Introduction on Slurm Job Submission

dicos-support@twgrid.org
Academia Sinica Grid-computing Centre (ASGC)

2026/05/27 Yi-Ru CHEN

Outline

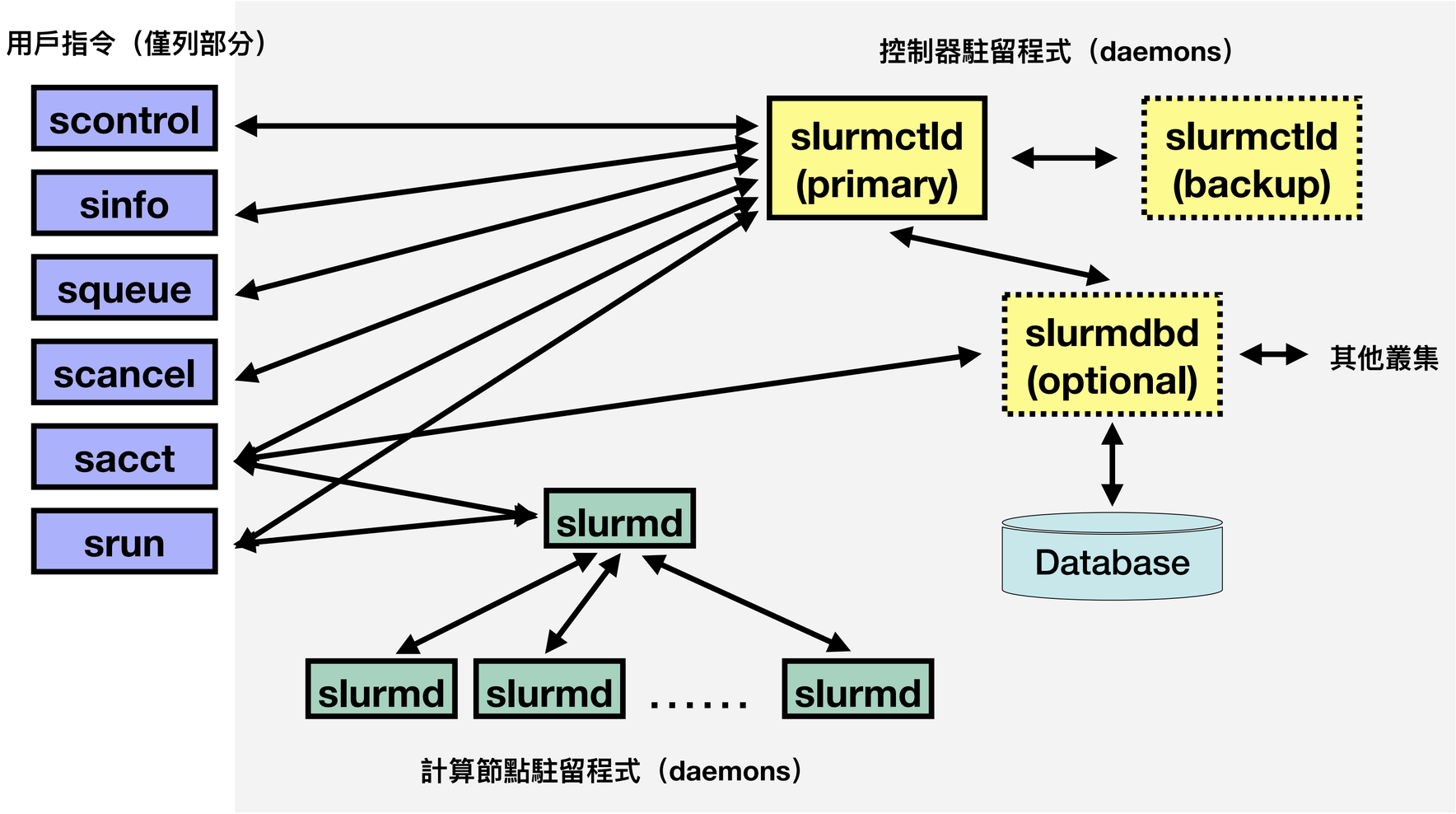
- Introduction of Slurm
- User Interfaces (Login Node)
- Basic Usage of Slurm System
- Environment Modules (Lmod)
- Python, Compilation and MPI Enviroment
- Available Software
- Job Submission Example

Introduction of Slurm

Introduction of slurm

Overview

- Slurm is an
  - Open source
  - Fault-tolerant
  - Highly scalable
- Cluster management
  - Job scheduling
- system



Introduction of slurm

ASGC resources

- Computing Machine Specifications Computing Nodes :

CPU Cluster	CPU Model	Nodes	RAM-Per-Node	Cores-Per-Node	Total Cores
EDR1	AMD Genoa 9654@2.4GHz	20	1.5 TB	192	3840
EDR2	AMD EPYC 9645 96-Core Processor	10	1.5 TB	192	1920
Intel-g4	Intel(R) Xeon(R) Gold 6448H	4	1.0 TB	128	512

GPU Cluster	GPU Model	Nodes	GPU-Boards-Per-Node	CPU Model	CPU Cores-Per-Node
GPU-A100	NVIDIA A100	2	8	Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz	64
GPU-V100	NVIDIA V100	6	8	AMD EPYC 7302 16-Core Processor	48
GPU-B6000	NVIDIA RTX pro 6000	2	8	Intel(R) Xeon(R) 6517P	64

https://dicos.grid.sinica.edu.tw/resources

User Interfaces (Login Nodes)

User Interfaces (Login Nodes)

