Introduction on Slurm Job Submission

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

2024/10/30 Yi-Ru CHEN

Outline

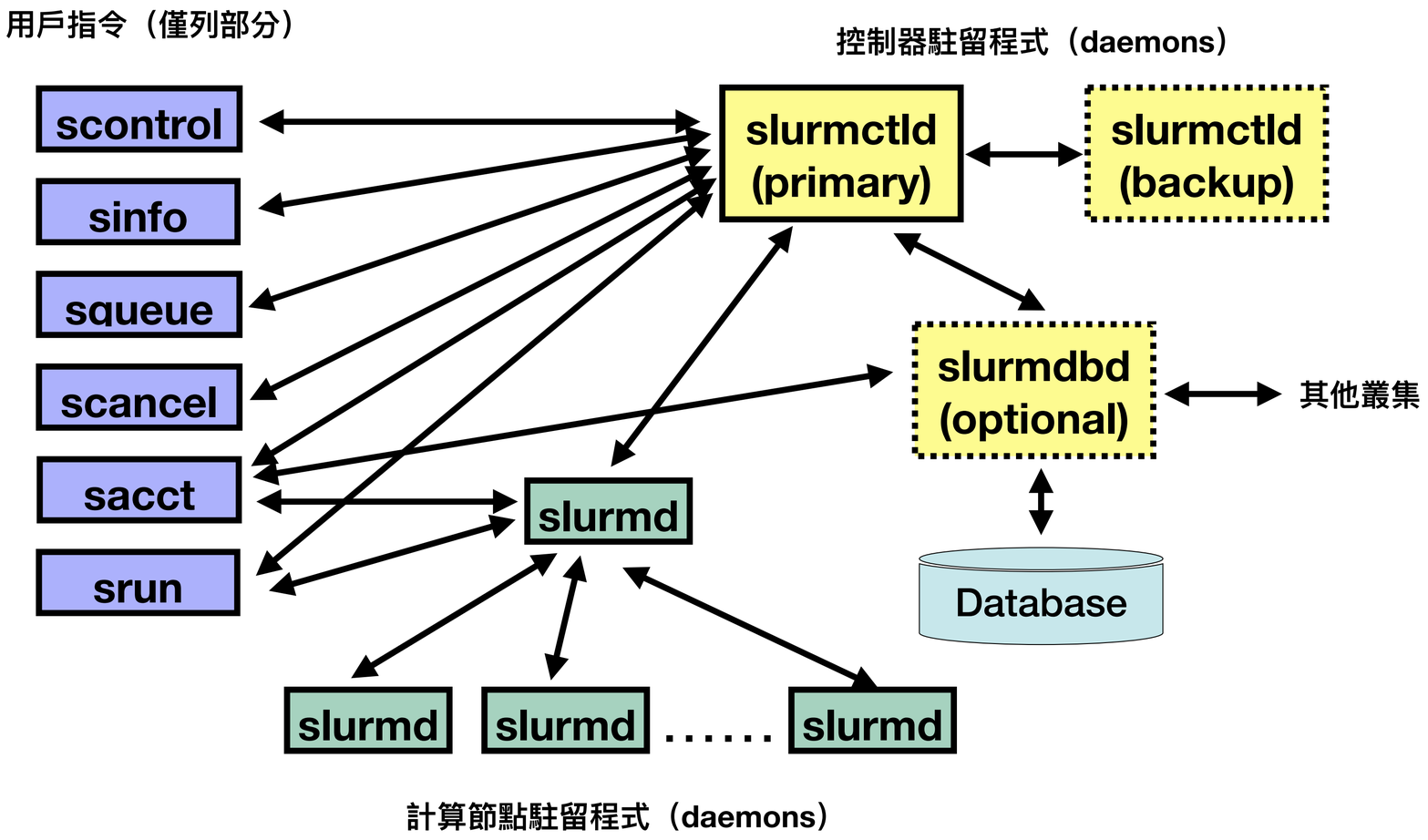
- Introduction of Slurm
- User Interfaces (Login Nodes)
- Basic Usage of Slurm System
- Environment Modules (Lmod)
- Python, Compilation and MPI Enviroment
- Slurm Job Submission Examples - Hands on

.

