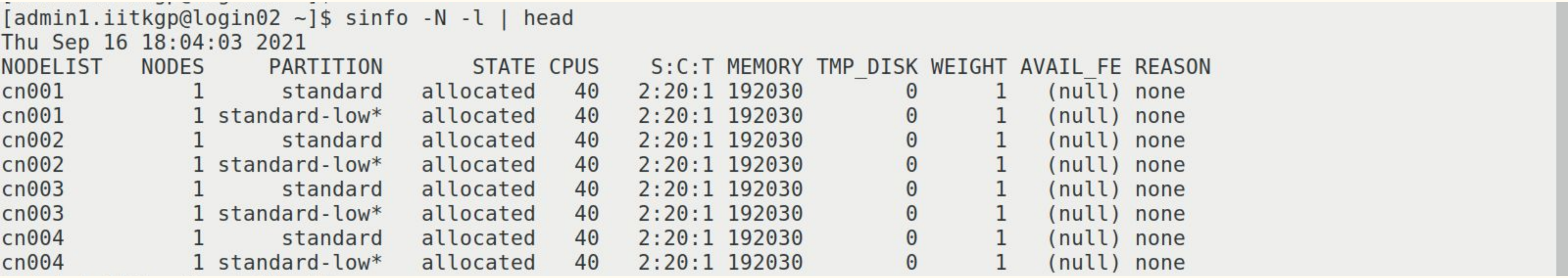
- view information about Slurm nodes and partitions/queues.
- *sinfo --help* to display all possible options



```
[admin1.iitkgp@login02 ~]$ sinfo --state=idle
PARTITION     AVAIL  TIMELIMIT  NODES  STATE NODELIST
standard         up 3-00:00:00     28   idle cn[071,126,133-137,144,146,178,196,207,216,270,360],hm[013-0
16,019-020,022-023,026-029,035]
gpu              up 3-00:00:00      4   idle gpu[005,017,019,021]
hm               up 3-00:00:00     13   idle hm[013-016,019-020,022-023,026-029,035]
standard-low*    up 3-00:00:00     28   idle cn[071,126,133-137,144,146,178,196,207,216,270,360],hm[013-0
16,019-020,022-023,026-029,035]
gpu-low          up 3-00:00:00      4   idle gpu[005,017,019,021]
```

sinfo

- view information about Slurm nodes and partitions/queues.
- *sinfo --help* to display all possible options



```
[admin1.iitkgp@login02 ~]$ sinfo -N -l | head
Thu Sep 16 18:04:03 2021
NODELIST   NODES     PARTITION       STATE CPUS    S:C:T MEMORY TMP_DISK WEIGHT AVAIL_FE REASON
cn001          1      standard   allocated   40   2:20:1 192030        0      1   (null) none
cn001          1 standard-low*   allocated   40   2:20:1 192030        0      1   (null) none
cn002          1      standard   allocated   40   2:20:1 192030        0      1   (null) none
cn002          1 standard-low*   allocated   40   2:20:1 192030        0      1   (null) none
cn003          1      standard   allocated   40   2:20:1 192030        0      1   (null) none
cn003          1 standard-low*   allocated   40   2:20:1 192030        0      1   (null) none
cn004          1      standard   allocated   40   2:20:1 192030        0      1   (null) none
cn004          1 standard-low*   allocated   40   2:20:1 192030        0      1   (null) none
```

Node state codes

https://slurm.schedmd.com/sinfo.html

- ALLOCATED
  - The node has been allocated to one or more jobs
- IDLE
  - The node is not allocated to any jobs and is available for use
- DOWN
  - Node is not available for use
- RESERVED
  - The node is in an advanced reservation and not generally available
- MIXED
  - The node has some of its CPUs ALLOCATED while others are IDLE
- DRAINED
  - The node is unavailable for use per system administrator request