Login into Slurm user interface

- The user interface node for slurm are :

  slurm-ui.twgrid.org

- Login in user interface :

  ssh <your account>@slurm-ui.twgrid.org

  Enter your password and 2FA verification code

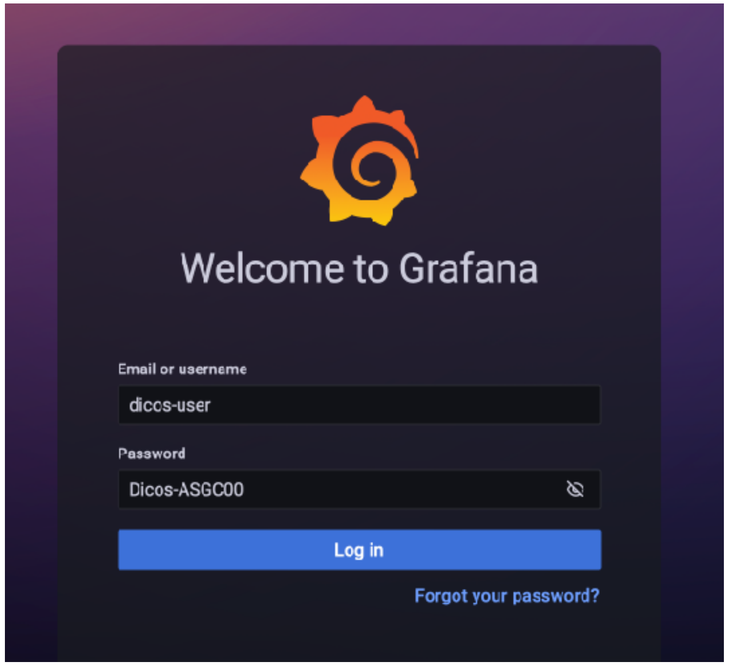
- For Windows users can download and install SSH client software (e.g. PuTTY, MobaXterm, VScode, etc.).

- For macOS users, you can open the built-in terminal directly.

- You will be prompted with the relative information of your account when login into the slurm user interfaces.

User Interfaces (Login Nodes)

Login into Slurm user interface



https://grafana01.twgrid.org:3000/d/3MezxIdDz/slurm-dashboard?

User Interfaces (Login Nodes)

Login into Slurm user interface

```
Notice:

1. Don't remove or edit /dicos_ui_home/<your account>/.google_authenticator (file under
   your home directory).

2. /dicos_ui_home/ is a temporary storage space.
   You are fully responsible for the security and integrity of your data in this directory
   (ASGC does not guarantee the integrity of the data in this directory).
   Please back up your data to the Ceph directory (/ceph/work/<your group>/) or transfer
   it to your own storage.

3. If you use /tmp on worker node as an intermediate working space, please remember to
   include a command in your submission script to remove files from /tmp when your tasks
   finish. You can include one of the following commands in your submission script:
   $ rm -rf /tmp/<your file name>
   or
   $ ls -al /tmp/ | grep <your account> | awk '{print $9}' | xargs -I {} rm -rf /tmp/{}

4. The /dicos_ui_home/ space migration is scheduled for Jan. 24, 2025.
   The original /dicos_ui_home/ space has been renamed to /dicos_ui_home_old_20250124.
   Please copy your data from the old space (/dicos_ui_home_old_20250124) to the new empty
   /dicos_ui_home/ space by Mar. 15, 2025.

5. Please avoid running computational jobs on login nodes.
   Use the job scheduler to submit your tasks.
   Jobs running on login nodes will be killed without notice.

6. New Configuration Update:
   - Set the maximum memory available per allocated CPU.
   If the number of CPUs specified by the user is insufficient to meet the memory
   requirements of the job, the Slurm system will automatically allocate enough CPU cores
   to satisfy the actual memory demand.
```

Cluster	Maximum memory per core
EDR1	7165
EDR2	7967
Intel-g4	7162
V100	15828
A100	15933
B6000	23277

User Interfaces (Login Nodes)

Mount Space

- User space (100 GB) : /dicos_ui_home/{your account}
- Group Space (3 TB) : /ceph/work/{your group}
  - For working and installing software

```
Last login: Thu Jan  2 05:23:36 2025 from 140.109.223.149
[yiruchen0101@slurm-ui01 ~]$ pwd
/dicos_ui_home/yiruchen0101
[yiruchen0101@slurm-ui01 ~]$ cd /ceph/work/ASGC/
[yiruchen0101@slurm-ui01 ASGC]$ pwd
/ceph/work/ASGC
[yiruchen0101@slurm-ui01 ASGC]$ ls
Workshop  alice525  felixlee  jyou  mandy4318  storage_handson  thwu  yiruchen0101
```

Basic Usage of Slurm System

Basic Usage of Slurm System

Partition/ Queues of slurm

- Slurm Partitions (Queues)
- The default queue is “edr1-al9_short”.
- Users could submit to different partitions by assigning partition parameters, e.g.

  sbatch -p edr1-al9_large my_script.sh

Cluster	Partition	Timelimit	Nodelist
HDR1	bigmemory_hdr1	14-00:00:00	hpa-wn[01,03]
EDR1/EDR2	global	03:00:00	hpa-wn[05-34]
EDR1	edr1-al9_short	03:00:00	hpa-wn[05-24]
	edr1-al9_short_serial	04:00:00	
	edr1-al9_large	14-00:00:00	
	edr1-al9_long_serial	14-00:00:00	
	edr1-al9_moderate_serial	2-00:00:0	
EDR2	edr2-al9_short	03:00:00	hpa-wn[25-34]
	edr2-al9_short_serial	04:00:00	
	edr2-al9_large	14-00:00:00	
	edr2-al9_long_serial	14-00:00:00	
	edr2-al9_moderate_serial	2-00:00:0	
Intel-g4	intel-g4-al9_short	03:00:00	hpi-wn[01-04]
	intel-g4-al9_short_serial	04:00:00	
	intel-g4-al9_large	14-00:00:00	
	intel-g4-al9_long_serial	14-00:00:00	
	intel-g4-al9_moderate_serial	2-00:00:0	
A100	a100-al9	5-00:00:00	hp-teslaa[01,03]
	a100_long-al9	7-00:00:00	
	a100_short-al9	06:00:00	
	a100_devel-al9	20:00	
V100	v100-al9	5-00:00:0	hp-teslav[01-06]
	v100-al9_short	06:00:00	
	v100-al9_long	7-00:00:00	
B6000	b6000-al9	5-00:00:0	hp-rtx6kp[01-02]
	b6000-al9_short	06:00:00	
	b6000-al9_long	7-00:00:00	

Basic Usage of Slurm System

Resource and QoS Limitations in Slurm

- By default, each job is allocated **6000 MB of memory per CPU**, unless the user requests a different amount.
  #SBATCH --mem==<amount> request memory for the entire job
- Quality of Service (QoS)
  - MaxTRES：maximum resources that can be requested in a QOS.
  - MinTRES：maximum number of CPUs an user can request in a QOS.