Introduction of Slurm

Overview

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



Introduction of slurm - Computing

- Computing Machine Specifications Computing Nodes :

CPU Cluster	CPU Model	Nodes	RAM-Per-Node	Cores-Per-Node	Total Cores
HDR1	AMD Rome 7662@2.0GHz	6	1.5 TB	128	768
EDR1	AMD Genoa 9654@2.4GHz	10	1.5 TB	192	1920
Intel-g4	Intel(R) Xeon(R) Gold 6448H	2	1.0 TB	128	256

GPU Cluster	GPU Model	Nodes	GPU-Boards (each node)
GPU-A100	NVIDIA A100	2	8
GPU-V100	NVIDIA V100	5	8

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

User Interfaces (Login Nodes)

Login into Slurm User Interface

- The user interface node for slurm are :

  slurm-ui[01-03].twgrid.org （CentOS 7）

  slurm-ui04.twgrid.org （AlmaLinux 9）

- Login in user interface :

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

- 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.

Basic Usage of Slurm System

Basic Usage of Slurm System

- Query nodes and partitions information

  sinfo or sinfo -N

- Submit your job with bash script (recommended)

  sbatch your_script.sh

- Query the jobs submitted by you

  sacct or sacct -u <your account>

- Cancel your job

  scancel <your JobID>

your_script.sh

```
#!/bin/bash
#SBATCH --job-name=helloworld
#SBATCH --error=%J.err
#SBATCH --output=%J.out

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

Basic Usage of Slurm System

- Show queue information

  squeue

- Show your job in the queue

  squeue -u <your account>

- Show the detailed job information

  scontrol show job <your JobID>

- Show the detailed node information

  scontrol show node <node name>

```
[yiruchen0101@slurm-ui01 workshop]$ scontrol show job 6449154
JobId=6449154 JobName=helloworld
 UserId=yiruchen0101(5019) GroupId=ASGC(525) MCS_label=N/A
 Priority=27147 Nice=0 Account=asgc QOS=normal
 JobState=COMPLETED Reason=None Dependency=(null)
 Requeue=0 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
 RunTime=00:00:00 TimeLimit=03:00:00 TimeMin=N/A
 SubmitTime=2024-06-19T18:47:54 EligibleTime=2024-06-19T18:47:54
 AccrueTime=2024-06-19T18:47:54
 StartTime=2024-06-19T18:47:55 EndTime=2024-06-19T18:47:55 Deadline=N/A
 SuspendTime=None SecsPreSuspend=0 LastSchedEval=2024-06-19T18:47:55
 Partition=intel-g4_short AllocNode:Sid=slurm-ui01:1532
 ReqNodeList=(null) ExcNodeList=(null)
 NodeList=hpi-wn01
 BatchHost=hpi-wn01
 NumNodes=1 NumCPUs=2 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
 TRES=cpu=2,node=1,billing=2
 Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
 MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
 Features=(null) DelayBoot=00:00:00
 OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
 Command=/dicos_ui_home/yiruchen0101/workshop/test.sh
 WorkDir=/dicos_ui_home/yiruchen0101/workshop
 StdErr=/dicos_ui_home/yiruchen0101/workshop/%J.err
 StdIn=/dev/null
 StdOut=/dicos_ui_home/yiruchen0101/workshop/%J.out
 Power=
 NtasksPerTRES:0
```

Partition/ Queues of slurm

- Slurm Partitions (Queues)
- The default queue is “edr1_short”. Users could submit to different partitions by assigning partition parameters, e.g.

  sbatch -p hdr1-al9_short my_script.sh

Cluster	Partition	Timelimit	Nodelist
HDR1	hdr1-al9_short	3:00:00	sma-wn[01-02] hpa-wn[01-04]
	hdr1-al9_short_serial	4:00:00	
	hdr1-al9_large	14-00:00:00	
	hdr1-al9_long_serial	14-00:00:0	
	hdr1-al9_moderate_serial	2-00:00:0	

Cluster	Partition	Timelimit	Nodelist
Intel-g4	intel-g4-al9_short	03:00:00	hpi-wn[01-02]
	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	
EDR1	edr1_short	03:00:00	hpa-wn[05-14]
	edr1_short_serial	04:00:00	
	edr1_large	14-00:00:00	
	edr1_long_serial	14-00:00:00	
	edr1_moderate_serial	2-00:00:0	
A100	a100	5-00:00:00	hp-teslaa[01,03]
	a100_long	7-00:00:00	
	a100_short	06:00:00	
	a100_devel	20:00	
V100	v100	5-00:00:0	hp-teslav[01-05]
	v100_short	06:00:00	
	v100_long	7-00:00:00	

Quality of Service (QoS) of slurm

- 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
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

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

Basic Usage of Environment Modules

- Show available modules in slurm-ui

  module avail or ml av

- Load module

  module load <module name> or ml <module name>

- Unload module

  module unload <module name> or ml -<module name>

- Show currently loaded modules

  module list or ml

- Unload all loadedmodule

  module purge or ml purge

