Introduction on Slurm Job Submission

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

2026/01/07 Yi-Ru CHEN

Outline

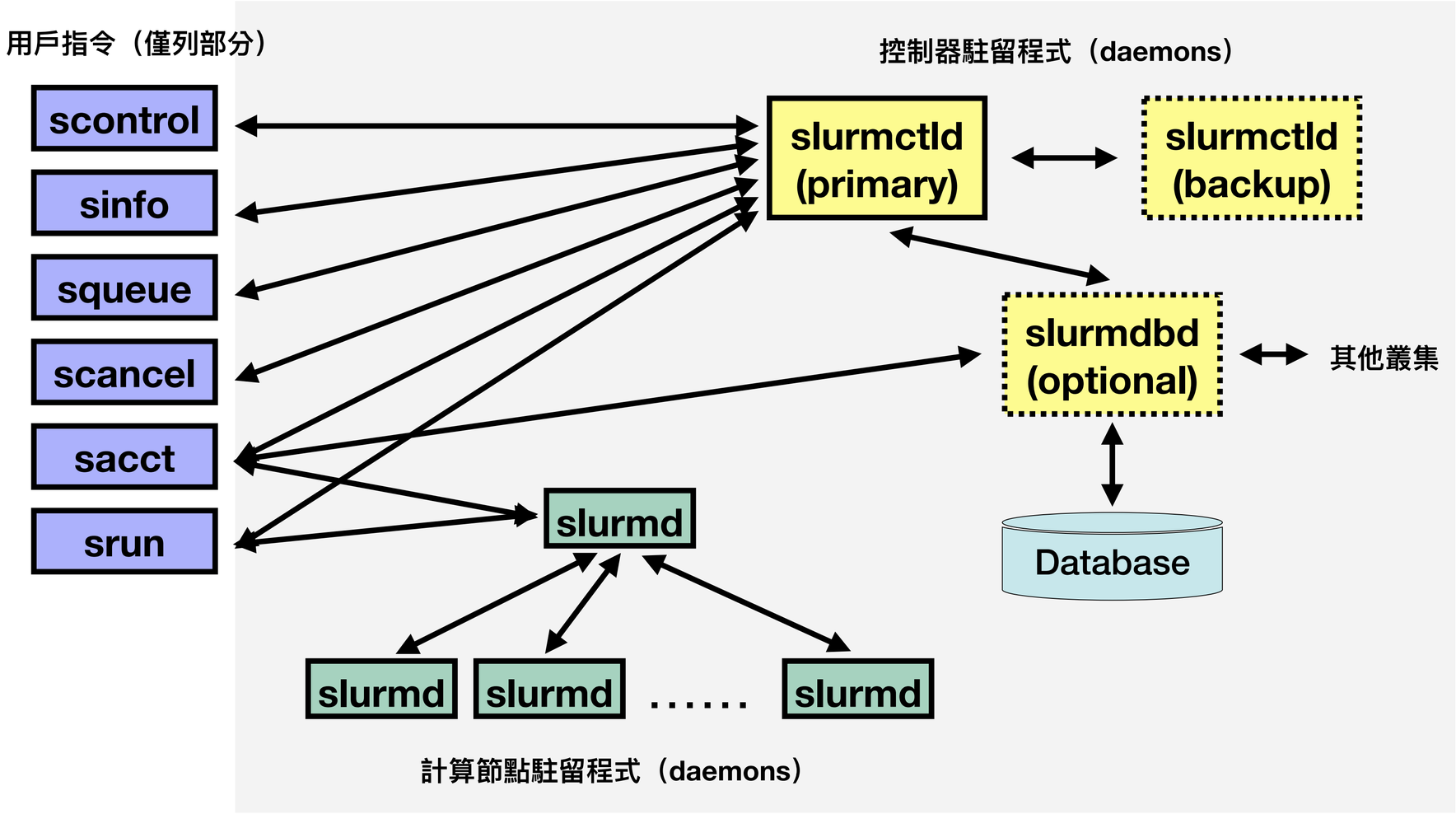
- Introduction of Slurm
- User Interfaces (Login Node)
- Basic Usage of Slurm System
- Environment Modules (Lmod)
- Python, Compilation and MPI Enviroment
- Available Software
- Job Submission Example

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
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-L40S	NVIDIA L40S	1	4	AMD EPYC 9374F 32-Core Processor	64

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

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 :