	Partition	MaxNodes	MaxTRES	MinTRES
CPU	**XXX_short_serial**	N/A	CPU=24	N/A
	XXX_long_serial	**1**	N/A	N/A
	XXX_moderate_serial	N/A	CPU=24	N/A
	XXX_large	N/A	N/A	N/A
GPU	**XXX**	N/A	gres/gpu=8	gres/gpu=1
	XXX_short	N/A	gres/gpu=8	gres/gpu=1
	XXX_long	N/A	gres/gpu=8	gres/gpu=1

Basic Usage of Slurm System

User commands : sinfo

- Query partitions and nodes information : sinfo or sinfo -N

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sinfo
PARTITION                    AVAIL  TIMELIMIT  NODES  STATE NODELIST
intel-g4-al9_short              up    3:00:00      4  alloc hpi-wn[01-04]
intel-g4-al9_short_serial       up    4:00:00      4  alloc hpi-wn[01-04]
intel-g4-al9_large              up 14-00:00:0      4  alloc hpi-wn[01-04]
intel-g4-al9_long_serial        up 14-00:00:0      4  alloc hpi-wn[01-04]
intel-g4-al9_moderate_serial    up 2-00:00:00      4  alloc hpi-wn[01-04]
edr1-al9_short*                 up    3:00:00     20  alloc hpa-wn[05-24]
edr1-al9_short_serial           up    4:00:00     20  alloc hpa-wn[05-24]
edr1-al9_large                  up 14-00:00:0     20  alloc hpa-wn[05-24]
edr1-al9_long_serial            up 14-00:00:0     20  alloc hpa-wn[05-24]
edr1-al9_moderate_serial        up 2-00:00:00     20  alloc hpa-wn[05-24]
v100-al9                        up 5-00:00:00      2    mix hp-teslav[01,06]
v100-al9                        up 5-00:00:00      4   idle hp-teslav[02-05]
v100-al9_short                  up    6:00:00      2    mix hp-teslav[01,06]
v100-al9_short                  up    6:00:00      4   idle hp-teslav[02-05]
v100-al9_long                   up 7-00:00:00      2    mix hp-teslav[01,06]
v100-al9_long                   up 7-00:00:00      4   idle hp-teslav[02-05]
a100-al9                        up 5-00:00:00      2    mix hp-teslaa[01,03]
a100_long-al9                   up 7-00:00:00      2    mix hp-teslaa[01,03]
a100_short-al9                  up    6:00:00      2    mix hp-teslaa[01,03]
a100_devel-al9                  up      20:00      2    mix hp-teslaa[01,03]
 .
 .
 .
```

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sinfo -N
NODELIST     NODES                    PARTITION STATE
hp-teslaa01      1                a100_long-al9 mix
hp-teslaa01      1               a100_devel-al9 mix
hp-teslaa01      1                     a100-al9 mix
hp-teslaa01      1               a100_short-al9 mix
hp-teslaa03      1                a100_long-al9 mix
hp-teslaa03      1               a100_devel-al9 mix
hp-teslaa03      1                     a100-al9 mix
hp-teslaa03      1               a100_short-al9 mix
hp-teslav01      1               v100-al9_short mix
hp-teslav01      1                     v100-al9 mix
hp-teslav01      1                v100-al9_long mix
hp-teslav02      1               v100-al9_short idle
hp-teslav02      1                     v100-al9 idle
hp-teslav02      1                v100-al9_long idle
hp-teslav03      1               v100-al9_short idle
hp-teslav03      1                     v100-al9 idle
hp-teslav03      1                v100-al9_long idle
hp-teslav04      1               v100-al9_short idle
hp-teslav04      1                     v100-al9 idle
hp-teslav04      1                v100-al9_long idle
hp-teslav05      1               v100-al9_short idle
hp-teslav05      1                     v100-al9 idle
 .
 .
 .
```

Basic Usage of Slurm System

User commands : sinfo

- Query partitions and nodes information : sinfo or sinfo -N

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sinfo
PARTITION                    AVAIL  TIMELIMIT  NODES  STATE NODELIST
intel-g4-al9_short              up    3:00:00      4  alloc hpi-wn[01-04]
intel-g4-al9_short_serial       up    4:00:00      4  alloc hpi-wn[01-04]
intel-g4-al9_large              up 14-00:00:0      4  alloc hpi-wn[01-04]
intel-g4-al9_long_serial        up 14-00:00:0      4  alloc hpi-wn[01-04]
intel-g4-al9_moderate_serial    up 2-00:00:00      4  alloc hpi-wn[01-04]
edr1-al9_short*                 up    3:00:00     20  alloc hpa-wn[05-24]
edr1-al9_short_serial           up    4:00:00     20  alloc hpa-wn[05-24]
edr1-al9_large                  up 14-00:00:0     20  alloc hpa-wn[05-24]
edr1-al9_long_serial            up 14-00:00:0     20  alloc hpa-wn[05-24]
edr1-al9_moderate_serial        up 2-00:00:00     20  alloc hpa-wn[05-24]
v100-al9                        up 5-00:00:00      2    mix hp-teslav[01,06]
v100-al9                        up 5-00:00:00      4   idle hp-teslav[02-05]
v100-al9_short                  up    6:00:00      2    mix hp-teslav[01,06]
v100-al9_short                  up    6:00:00      4   idle hp-teslav[02-05]
v100-al9_long                   up 7-00:00:00      2    mix hp-teslav[01,06]
v100-al9_long                   up 7-00:00:00      4   idle hp-teslav[02-05]
a100-al9                        up 5-00:00:00      2    mix hp-teslaa[01,03]
a100_long-al9                   up 7-00:00:00      2    mix hp-teslaa[01,03]
a100_short-al9                  up    6:00:00      2    mix hp-teslaa[01,03]
a100_devel-al9                  up      20:00      2    mix hp-teslaa[01,03]
 .
 .
 .
```

STATE

idle	:	The node is not allocated to any jobs and is available for use.
alloc	:	The node has been allocated to one or more jobs.
mix	:	The node has some of its CPUs **ALLOCATED** while others are **IDLE**.
down	:	The node is unavailable for use.
drain	:	The node is unavailable for use per system administrator request.
drang	:	The node is currently allocated a job, but will not be allocated additional jobs.

Basic Usage of Slurm System

User commands : sinfo

- Query partitions and nodes CPU and GPU Usage by Node:

  sinfo -o "%.30P %.5a %.6D %.15A %.16C"

  sinfo -O NodeHost:15,CPUsState:20,Gres:10,GresUsed

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sinfo  -o "%.30P %.5a %.6D %.15A %.16C"
                  PARTITION AVAIL  NODES      NODES(A/I)    CPUS(A/I/O/T)
          reservation_b6000    up      1             1/0       16/48/0/64
        reservation_b6000_2    up      1             1/0       16/48/0/64
            bigmemory_hdr1    up      2             2/0     192/64/0/256
                    golbal    up     30            30/0  4790/970/0/5760
           edr1-al9_short*    up     20            20/0  3150/690/0/3840
      edr1-al9_short_serial    up     20            20/0  3150/690/0/3840
            edr1-al9_large    up     20            20/0  3150/690/0/3840
       edr1-al9_long_serial    up     20            20/0  3150/690/0/3840
    edr1-al9_moderate_serial    up     20            20/0  3150/690/0/3840
            edr2-al9_short    up     10            10/0  1640/280/0/1920
                         .
                         .
                         .
```

