Lecture 0: Course Introduction && Slurm Intro

Lecturer : Atseng

AMD x NYCU SDC

GPU Programming Training Program

Course Contents

01 About this Course

02 What is Slurm

03 How to launch jobs in Slurm

04 Hands-on time !

01 About this Course

Course Information

Join the Discord First !

All the class information call discuss in Discord

https://discord.gg/wEGVXNT7Mh

Hosts

Nuss

HiPAC第一屆國網盃應用程式效能優化競賽冠軍

The 5th APAC HPC-AI competition second place

ASC世界大學生超級計算機競賽
ePrize算挑戰獎、一等獎

高速計算人工智慧冬令營擔任助教及講師

進階高效能計算叢集電腦實務助教

Isaac

ASC 世界大學生超級計算機競賽遠端組冠軍、團隊競賽獎、應用創新獎

Atseng

2024 NCHC, NVIDIA, OpenACC Hackthon

SCHEDULE

Tue

18:30

|

21:30

3/11	Week 4	Course Introduction && Slurm Intro	@atseng
3/18	Week 5	OpenMP Intro	@atseng
3/25	Week 6	Lecture 1 What's ROCm (HW 1 announcement)	@atseng
4/1	Week 7	期中週	
4/8	Week 8	期中週	
4/15	Week 9	Lecture 2 GPU basic parallel programming && HIP intro	@atseng
4/22	Week 10	AMD Invited Lecture (HW1 deadline 4/19)	
4/29	Week 11	Lecture 3 Deep into Memory	@atseng
5/6	Week 12	Lecture 4 Advanced skill in HIP	@atseng
5/13	Week 13	Lecture 5 Profiling & Optimization	@atseng
5/20	Week 14	Final Presentation	@atseng
5/27	Week 15	期末週	
6/3	Week 16	期末週	

What You Can Learn

Memory management

GPU Kernels for Parallel Execution

HIP programming model

Basic Knowledge of AMD GPU Architecture (RDNA 3)

Synchronize CPU and GPU Workloads

Profiling & Optimization Skills, etc.

02 What is Slurm

Slurm Intro

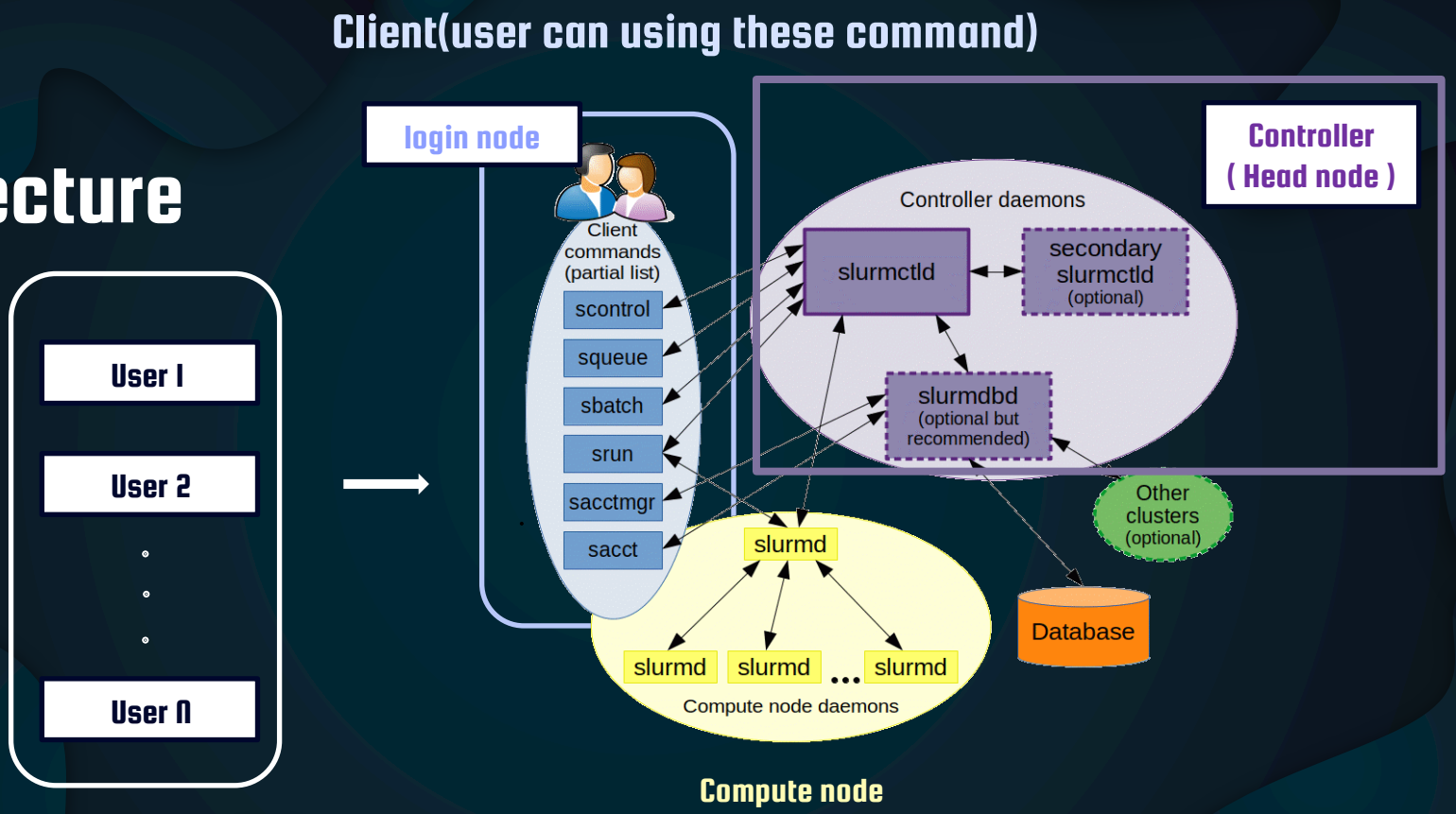
Slurm = Simple Linux Utility for Resource Manager