Serial and parallel jobs

- Parallel computation algorithms fall into three categories:
  - Message-passing interface models (MPI) -- processes
  - Shared memory models (OpenMP) -- threads
  - Hybrid (OpenMP + MPI)
  - GPU general computing models
- A process means executing a program
- One or more threads run in the context of the process
- Processes and threads are independent sequences of execution; work on any part of the code
- np process x nt threads = np*nt processors required

Serial and parallel jobs

- A thread cannot exist by itself, a process must start a thread. A process can start multiple threads in other words “threads are not independent like processes”
- Use of a process means you also need Inter Process Communication (IPC) to get data in and out of the process
- Threads on the same process are much more lightweight and reside in the same memory space
- Switching among threads in the same process is much faster because the OS only switches registers, not memory mapping

Serial and parallel jobs

- If you have a serial code, and submitted on many processors, will it run faster?
  - NO
- Can you run serial code on more than one processor without any additional programming or using tools?
  - NO
- If you have a software, how do you know whether you can run on it on more than one processor?
  - Check software documentation or open the code in editor and check for words starting with MPI, OMP, CUDA, KERNEL, etc
- Is it easy to convert serial code to parallel?
  - YES AND NO

Serial and parallel jobs

- GPUs consist hundreds of cores vs CPUs a few cores
- For GPUs, improved memory bandwidth, around a factor of 5x compared to CPUs-systems (350 GB/s vs 70 GB/s)
- For programming GPUs: OpenCL, CUDA, and Halide languages
- Speedup with GPUs against a well-optimized CPU code would be in the 2x to 10x range, with an average of around 5x

https://forums.developer.nvidia.com/

Serial and parallel jobs

- Can you run a CPU parallel code on GPU cards?
  - NO
- If you use four GPU cards instead of two for a GPU code, will it run faster ?
  - NO. Advisable to use one or two GPU cards

Submitting jobs - sbatch, salloc, srun

- *sbatch* and *salloc* allocate resources to the job
- *sbatch* script.sh
- *srun* launches parallel tasks across allocated resources
- *srun* can also be used outside the resource allocation

salloc

Allocate resources for interactive bash session or for executing a script (which originates from the login node)

Useful for debugging the codes/profiling

- *salloc --time=00:10:00 -n=4*
  - It will reserve 4 cores for 10 min. Once resources are allocated, you can log into the nodes to run the command
  - Time is in HH:MM:SS format (other available formats are MM, MM:SS, HH:MM:SS, DD-HH, DD-HH:MM, DD-HH:MM:SS)
- *slloc --time=0:10:00 -n=4 mpirun -np 4 ./executable*

Workflow for job submission

- Create job script
- Submit the job script with *sbatch* command
- If job submission is successful, you will see a jobID printed on the commandline.

  Else, check the script and submit again
- Check job status with *squeue* command
- Use *sacct*, *seff* commands to check job information, if needed

Rules for writing the Slurm script

- Contain the options preceded with *#SBATCH*
- Executable commands come after all *#SBATCH* directives
- Any *#SBATCH* directive after executable command will not be processed
- By default, the script/Slurm will check for files in current working directory
- Load the corresponding modules before using software executable
- Spaces are allowed
- For adding comments, use #

  (*#SBATCH* directive --> *##SBATCH*)