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sinfo -O NodeHost:15,CPUsState:20,Gres:10,GresUsed
 HOSTNAMES      CPUS(A/I/O/T)       GRES      GRES_USED
 hp-rtx6kp02    16/48/0/64          gpu:8     gpu:8
  .
  .
  .
 hpi-wn03       100/28/0/128        (null)    (null)
 hpi-wn04       60/68/0/128         (null)    (null)
 hp-teslav01    16/32/0/48          gpu:8     gpu:8
 hp-teslav02    40/8/0/48           gpu:8     gpu:8
 hp-teslav03    16/32/0/48          gpu:8     gpu:8
 hp-teslav04    14/34/0/48          gpu:8     gpu:7
 hp-teslav06    16/16/0/32          gpu:8     gpu:8
 hp-teslav05    0/48/0/48           gpu:8     gpu:0
```

Basic Usage of Slurm System

User commands : sbatch

- Submit your job with bash script (recommended) : sbatch your_script.sh

your_script.sh

```
#!/bin/bash
#SBATCH --job-name=hello_world                # Job name
#SBATCH --partition=intel-g4-al9_short        # Partition name
#SBATCH --nodes=1                             # Number of nodes
#SBATCH --ntasks-per-node=1                   # Number of tasks per node
#SBATCH --cpus-per-task=1                     # Number of CPUs per task
#SBATCH --output=%j.out                       # Standard output file (%j: Job ID)
#SBATCH --error=%j.err                        # Standard error file (%j: Job ID)
#SBATCH --time=00-01:00:00                    # Time limit
#SBATCH --mail-type=ALL                       # Email notifications = BEGIN, END, FAIL, ALL
#SBATCH --mail-user=jennifer.chen@twgrid.org  # Email address to send notifications

date
echo "Hello DiCOS Users!"
echo "Hello World!"
hostname
```

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sbatch your_script.sh
Submitted batch job 812234
[yiruchen0101@slurm-ui01 yiruchen0101]$
```

Job ID

Basic Usage of Slurm System

User commands : sacct

- Query the jobs submitted by you : sacct or sacct -u <your account>

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sacct
JobID           JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
812234       helloworld edr1-al9_+       asgc          1  COMPLETED      0:0
812234.batch      batch                  asgc          1  COMPLETED      0:0
812234.0           echo                  asgc          1  COMPLETED      0:0
```

Basic Usage of Slurm System

User commands : squeue

- Show queue information : squeue
- Show your job in the queue : squeue -u <your account>

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ squeue
  JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
 869316  a100-al9 symbreak    gtelo PD       0:00      1 (Priority)
 869315  a100-al9 symbreak    gtelo PD       0:00      1 (Nodes required for
job are DOWN, DRAINED or reserved for jobs in higher priority partitions)
 851811  a100-al9    AIREC hsinyich  R 2-18:02:00      1 hp-teslaa03
 829335  a100-al9    AIREC hsinyich  R 4-08:28:54      1 hp-teslaa03
 736972 a100_deve    test0 yungilin PD       0:00      1 (Resources)
    .
    .
    .
```

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ squeue -u yiruchen0101
  JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
 949585 intel-g4- hellowor yiruchen PD       0:00      1 (None)
```

Basic Usage of Slurm System

User commands : scancel

- Cancel your job : scancel <your JobID>

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sbatch job
Submitted batch job 949585
[yiruchen0101@slurm-ui01 yiruchen0101]$ scancel 949585
[yiruchen0101@slurm-ui01 yiruchen0101]$ squeue -u yiruchen0101
   JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[yiruchen0101@slurm-ui01 yiruchen0101]$
```

Basic Usage of Slurm System

User commands : scontrol

- Show the detailed job information : scontrol show job <your JobID>
- Show the detailed node information : scontrol show node <node name>