```
[yiruchen0101@slurm-ui04 workshop]$ module avail

------------------- /ceph/sharedfs/apps/amd/almalinux9/hdr1-al9/modulefiles -------------------
   aocc/4.2.0    gcc/13.1.0                (D)    mpi/openmpi-5.0.3/aocc420
   aocl/4.2.0    hdf5/1.8.12                      mpi/openmpi-5.0.3/gcc13.1.0 (D)
   gcc/11.5.0    mpi/mpich-4.2.2/gcc13.1.0

------------------------ /ceph/sharedfs/apps/general/centos7/modulefiles ------------------------
   R/4.0.5                  gcc/12.1.0                          mumax3/cuda-11.3/10_linux
   R/4.2.1                  git/2.37.1                          nvhpc-byo-compiler/20.11
   R/4.4.0                  golang/1.21.7                       nvhpc-byo-compiler/24.3   (D)
   R/4.4.1           (D)    gromacs/2021.swaxs-0.1/gpu          nvhpc-hpcx-cuda12/24.3
   anaconda3/4.9.2          gromacs/2022.swaxs-0.1/cpu          nvhpc-hpcx/24.3
   anaconda3/4.10.3         gromacs/2022.swaxs-0.1/gpu    (D)   nvhpc-nompi/20.11
   anaconda3/4.12.0 (D)     gromacs/2024.0/a100_gpu             nvhpc-nompi/24.3          (D)
   binutils/2.35.2          gromacs/2024.0/cpu                  nvhpc-openmpi3/24.3
   cmake/3.20.3             gromacs/2024.0/gpu            (D)   nvhpc/20.11
   cuda/11.3.0              hdf5/1.10.5-parallel                nvhpc/24.3                (D)
   cuda/11.8.0       (D)    hdf5/1.10.5                   (D)   paraview/5.8.0
   gcc/4.8.5                julia/1.8.0                         python/3.9.5
   gcc/8.2.0                make/4.3                            python/3.9.18             (D)
   gcc/8.3.0                mpi/openmpi-2.0.2/gcc485            root/v6.30
   gcc/9.3.0                mpi/openmpi-3.1.6/cuda/gcc930       singularity/4.1.2
   gcc/10.3.0               mpi/openmpi-3.1.6/gcc930
   gcc/11.1.0               mpi/openmpi-4.1.0/gcc485

------------------------- /ceph/sharedfs/apps/intel/centos7/modulefiles -------------------------
   clck/2021.6.0               debugger/2021.6.0         init_opencl/2022.1.0    mkl/2022.1.0
   compiler-rt/2022.1.0        dev-utilities/2021.6.0    inspector/2022.1.0      tbb/2021.6.0
   compiler/2022.0.2           dpl/2021.7.0              intel_mpi/2021.6.0
   compiler/2022.1.0    (D)    icc/2022.1.0              itac/2021.6.0

---------------------------- /usr/share/lmod/lmod/modulefiles/Core ----------------------------
    lmod    settarg
```

Python, Compilation and MPI Enviroment

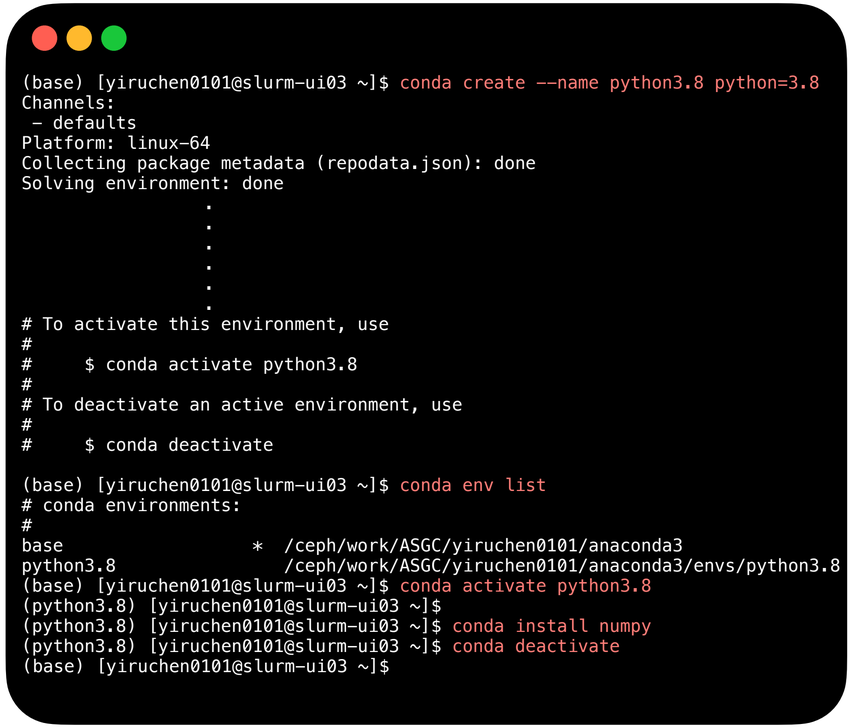
Python

- The default system python on CentOS 7 is python 2.7.5
- The default system python on Almalinux 9 is python 3.9.18
- Use python 3, please consider using anaconda with python3 first

  module load anaconda3/4.9.2

- Install additional applications /ceph/work/<groupname>/
- Before installing a special python package, use the virtual environment :