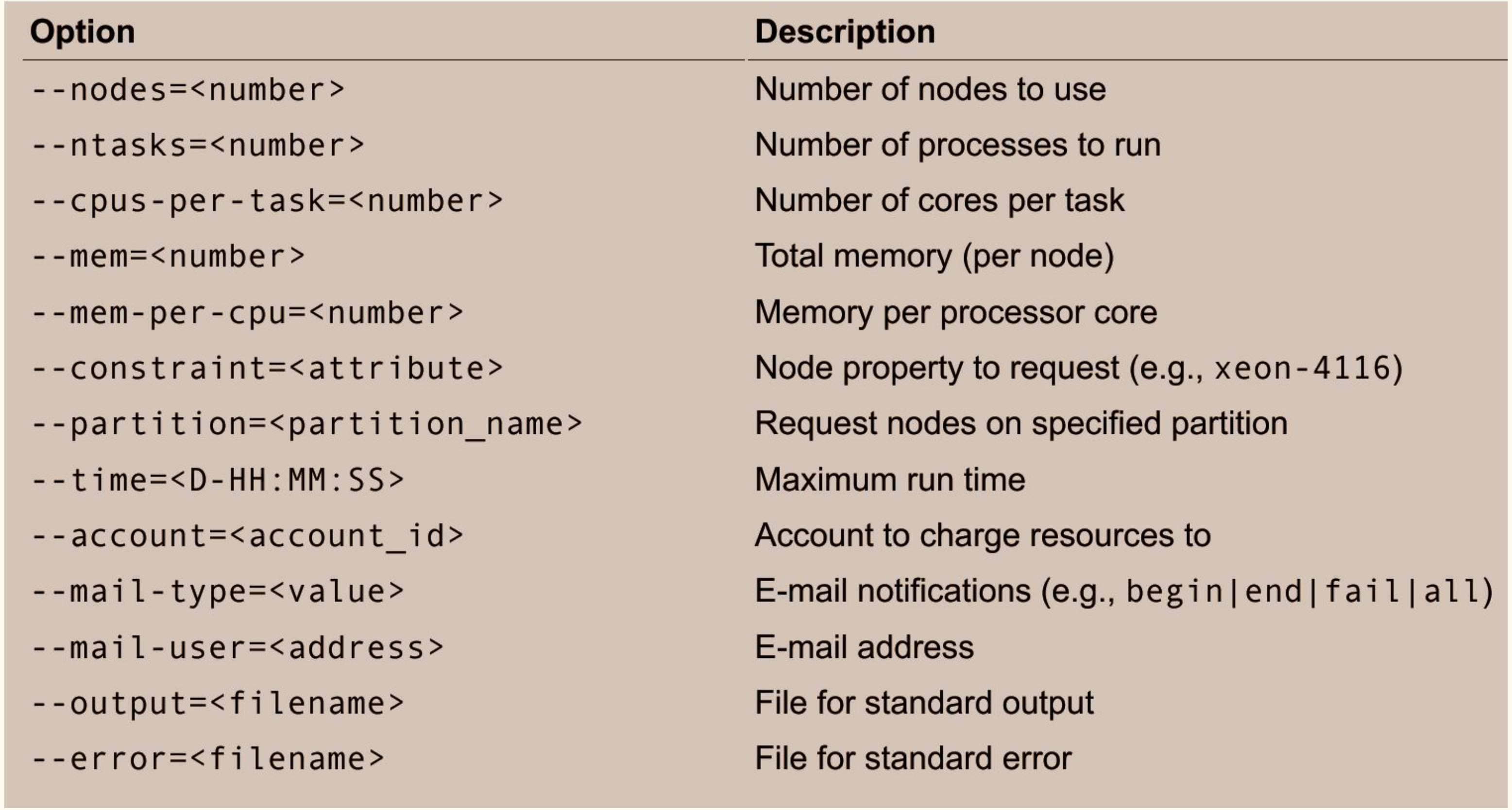
sbatch

Slurm script files are at: (on PS)

/home/iitkgp/slurm-scripts/

sbatch

https://slurm.schedmd.com/sbatch.html



Option	Description
`--nodes=<number>`	Number of nodes to use
`--ntasks=<number>`	Number of processes to run
`--cpus-per-task=<number>`	Number of cores per task
`--mem=<number>`	Total memory (per node)
`--mem-per-cpu=<number>`	Memory per processor core
`--constraint=<attribute>`	Node property to request (e.g., `xeon-4116`)
`--partition=<partition_name>`	Request nodes on specified partition
`--time=<D-HH:MM:SS>`	Maximum run time
`--account=<account_id>`	Account to charge resources to
`--mail-type=<value>`	E-mail notifications (e.g., `begin|end|fail|all`)
`--mail-user=<address>`	E-mail address
`--output=<filename>`	File for standard output
`--error=<filename>`	File for standard error