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ scontrol show job 949681
JobId=949681 JobName=helloword
   UserId=yiruchen0101(5019) GroupId=ASGC(525) MCS_label=N/A
   Priority=25084 Nice=0 Account=asgc QOS=normal
   JobState=PENDING Reason=None Dependency=(null)
   Requeue=0 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=01:00:00 TimeMin=N/A
   SubmitTime=2025-01-12T18:22:54 EligibleTime=2025-01-12T18:22:54
   AccrueTime=2025-01-12T18:22:54
   StartTime=Unknown EndTime=Unknown Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2025-01-12T18:22:54 Scheduler=Main
   Partition=intel-g4-al9_short AllocNode:Sid=slurm-ui04:1885932
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=
   NumNodes=1-1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   ReqTRES=cpu=1,mem=927000M,node=1,billing=1
   AllocTRES=(null)
   Socks/Node=* NtasksPerN:B:S:C=1:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/ceph/work/ASGC/yiruchen0101/job
   WorkDir=/ceph/work/ASGC/yiruchen0101
   StdErr=/ceph/work/ASGC/yiruchen0101/949681.err
   StdIn=/dev/null
   StdOut=/ceph/work/ASGC/yiruchen0101/949681.out
   TresPerTask=cpu=1
   MailUser=jennifer.chen@twgrid.org MailType=INVALID_DEPEND,BEGIN,END,FAIL,REQUEUE,STAGE_OUT
```

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ scontrol show node hpa-wn05
NodeName=hpa-wn05 Arch=x86_64 CoresPerSocket=96
   CPUAlloc=174 CPUEfctv=192 CPUTot=192 CPULoad=111.72
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=hpa-wn05 NodeHostName=hpa-wn05 Version=24.05.1
   OS=Linux 5.14.0-503.16.1.el9_5.x86_64 #1 SMP PREEMPT_DYNAMIC Fri Dec 13 01:47:05 EST
2024
   RealMemory=1386000 AllocMem=1031680 FreeMem=1444233 Sockets=2 Boards=1
   MemSpecLimit=10240
   State=MIXED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=golbal,edr1-al9_short,edr1-al9_short_serial,edr1-al9_large,edr1-
al9_long_serial,edr1-al9_moderate_serial
   BootTime=2026-04-28T07:25:52 SlurmdStartTime=2026-05-18T08:25:38
   LastBusyTime=2026-05-15T20:20:28 ResumeAfterTime=None
   CfgTRES=cpu=192,mem=1386000M,billing=192
   AllocTRES=cpu=174,mem=1007.50G
   CurrentWatts=0 AveWatts=0
```

Environment Modules (Lmod)

Environment Modules (Lmod)

Introduction

- In DiCOS Slurm system, we have environment modules installed in user interfaces and worker nodes.
- Detailed information please refer to the original document :

  https://modules.readthedocs.io/en/latest/
- Environment-modules help user to setup environment and environment variables properly for specific software environments.
  - User doesn’t need to worry about the complex settings of the environments.

Environment Modules (Lmod)

User commands : module avail

- Show available modules in slurm-ui : module avail or ml av

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ module avail
------------------------------------------- /ceph/sharedfs/apps/general/almalinux9/modulefiles -------------------------------------------
 R/4.2.1                    gcc/11.5.0                          lapack/v3.12.1                              nvhpc/25.1        (D)
 R/4.4.0                    gcc/13.1.0                   (D)    mkl/latest                                  python3/3.9.23
 R/4.4.1             (D)    gromacs/2022.swaxs-0.1/a100-gpu     mkl/2023.2.0                         (D)    relion/4.0.2
 anaconda3/2024.10-1 (D)    gromacs/2022.swaxs-0.1/cpu          mpi/openmpi-4.1.5/cuda-12.2/gcc-11.5.0      relion/5.0.0      (D)
 aocc/5.0.0          (D)    gromacs/2022.swaxs-0.1/v100-gpu (D) mpi/openmpi-5.0.5/cuda-12.4/gcc-13.1.0      samtools/1.21
 bcftools/1.21              gromacs/2024.0/a100_gpu-al9         mpi/openmpi-5.0.8/gcc-11.5.0                singularity/4.1.2
 cmake/3.28          (D)    gromacs/2024.0/cpu                  nvhpc-byo-compiler/25.1              (D)    tbb/latest
 cuda/12.2.0                gromacs/2024.0/gpu           (D)    nvhpc-hpcx-2.20-cuda12/25.1                 tbb/2021.10.0     (D)
 cuda/12.4.0                gromacs/2025.2.0/l40s_gpu-al9       nvhpc-hpcx-cuda12/25.1               (D)
 cuda/12.6.0         (D)    htslib/1.21                         nvhpc-hpcx/25.1                      (D)
 dorado/v0.8.3              julia/1.11.5                 (D)    nvhpc-nompi/25.1                     (D)
------------------------------------------- /ceph/sharedfs/apps/amd/almalinux9/hdr1-al9/modulefiles -------------------------------------------
 aocc/4.2.0    gcc/11.5.0    hdf5/1.8.12                mpi/openmpi-5.0.3/aocc420
 aocl/4.2.0    gcc/13.1.0    mpi/mpich-4.2.2/gcc13.1.0  mpi/openmpi-5.0.3/gcc13.1.0 (D)
------------------------------------------- /ceph/sharedfs/apps/general/centos7/modulefiles -------------------------------------------
 R/4.0.5             gcc/4.8.5        gromacs/2021.swaxs-0.1/gpu        mpi/openmpi-2.0.2/gcc485         nvhpc-nompi/20.11
 R/4.2.1             gcc/8.2.0        gromacs/2022.swaxs-0.1/gpu        mpi/openmpi-3.1.6/cuda/gcc930    nvhpc-nompi/24.3
 anaconda3/4.9.2     gcc/8.3.0        gromacs/2024.0/a100_gpu           mpi/openmpi-3.1.6/gcc930         nvhpc-openmpi3/24.3
 anaconda3/4.10.3    gcc/9.3.0        gromacs/2024.0/cpu                mpi/openmpi-4.1.0/gcc485         nvhpc/20.11
 anaconda3/4.12.0    gcc/10.3.0       gromacs/2024.0/gpu                mumax3/cuda-11.3/10_linux        nvhpc/24.3
 binutils/2.35.2     gcc/11.1.0       hdf5/1.10.5-parallel              nvhpc-byo-compiler/20.11         paraview/5.8.0
 cmake/3.20.3        gcc/12.1.0       hdf5/1.10.5                (D)    nvhpc-byo-compiler/24.3          python/3.9.5
 cuda/11.3.0         git/2.37.1       julia/1.8.0                       nvhpc-hpcx-cuda12/24.3           python/3.9.18       (D)
 cuda/11.8.0         golang/1.21.7    make/4.3                          nvhpc-hpcx/24.3                  root/v6.30
------------------------------------------- /ceph/sharedfs/apps/intel/centos7/modulefiles -------------------------------------------
 clck/2021.6.0           compiler/2022.1.0      (D)    dpl/2021.7.0           inspector/2022.1.0    mkl/2022.1.0
 compiler-rt/2022.1.0    debugger/2021.6.0             icc/2022.1.0           intel_mpi/2021.6.0    tbb/2021.6.0
 compiler/2022.0.2       dev-utilities/2021.6.0        init_opencl/2022.1.0   itac/2021.6.0
```

Environment Modules (Lmod)

User commands : module list & module load

- Show currently loaded modules : module list or ml
- Load module : module load <module name> or ml <module name>

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ module list
No modules loaded
[yiruchen0101@slurm-ui01 yiruchen0101]$ module load mpi/mpich-4.2.2/gcc13.1.0
[yiruchen0101@slurm-ui01 yiruchen0101]$ module list
Currently Loaded Modules:
  1) gcc/13.1.0   2) mpi/mpich-4.2.2/gcc13.1.0
```