```
conda create --name <myenv>  # Create a virtual environment called myenv.
conda env list               # List the current state of the virtual environment.
conda activate <myenv>       # Starting a new virtual environment.
conda install <your_package> # Install the required packages in this virtual environment.
conda deactivate             # Leaving the Virtual Environment.
```



Compilation

- Intel compiler

  module load icc/2022.1.0

- AMD compiler

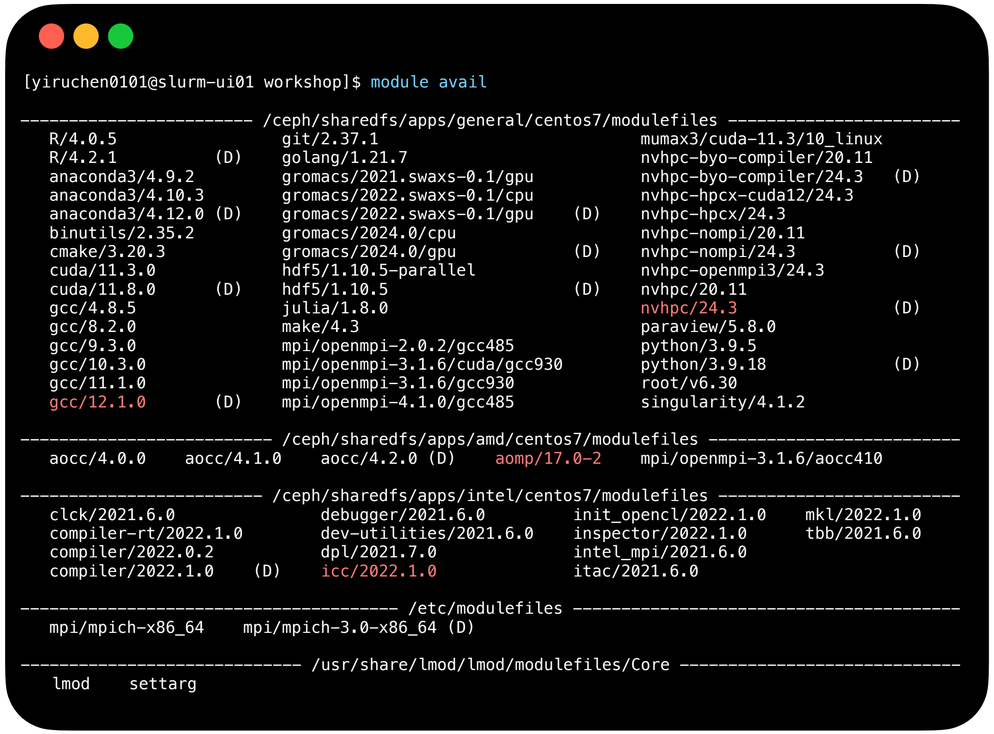
  module load aomp/17.0-2

- GCC

  module load gcc/12.1.0

- nvidia development kit (nvcc, for GPU program development)

  module load nvhpc/24.3



MPI

- Load compiler first, e.g. intel compiler

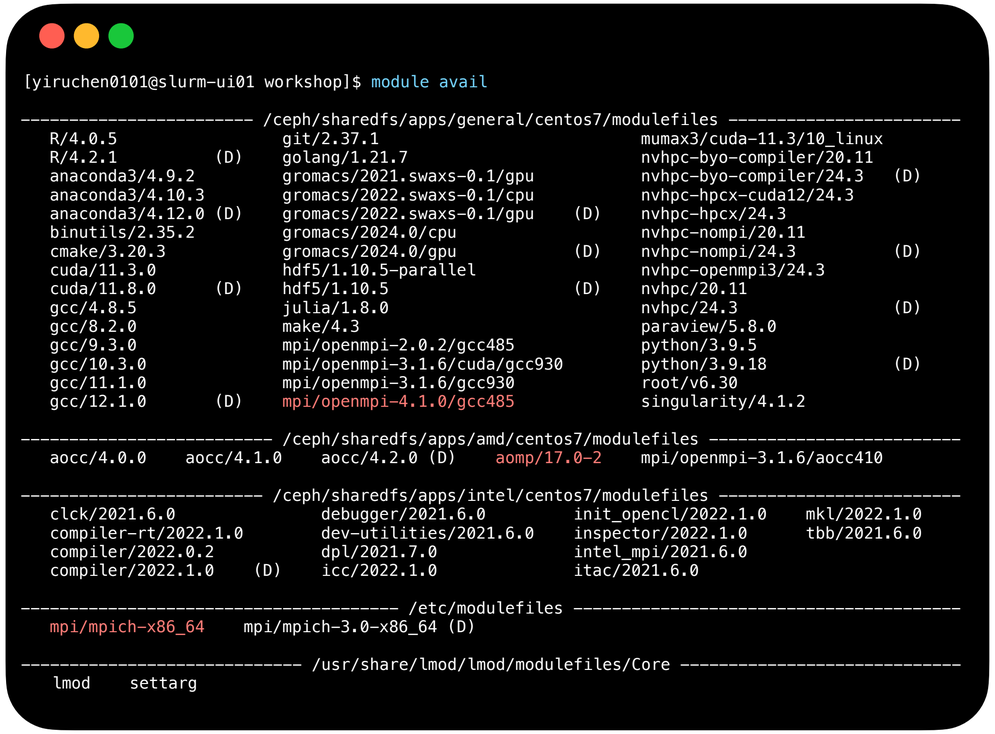
  module load icc/2022.1.0

- Load different MPI implementation
  - mpich

    module load mpi/mpich-x86_64
  - openmpi

    module load mpi/openmpi-4.1.0/gcc485
  - mvapich2

    module load aomp/17.0-2



Slurm Job Submission Examples - Hands on

Preparation

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

- Open your terminal
- Login : ssh asgctest<XX>@slurm-ui.twgrid.org
- Go to the working folder : cd /ceph/work/Others/
- Build your folder : mkdir asgctest<XX>
- Go to your folder : cd asgctest<XX>
- Copy file : cp -r ~/* .
- Go to slurm-handson : cd slurm-handson

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

hello_world.sh

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

slurm-%J.out