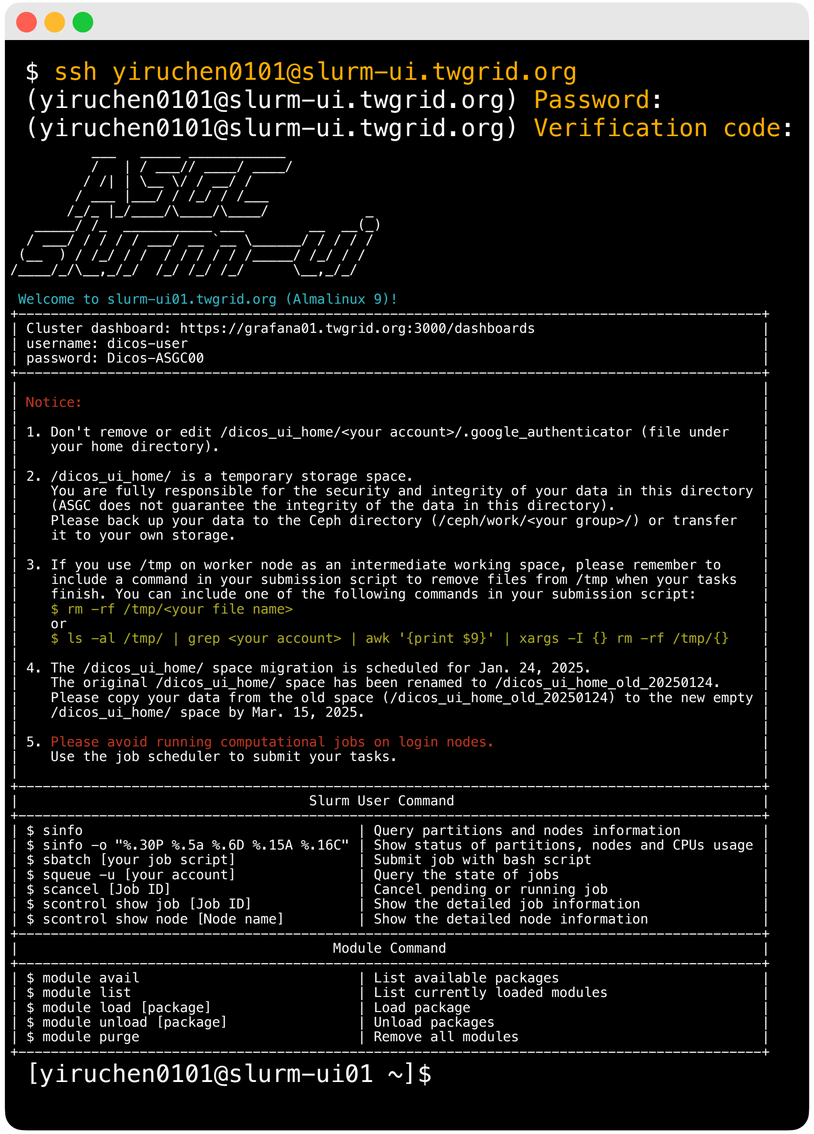
  ssh <your account>@slurm-ui.twgrid.org

  Enter your password and 2FA verification code

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

- 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.



User Interfaces (Login Nodes)

Mount Space

- User space (100 GB) : /dicos_ui_home/{user}
- Group Space (3 TB) : /ceph/work/{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
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	
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	
L40S	l40s-al9	5-00:00:0	sm-gpu21
	l40s-al9_short	06:00:00	
	l40s-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.
  - MaxJobsPA：Maximum number of running jobs per group.

	Partition	MaxNodes	MaxTRES	MinTRES	MaxJobsPA
CPU	**XXX_short_serial**	N/A	CPU=24	N/A	N/A
	XXX_long_serial	**1**	N/A	N/A	45
	XXX_moderate_serial	N/A	CPU=24	N/A	315
	XXX_large	N/A	N/A	N/A	45
GPU	**XXX**	N/A	gres/gpu=8	gres/gpu=1	N/A
	XXX_short	N/A	gres/gpu=8	gres/gpu=1	N/A
	XXX_long	N/A	gres/gpu=8	gres/gpu=1	N/A

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]
l40s-al9                        up 5-00:00:00      1    mix sm-gpu21
l40s-al9_short                  up    6:00:00      1    mix sm-gpu21
l40s-al9_long                   up 7-00:00:00      1    mix sm-gpu21
```

```
[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]
l40s-al9                        up 5-00:00:00      1    mix sm-gpu21
l40s-al9_short                  up    6:00:00      1    mix sm-gpu21
l40s-al9_long                   up 7-00:00:00      1    mix sm-gpu21
```

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 : sbatch

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

your_script.sh