An Open-source resource management and job scheduling system

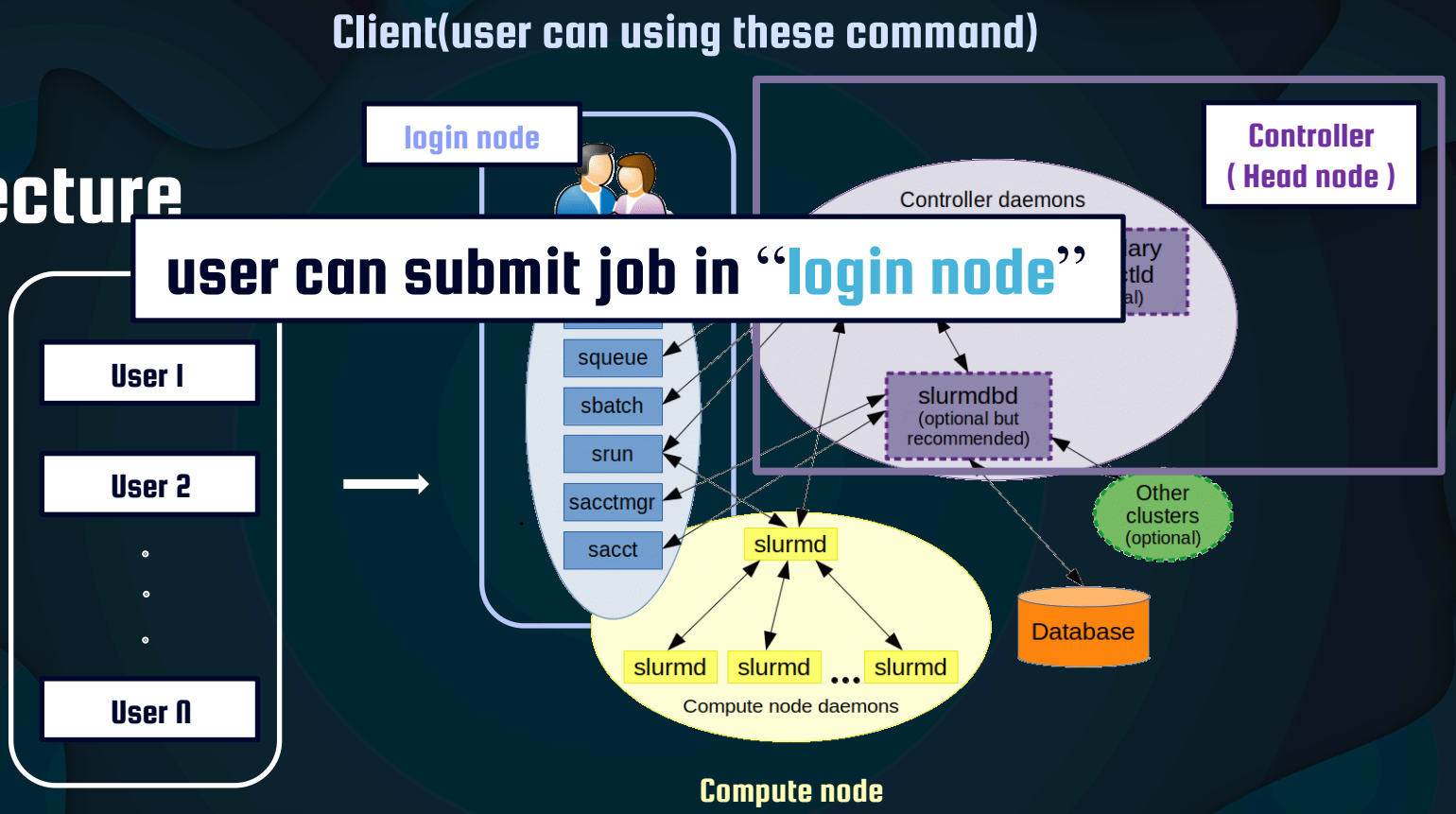
Slurm's Mission

- Resource Management: Allocates designated computing resources to users.
- Job Scheduling: Allows resources to be allocated based on priority.