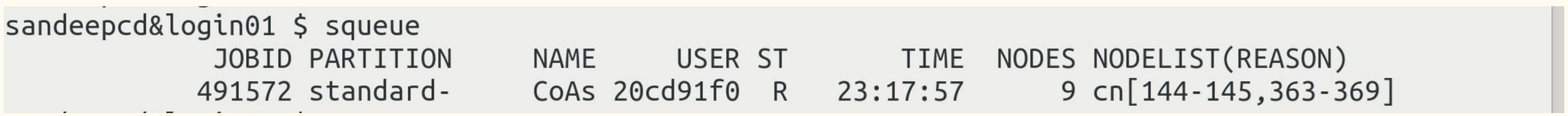
Environment variables

SLURM_JOB_ID	ID of the job allocation
SLURM_SUBMIT_DIR	directory from which sbatch was invoked
SLURM_GPUS	Number of GPUs requested
SLURM_JOB_NODELIST	List of nodes allocated to the job
SLURM_NTASKS	Number of tasks requested
SLURM_CPUS_PER_TASK	Number of cpus requested per task
SLURM_ARRAY_TASK_ID	Job array ID (index) number

https://slurm.schedmd.com/sbatch.html

squeue

- Display the submitted jobs in your account
- *squeue --help*
- https://slurm.schedmd.com/squeue.html



```
sandeepcd&login01 $ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            491572 standard-     CoAs 20cd91f0  R   23:17:57      9 cn[144-145,363-369]
```

Common codes for job states

https://slurm.schedmd.com/squeue.html#lbAG

Status	Code	Explanation
COMPLETED	CD	The job has completed successfully.
COMPLETING	CG	The job is finishing but some processes are still active.
FAILED	F	The job terminated with a non-zero exit code and failed to execute.
PENDING	PD	The job is waiting for resource allocation. It will eventually run.
PREEMPTED	PR	The job was terminated because of preemption by another job.
RUNNING	R	The job currently is allocated to a node and is running.
SUSPENDED	S	A running job has been stopped with its cores released to other jobs.
STOPPED	ST	A running job has been stopped with its cores retained.

Common codes for pending reason

https://slurm.schedmd.com/squeue.html#lbAF

Reason Code	**Explanation**
Priority	One or more higher priority jobs is in queue for running. Your job will eventually run
Dependency	This job is waiting for a dependent job to complete and will run afterwards
Resources	The job is waiting for resources to become available and will eventually run
InvalidAccount	The job's account is invalid. Cancel the job and rerun with correct account
QOSGrpMaxJobsLimit	Maximum number of jobs for your job's QoS have been met; job will run eventually
ReqNodeNotAvail	Some node specifically required by the job is not currently available; job will run eventually