Environment Modules (Lmod)

User commands : module unload & module purge

- Unload module : module unload <module name> or ml -<module name>
- Unload all loaded module : module purge or ml purge

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ module list
Currently Loaded Modules:
  1) gcc/13.1.0   2) mpi/mpich-4.2.2/gcc13.1.0
[yiruchen0101@slurm-ui01 yiruchen0101]$ module unload gcc/13.1.0
[yiruchen0101@slurm-ui01 yiruchen0101]$ module list
Currently Loaded Modules:
  1) mpi/mpich-4.2.2/gcc13.1.0
[yiruchen0101@slurm-ui01 yiruchen0101]$ module load mpi/mpich-4.2.2/gcc13.1.0
[yiruchen0101@slurm-ui01 yiruchen0101]$ module list
Currently Loaded Modules:
  1) gcc/13.1.0   2) mpi/mpich-4.2.2/gcc13.1.0
[yiruchen0101@slurm-ui01 yiruchen0101]$ module purge
[yiruchen0101@slurm-ui01 yiruchen0101]$ module list
No modules loaded
```

Python, Compilation and MPI Enviroment

Python, Compilation and MPI Enviroment

Python

- The default system python version is 3.9.18.
- Use python , please consider using anaconda with python first :

  module load anaconda3

- Install additional applications : /ceph/work/<groupname>/

Python, Compilation and MPI Enviroment

Python

- Before installing a special python package, use the virtual environment :

conda create --name <myenv>

conda env list

conda activate <myenv>

conda install <your_package>

conda deactivate

```
[yiruchen0101@slurm-ui03 yiruchen0101]$ mkdir -p /ceph/work/ASGC/yiruchen0101/conda_envs
[yiruchen0101@slurm-ui03 yiruchen0101]$ module load anaconda3
[yiruchen0101@slurm-ui03 yiruchen0101]$ conda config --add envs_dirs /ceph/work/ASGC/yiruchen0101/conda_envs/
[yiruchen0101@slurm-ui03 yiruchen0101]$ conda create -n py311 python=3.11
  Retrieving notices: ...working... done
  Channels:
   - defaults
  Platform: linux-64
  Collecting package metadata (repodata.json): done
  Solving environment: done
  Proceed ([y]/n)? y
                .
                .
                .
  Downloading and Extracting Packages:

  Preparing transaction: done
  Verifying transaction: done
  Executing transaction: done
  #
  # To activate this environment, use
  #
  #     $ conda activate py311
  #
  # To deactivate an active environment, use
  #
  #     $ conda deactivate
  [yiruchen0101@slurm-ui03 yiruchen0101]$ conda env list
  # conda environments:
  #
  base                  /ceph/sharedfs/software/compiler/anaconda/almalinxu9/x86_el9
  py311                 /ceph/work/ASGC/yiruchen0101/conda_envs/py311
  (base) [yiruchen0101@slurm-ui03 yiruchen0101]$ conda activate py311
  (/ceph/work/ASGC/yiruchen0101/conda_envs/py311) [yiruchen0101@slurm-ui03 yiruchen0101]$ conda install numpy
```

https://docs.conda.io/projects/conda/en/latest/commands/index.html#

Python, Compilation and MPI Enviroment

Compilation

- Before compiling, enter an interactive session on the desired CPU type (Intel or AMD) using Slurm :
  salloc —partition=(edr1-al9_short / intel-g4-al9_short) ---ntasks=1 --tasks-per-node=1

  - Intel compiler

    module load icc/2022.1.0
  - AMD compiler

    module load aocc/4.2.0
  - GCC

    module load gcc/12.1.0
  - nvidia development kit (nvcc, for GPU program development)

    module load nvhpc/24.3

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ salloc --partition=edr1-al9_short --ntasks=1 --tasks-per-node=1
salloc: Pending job allocation 1583407
salloc: job 1583407 queued and waiting for resources
salloc: job 1583407 has been allocated resources
salloc: Granted job allocation 1583407
[yiruchen0101@slurm-ui01 yiruchen0101]$ module load aocc/4.2.0
[yiruchen0101@slurm-ui01 yiruchen0101]$ srun hostname
hpa-wn05.twgrid.org
[yiruchen0101@slurm-ui01 yiruchen0101]$ exit
exit
salloc: Relinquishing job allocation 1583407
salloc: Job allocation 1583407 has been revoked.
```

Python, Compilation and MPI Enviroment

MPI enviroment

- Load different MPI implementation
  - mpich

    module load mpi/mpich-4.2.2/gcc13.1.0
  - openmpi

    module load mpi/openmpi-4.1.0/gcc485

    module load mpi/openmpi-5.0.3/aocc420

Available Software

Available Software

software

- Anaconda Python packages：

  TensorFlow, PyTorch, PyRoot ……

  - CUDA version on GPU cluster : 12.6

- Some Customized Requirements needs to deploy by Singularity.

  AlphaFold. (* Build fee)

Job Submission Examples

Job Submission Examples

Linux commands

- Show current path : pwd
- List current folder’s file : ls
- Change directory : cd <path>
- Create directory : mkdir <dir>
- Edit text file : vim <file>
- Copy : cp <file> <new file>
- Move or change file name : mv <file> <path or new file name>
- Remove file : rm <file>
- Show manual : man <command>