Slurm Architecture

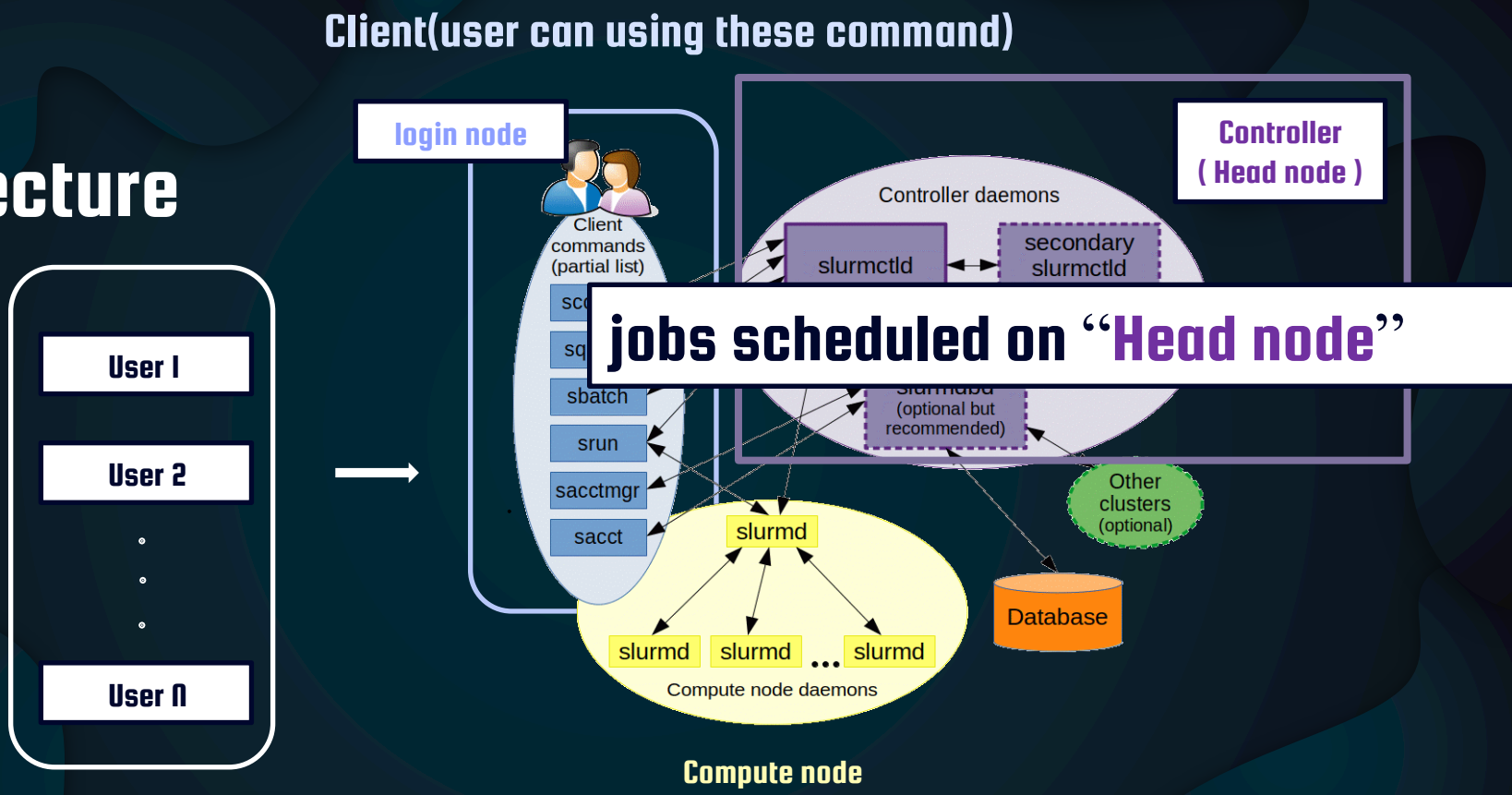


Slurm Architecture