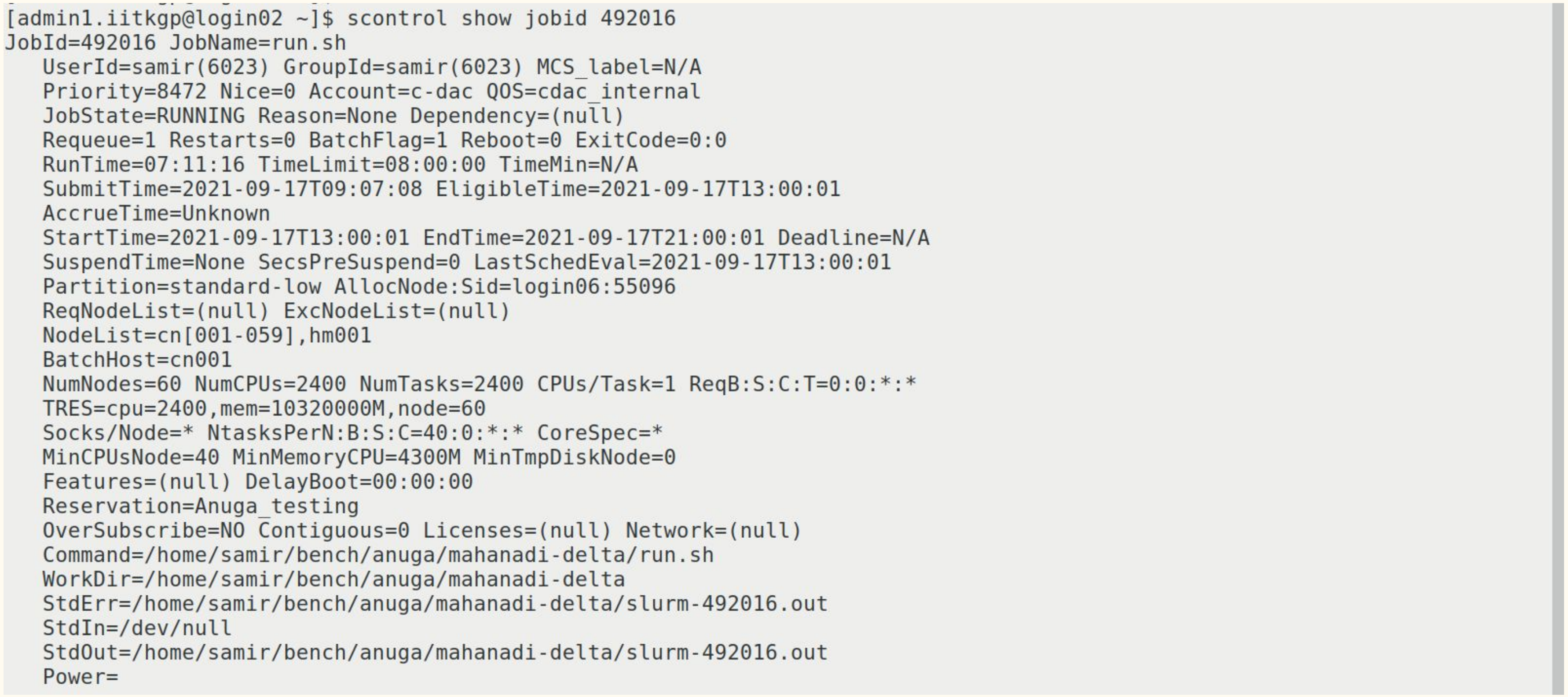
scancel

- *scancel --help*
- https://slurm.schedmd.com/scancel.html
- *scancel JOBID*
    - *scancel 492806*

scontrol

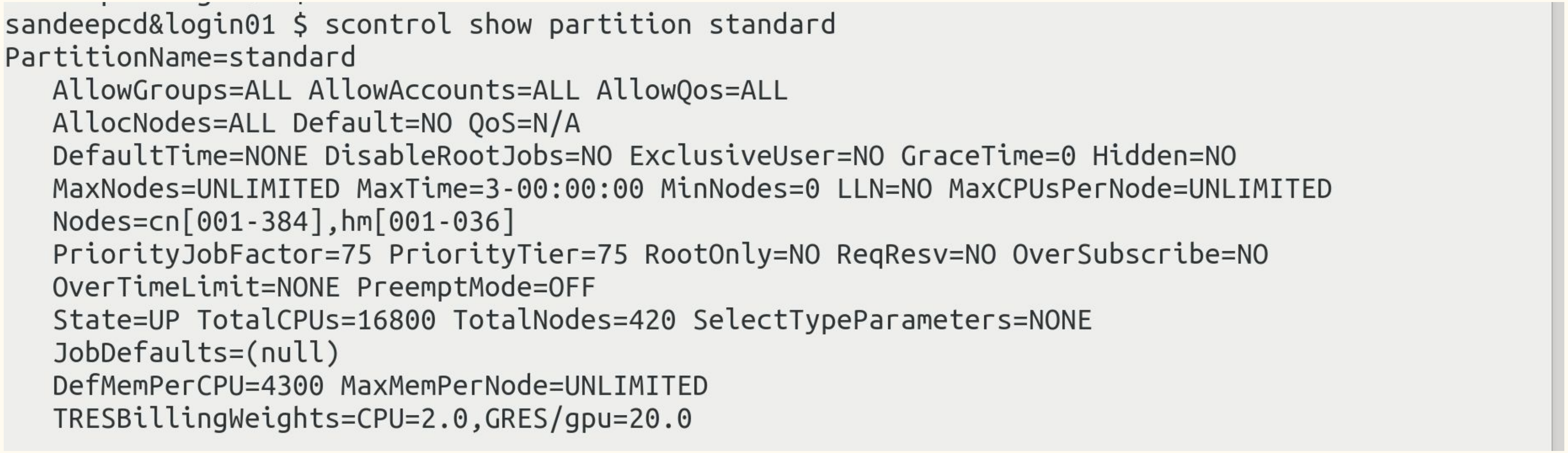
- scontrol show partition <partition>
  - *scontrol show partition standard-low*
- scontrol show node <nodeid>
  - *scontrol show node cn200*
- scontrol show job <jobid>
  - *scontrol show job 238242*