```
#!/bin/bash
#SBATCH --job-name=helloword                  # Job name
#SBATCH --partition=intel-g4-al9_short        # Partiotion 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


srun /bin/echo "Hello World!"
```

```
[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]$ scancel 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=121 CPUEfctv=192 CPUTot=192 CPULoad=108.93
   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=682880 FreeMem=1163994 Sockets=2 Boards=1
   MemSpecLimit=10240
   State=MIXED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=edr1-al9_short,edr1-al9_short_serial,edr1-al9_large,edr1
al9_long_serial,edr1-al9_moderate_serial
   BootTime=2025-08-18T02:20:55 SlurmdStartTime=2025-09-06T06:58:33
   LastBusyTime=2025-09-04T00:29:57 ResumeAfterTime=None
   CfgTRES=cpu=192,mem=1386000M,billing=192
   AllocTRES=cpu=121,mem=682880M
   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

```
(base) [yiruchen0101@slurm-ui03 yiruchen0101]$ conda create --name python3.8 python=3.8
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done
                .
                .
                .
# To activate this environment, use
#
#     $ conda activate python3.8
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) [yiruchen0101@slurm-ui03 yiruchen0101]$ conda env list
# conda environments:
#
base                  *  /ceph/work/ASGC/yiruchen0101/anaconda3
python3.8                /ceph/work/ASGC/yiruchen0101/anaconda3/envs/python3.8
(base) [yiruchen0101@slurm-ui03 yiruchen0101]$ conda activate python3.8
(python3.8) [yiruchen0101@slurm-ui03 yiruchen0101]$
(python3.8) [yiruchen0101@slurm-ui03 yiruchen0101]$ conda install numpy
(python3.8) [yiruchen0101@slurm-ui03 yiruchen0101]$ conda deactivate
(base) [yiruchen0101@slurm-ui03 yiruchen0101]$
```

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

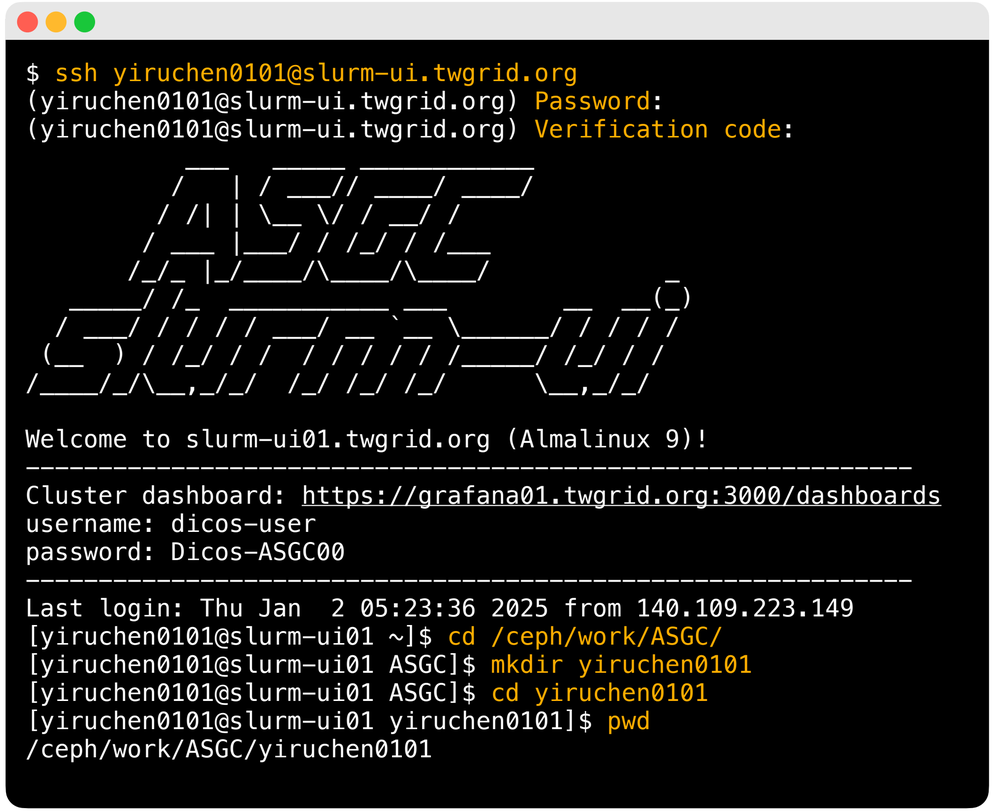
Mobaxterm : https://mobaxterm.mobatek.net/download-home-edition.html

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
```

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        # Partiotion 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        # Partiotion 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
```

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.

Questionnaire