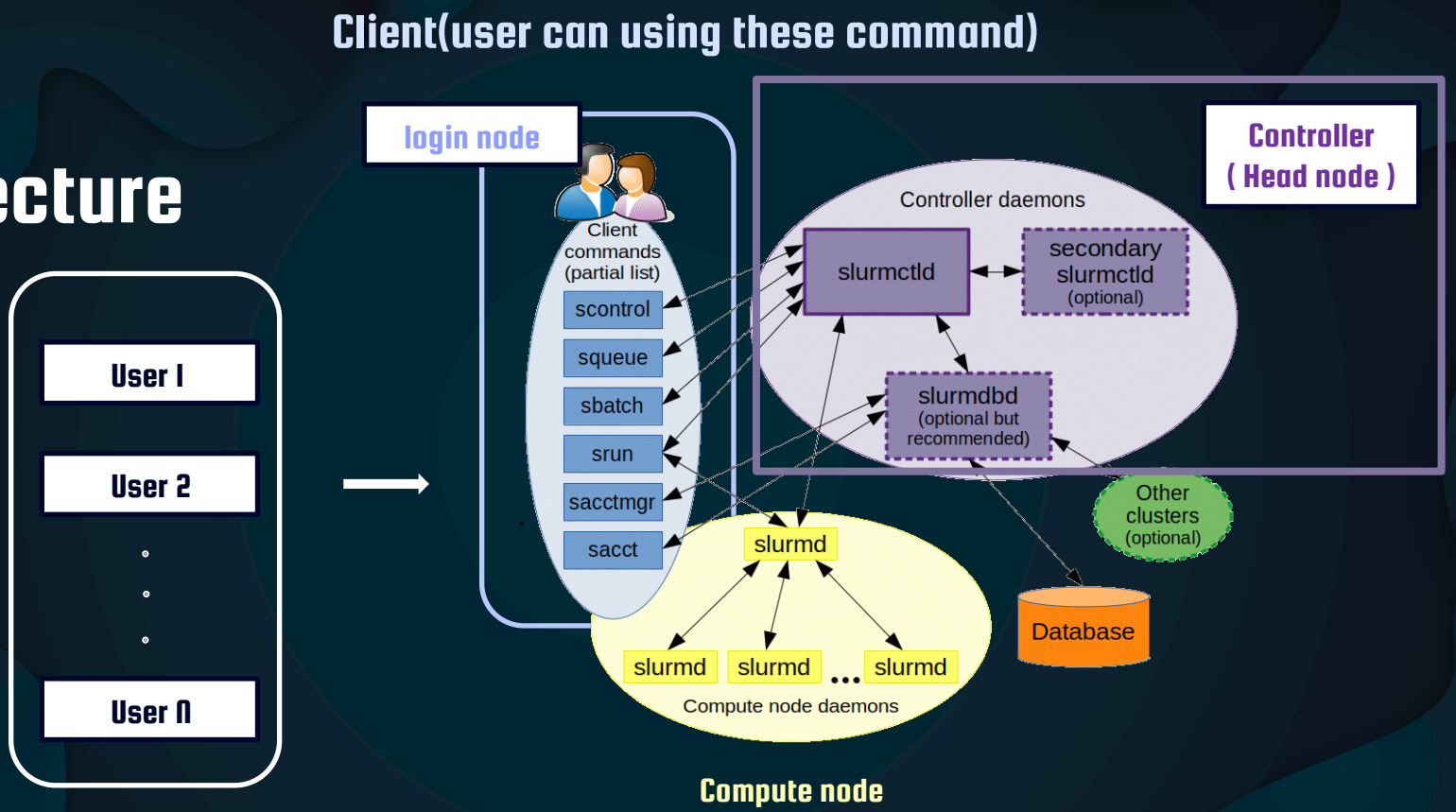
user can submit job in “login node”