PuTTY : https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

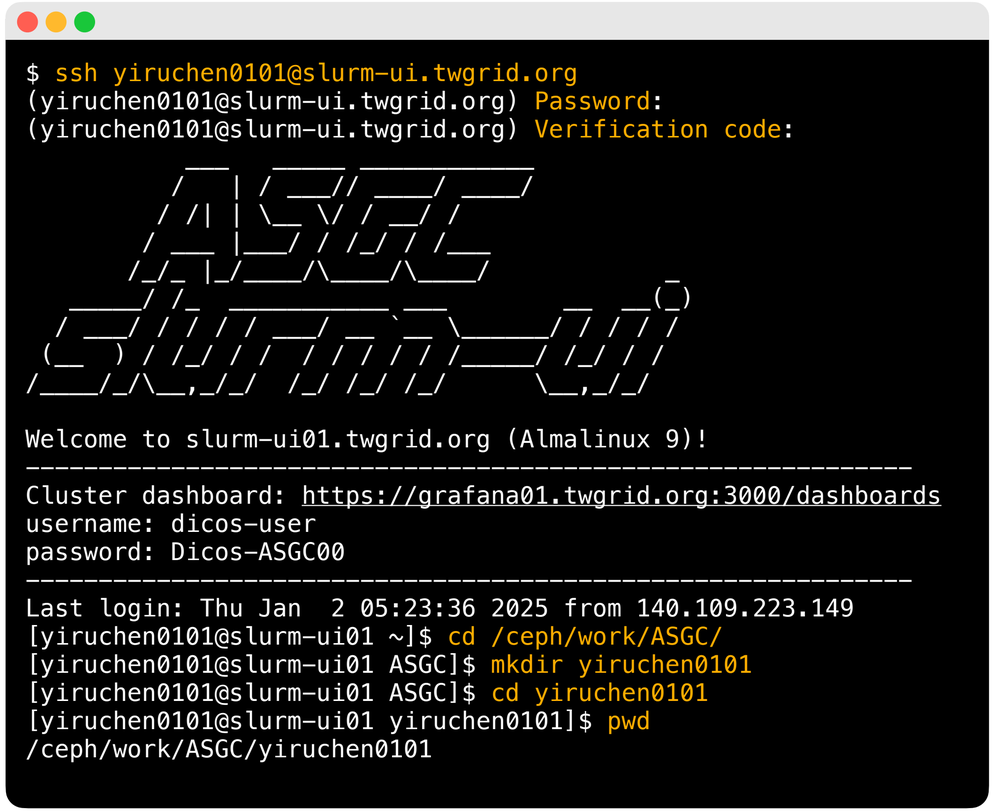
Mobaxterm : https://mobaxterm.mobatek.net/download-home-edition.html

VScode : https://code.visualstudio.com/download

Job Submission Examples

Preparation

- Open your terminal
- Login : ssh <your account>@slurm-ui.twgrid.org
- Go to the working folder : cd /ceph/work/{group}
- Build your folder : mkdir <your account>
- Go to your folder : cd <your account>



Job Submission Examples

Example 1 - Simple Job Submission (Hello World)

- Prepare a user defined shell script : 01_hello_world.sh
- Submit the job with sbatch

  sbatch 01_hello_world.sh

01_hello_world.sh

```
date
echo "Hello DiCOS Users!”
hostname
```

slurm-%J.out

```
三  6月 19 22:03:40 UTC 2024
Hello DiCOS Users!
hpa-wn05.twgrid.org
```

01_hello_world_2.sh

```
#!/bin/bash
#SBATCH --job-name=hello_world                # Job name
#SBATCH --partition=intel-g4-al9_short        # Partition name
#SBATCH --nodes=1                             # Number of nodes
#SBATCH --ntasks-per-node=1                   # Number of tasks per node
#SBATCH --cpus-per-task=1                     # Number of CPUs per task
#SBATCH --output=%j.out                       # Standard output file (%j: Job ID)
#SBATCH --error=%j.err                        # Standard error file (%j: Job ID)
#SBATCH --time=00-01:00:00                    # Time limit
#SBATCH --mail-type=ALL                       # Email notifications = BEGIN, END, FAIL, ALL
#SBATCH --mail-user=jennifer.chen@twgrid.org  # Email address to send notifications

date
echo "Hello DiCOS Users!”
echo "Hello World!”
hostname
```

Job Submission Examples

Example 2 - Submit a MCORE job

- You will need to assign in your preamble of the script for the requesting resources.

  E.g. 02_mcore.sh

- Submit job : sbatch 02_mcore.sh

- This example will submit a job which requesting 10 CPU cores

02_mcore.sh

```
#!/bin/bash
#SBATCH --job-name=stress_test                # Define the name of your job
#SBATCH --partition=intel-g4-al9_short        # Partition name
#SBATCH --nodes=1                             # Numbers of nodes
#SBATCH --ntasks-per-node=1                   # Number of tasks per node
#SBATCH --cpus-per-task=10                    # Number of CPUs per task
#SBATCH --output=%j.out                       # Standard output file (%j: Job ID)
#SBATCH --error=%j.err                        # Standard error file (%j: Job ID)
#SBATCH --time=00-01:00:00                    # Time limit
#SBATCH --mail-type=ALL                       # Email notifications = BEGIN, END, FAIL, ALL
#SBATCH --mail-user=jennifer.chen@twgrid.org  # Email address to send notifications


srun stress -c 10 -t 100
```

job.%J.out

```
stress: info: [501543] dispatching hogs: 10 cpu, 0 io, 0 vm, 0 hdd
stress: info: [501543] successful run completed in 100s
```

Job Submission Examples

Example 3 - Submit a python job using anaconda3 python3

- Prepare a python script that calculate $\int_0^1 x^2 dx$: 03_integrate.py

03_integrate.py

```
# integrate_x2.py

import numpy as np

def f(x):
    return x**2

def integrate(f, a, b, num_points):
    x = np.linspace(a, b, num_points)
    y = f(x)
    dx = (b - a) / (num_points - 1)
    integral = np.sum(y) * dx
    return integral

a, b = 0, 1  # upper and lower limits of integral
num_points = 1000000  # number of sample points
result = integrate(f, a, b, num_points)
print(f"The integral of x^2 from {a} to {b} is approximately = {result}")
```

Job Submission Examples

Example 3 - Submit a python job using anaconda3 python3

- Prepare a shell script that wrapping the environment modules and run python script: 03_integrate.sh
- Submit job using sbatch : sbatch 03_integrate.sh