```
三  6月 19 22:03:40 UTC 2024
Hello DiCOS Users!
hpi-wn01.twgrid.org
```

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

mcore.sh

```
#!/bin/bash
#SBATCH --job-name=My_MCORE_Job  # define the name of your job
#SBATCH --time=01:00:00          # specify the required time
#SBATCH --nodes=1                # number of nodes allocated to the job
#SBATCH --ntasks-per-node=1      # number of tasks to invoke on each node
#SBATCH --cpus-per-task=10       # number of CPUs required per task
#SBATCH --error=job.%J.err       # job error. By default, both files are directed to a file of the name slurm-%j.err
#SBATCH --output=job.%J.out      # job output. By default, both files are directed to a file of the name slurm-%j.out

srun stress -c 10 -t 100
```

job.%J.out

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

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}")
```

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

slurm-%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.

MATLAB 

R2024a

- /ceph/sharedfs/software/hpc/MATLAB/R2024a/bin/matlab

```
#!/bin/bash
#SBATCH –job-name=matlab_job
#SBATCH –output=matlab.job.%J.out
#SBATCH –error=matlab.job.%J.err
#SBATCH –partition=intel-g4-al9_short
#SBATCH –nodes=1
#SBATCH ––ntasks-per-node=1
#SBATCH ––cpus-per-task=16

hostname
/ceph/sharedfs/software/hpc/MATLAB/R2024a/bin/matlab –nodisplay –nodesktop –nosplash < matlab_example.m
```