Slurm Architecture



jobs scheduled on "Head node"

Slurm Architecture



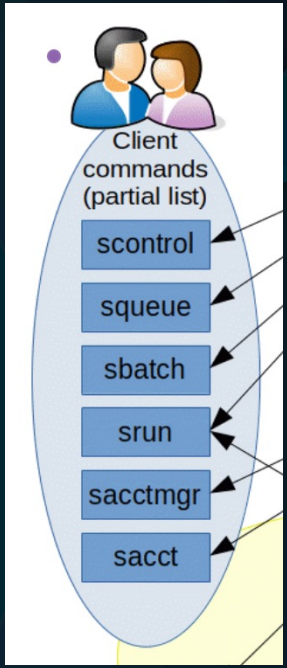
jobs excuted on "Compute node(s)"

Login Node

- Provides users with an entry point to log in to the cluster (via SSH)
- Job Submission (sbatch, salloc, and srun)

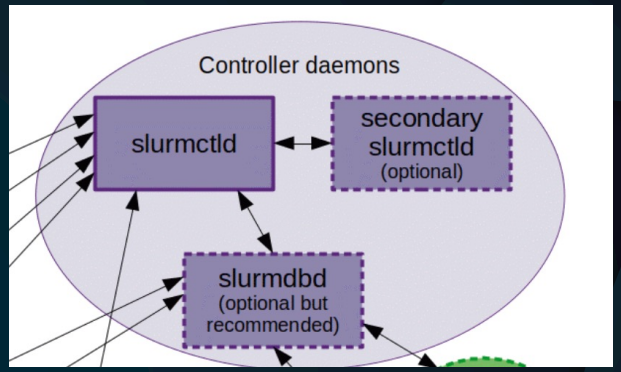
Login node 負責發 job

- Provides basic resource queries (sinfo...)



!!! Please Don't run your code in Login Node !!!

Controller Node (head node) (master node)



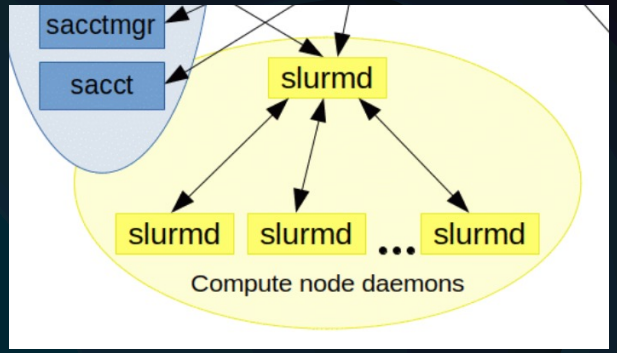
- Serves as the central management hub of the cluster.