scontrol



```
[admin1.iitkgp@login02 ~]$ scontrol show jobid 492016
JobId=492016 JobName=run.sh
   UserId=samir(6023) GroupId=samir(6023) MCS_label=N/A
   Priority=8472 Nice=0 Account=c-dac QOS=cdac_internal
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=07:11:16 TimeLimit=08:00:00 TimeMin=N/A
   SubmitTime=2021-09-17T09:07:08 EligibleTime=2021-09-17T13:00:01
   AccrueTime=Unknown
   StartTime=2021-09-17T13:00:01 EndTime=2021-09-17T21:00:01 Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2021-09-17T13:00:01
   Partition=standard-low AllocNode:Sid=login06:55096
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=cn[001-059],hm001
   BatchHost=cn001
   NumNodes=60 NumCPUs=2400 NumTasks=2400 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=2400,mem=10320000M,node=60
   Socks/Node=* NtasksPerN:B:S:C=40:0:*:* CoreSpec=*
   MinCPUsNode=40 MinMemoryCPU=4300M MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   Reservation=Anuga_testing
   OverSubscribe=NO Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/samir/bench/anuga/mahanadi-delta/run.sh
   WorkDir=/home/samir/bench/anuga/mahanadi-delta
   StdErr=/home/samir/bench/anuga/mahanadi-delta/slurm-492016.out
   StdIn=/dev/null
   StdOut=/home/samir/bench/anuga/mahanadi-delta/slurm-492016.out
   Power=
```

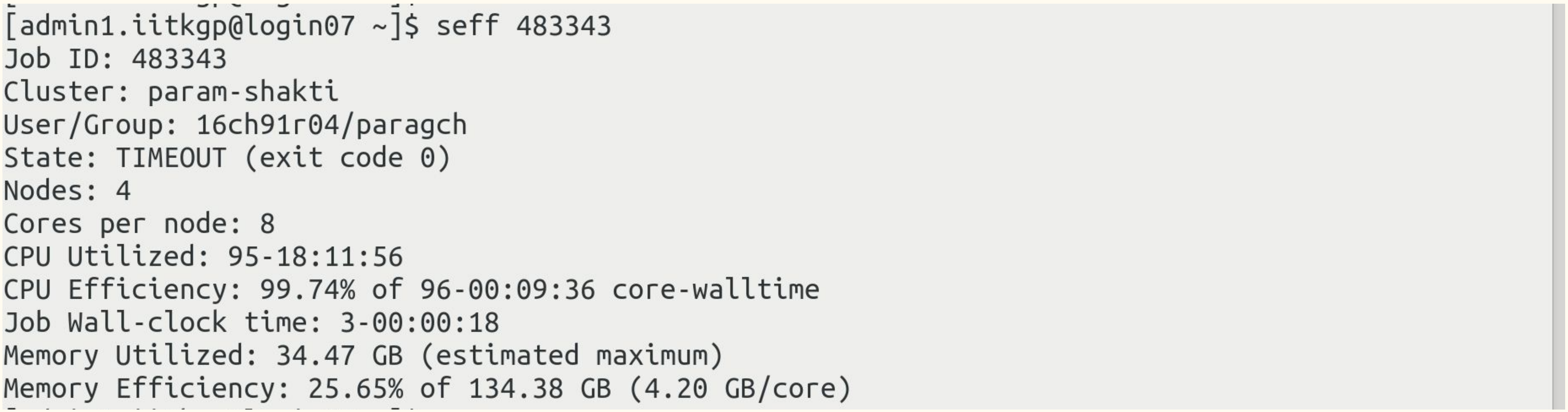
scontrol



```
sandeepcd&login01 $ scontrol show partition standard
PartitionName=standard
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=NO QoS=N/A
   DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=3-00:00:00 MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=cn[001-384],hm[001-036]
   PriorityJobFactor=75 PriorityTier=75 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=16800 TotalNodes=420 SelectTypeParameters=NONE
   JobDefaults=(null)
   DefMemPerCPU=4300 MaxMemPerNode=UNLIMITED
   TRESBillingWeights=CPU=2.0,GRES/gpu=20.0
```