03_integrate.sh

```
#!/bin/bash
#SBATCH --job-name=stress_test                # Define the name of your job
#SBATCH --partition=intel-g4-al9_short        # Partition name
#SBATCH --nodes=1                             # Numbers of nodes
#SBATCH --ntasks-per-node=1                   # Number of tasks per node
#SBATCH -cpus-per-task=1                      # Number of CPUs per task
#SBATCH --output=%j.out                       # Standard output file (%j: Job ID)
#SBATCH --error=%j.err                        # Standard error file (%j: Job ID)
#SBATCH --time=00-01:00:00                    # Time limit
#SBATCH --mail-type=ALL                       # Email notifications = BEGIN, END, FAIL, ALL
#SBATCH --mail-user=jennifer.chen@twgrid.org  # Email address to send notifications

module load anaconda3/4.9.2
python 03_integrate.py
```

job.%J.out

```
The integral of x^2 from 0 to 1 is approximately = 0.33333383333399996
```

Job Submission Examples

Example 4 - Submit array job

- Prepare a shell script for array jobs : 04_array.sh
- Submit job using sbatch : sbatch 04_array.sh

04_array.sh

```
#!/bin/bash
#SBATCH --job-name=array_job                 # Job name
#SBATCH --partition=edr1-al9_short           # Partition name
#SBATCH --nodes=1                            # Number of nodes
#SBATCH --ntasks-per-node=1                  # Number of tasks per node
#SBATCH --cpus-per-task=1                    # Number of CPUs per task
#SBATCH --array=0-4                          # Array job
#SBATCH --output=%A_%a.out                   # Standard output file (%j: Job ID, %A: Array Job ID, %a: Task ID )
#SBATCH --error=%A_%a.err                    # Standard error file (%j: Job ID, %A: Array Job ID, %a: Task ID)
#SBATCH --time=00-01:00:00                   # Time limit
#SBATCH --mail-type=ALL                      # Email notifications = BEGIN, END, FAIL, ALL
#SBATCH --mail-user=jennifer.chen@twgrid.org  # Email address to send notifications


date
hostname
TASK_ID=${SLURM_ARRAY_TASK_ID}
DURATION=$(sed -n "$((TASK_ID+1))p" stress_args.txt)
echo "Array Job ID : $SLURM_ARRAY_JOB_ID"
echo "Task ID      : $TASK_ID"
echo "Stress Time  : ${DURATION}s"

srun stress -c 1 -t ${DURATION}
```

stress_args.txt

```
10
20
30
40
50
```

%A_%a.out

```
Tue May 26 08:33:27 UTC 2026
hpa-wn15.twgrid.org
Array Job ID : 3465516
Task ID      : 0
Stress Time  : 10s
stress: info: [2676362] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
stress: info: [2676362] successful run completed in 10s
```

Job Submission Examples

Example 5 - Submit dependency job

- Prepare shell script for preprocessing and analysis steps: 05_preprocess.sh 05_analysis.sh

05_preprocess.sh

```
#!/bin/bash
#SBATCH --job-name=preprocess                 # Define the name of your job
#SBATCH --partition=edr1-al9_short            # Partition name
#SBATCH --nodes=1                             # Numbers of nodes
#SBATCH --ntasks-per-node=1                   # Number of tasks per node
#SBATCH -cpus-per-task=1                      # Number of CPUs per task
#SBATCH --output=%j.out                       # Standard output file (%j: Job ID)
#SBATCH --error=%j.err                        # Standard error file (%j: Job ID)
#SBATCH --time=00-01:00:00                    # Time limit
#SBATCH --mail-type=ALL                       # Email notifications = BEGIN, END, FAIL, ALL
#SBATCH --mail-user=jennifer.chen@twgrid.org  # Email address to send notifications

echo "[$(date)] Start preprocessing”
hostname
sleep 30
echo "[$(date)] Preprocessing finished"
```

05_analysis.sh

```
#!/bin/bash
#SBATCH --job-name=analysis                   # Define the name of your job
#SBATCH --partition=edr1-al9_short            # Partition name
#SBATCH --nodes=1                             # Numbers of nodes
#SBATCH --ntasks-per-node=1                   # Number of tasks per node
#SBATCH -cpus-per-task=1                      # Number of CPUs per task
#SBATCH --output=%j.out                       # Standard output file (%j: Job ID)
#SBATCH --error=%j.err                        # Standard error file (%j: Job ID)
#SBATCH --time=00-01:00:00                    # Time limit
#SBATCH --mail-type=ALL                       # Email notifications = BEGIN, END, FAIL, ALL
#SBATCH --mail-user=jennifer.chen@twgrid.org  # Email address to send notifications

echo "[$(date)] Start analysis”
hostname
sleep 60
echo "[$(date)] Preprocessing analysis"
```

Job Submission Examples

Example 5 - Submit dependency job

- Prepare a shell script that submits the first job (preprocessing) and captures its Job ID: 05_submit_workflow.sh
- Submit job using bash : bash 05_submit_workflow.sh

05_submit_workflow.sh

```
#!/bin/bash
# submit preprocessing job
JOB1=$(sbatch 05_preprocess.sh | awk '{print $4}')
echo "Preprocess Job ID = $JOB1"
# submit analysis job (only runs if preprocess succeeds)
JOB2=$(sbatch --dependency=afterok:$JOB1 05_analysis.sh | awk '{print $4}')
echo "Analysis Job ID = $JOB2"
```

```
[yiruchen0101@slurm-ui01 yiruchen0101]$ sh 05_submit_workflow.sh
Preprocess Job ID = 3466743
Analysis Job ID = 3466744
[yiruchen0101@slurm-ui01 yiruchen0101]$ squeue -u yiruchen0101
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           3466744 edr1-al9_ analysis yiruchen PD       0:00      1 (Dependency)
           3466743 edr1-al9_ preproce yiruchen  R       0:10      1 hpa-wn11
```

%J.out

```
[Tue May 26 11:20:18 UTC 2026] Start preprocessing
hpa-wn11
[Tue May 26 11:20:48 UTC 2026] Preprocessing finished
[Tue May 26 11:20:49 UTC 2026] Start analysis
hpa-wn11
[Tue May 26 11:21:49 UTC 2026] Preprocessing analysis
```

Problem Report and FAQ

- Online documents: https://dicos.grid.sinica.edu.tw/wiki/
- Email channel to ASGC admins: DiCOS-Support@twgrid.org
- Regular face-to-face (on-site) video conferences:

  ASGC DiCOS user meetings held every Wednesday at 14:30 (UTC+8), please ask our staff for meeting information.