seff

- Reports how much % of memory and CPUs are used
- *seff <jobid>*



```
[admin1.iitkgp@login07 ~]$ seff 483343
Job ID: 483343
Cluster: param-shakti
User/Group: 16ch91r04/paragch
State: TIMEOUT (exit code 0)
Nodes: 4
Cores per node: 8
CPU Utilized: 95-18:11:56
CPU Efficiency: 99.74% of 96-00:09:36 core-walltime
Job Wall-clock time: 3-00:00:18
Memory Utilized: 34.47 GB (estimated maximum)
Memory Efficiency: 25.65% of 134.38 GB (4.20 GB/core)
```

Job priority

Five important factors that decide job priority

- Age
  - the length of time a job has been waiting in the queue, eligible to be scheduled
- Fair-share
  - the difference between the portion of the computing resource that has been promised and the amount of resources that has been consumed
- Job size
  - the number of nodes or CPUs a job is allocated
- Partition
  - a factor associated with each node partition
- QOS
  - a factor associated with each Quality Of Service

Slurm coordinator

Faculty adviser can manage resources within his research group on PS.

Examples:

- *sacctmgr modify user student1 set MaxTRES=billing=1000000*
  - Limit on the billing consumption Rs.10000 of a user in a chargeable partition
- *sacctmgr modify user student2   set GrpCPUs=64*
  - Limit on the number of processor cores for a user
- *sacctmgr modify user student3 set Fairshare=10*
  - Default value = 1. This number is relative to other users in that faculty account.

http://www.hpc.iitkgp.ac.in/HPCF/slurmcoord

Is it correct?

Walltime of a job as a function of no. of processes

Num processors	walltime (sec)
8	1000
16	500
32	250
64	125
128	63
256	32
512	16

Is it correct?

Walltime of a job as a function of no. of processes

Num processors	walltime (hrs)
8	1000
16	500
32	250
64	125
128	63
256	32
512	16

- No, it is almost impossible to get a perfect scaling and it depends on many factors
- One should always have a idea about the scaling behaviour of the code
- Otherwise, do small tests and notice the walltime (Slurm backfill scheduling)
- Keep time to as low as possible, eg. -t 00:02:00 while doing scaling tests

Best practises

- Check parallel performance of the code or have an idea of the performance
- Do small tests before submitting when needed
- Do not always go with maximum walltime in the Slurm script.
- Any software can be installed in your home directory. No permission or intimation is required

Additional resources

- Official Slurm documentation

  https://slurm.schedmd.com/documentation.html

- Our website http://www.hpc.iitkgp.ac.in/

- https://paramshakti.iitkgp.ac.in/support/ for any query/support

- Bootcamp material is at:

  http://www.hpc.iitkgp.ac.in/HPCF/bootcampIITKgp