- Handling job scheduling, resource allocation.

- Maintains communication with compute nodes, monitoring their health, job execution status, and resource usage.

- Handles node failures, job errors, and recovery events to ensure the stability and smooth operation of the cluster.

Compute Node



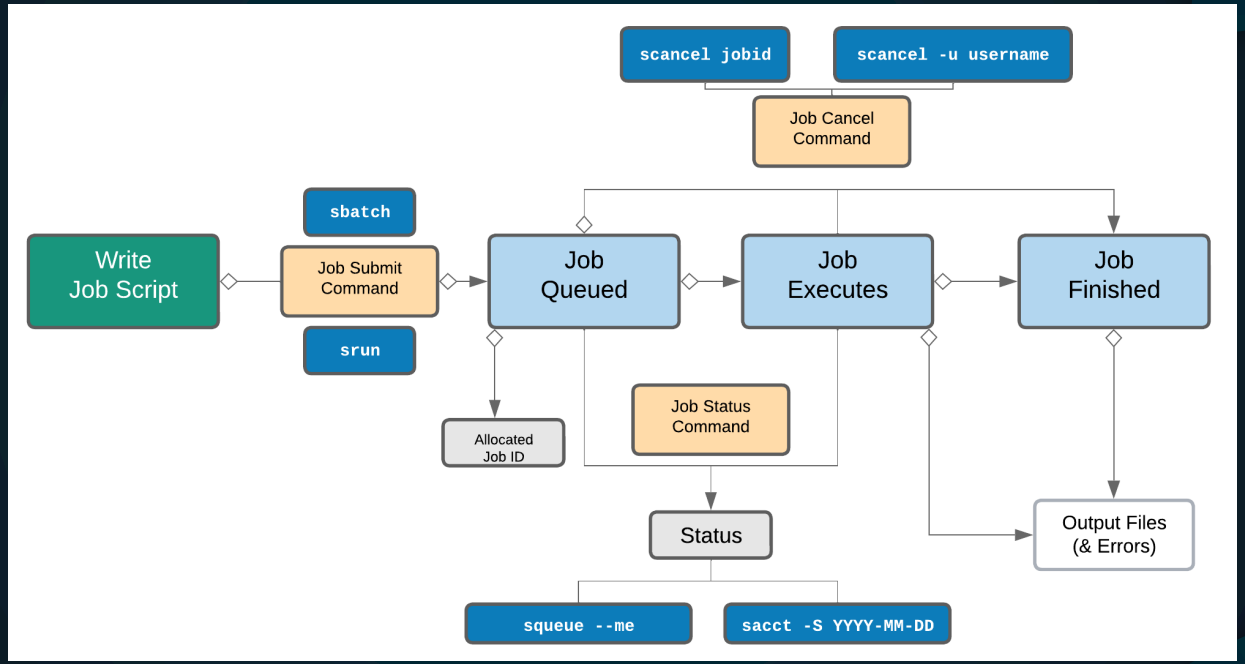
- The actual compute resources for executing jobs .

- Runs the Slurm execution daemon (slurmd), which accepts job tasks from the controller node and manages the job's runtime execution.

- Executes job steps as directed by the controller node, allocating CPU, memory, disk, and other resources as required.

- Sends job completion notifications or error messages to the controller node once a job finishes or encounters issues, aiding in monitoring and logging.

Life Cycle of a Job



https://genomicsaotearoa.github.io/Workshop-Bash_Scripting_And_HPC_Job_Scheduler/5_working_with_job_scheduler/

The things that you should know !

Job

- The unit submitted by the user; the unit for which Slurm applies for and allocates resources.
- Represents the work to be executed on compute nodes.

Job Step

- A job can include one or more job steps.
- Typically consists of job commands. (sinfo . . .)

Task

- An instance of a job execution, serving as the basic unit that performs the actual computation.
- A job step can include one or more tasks.
- Usually corresponds to a process. (one task ~= one process run that task)

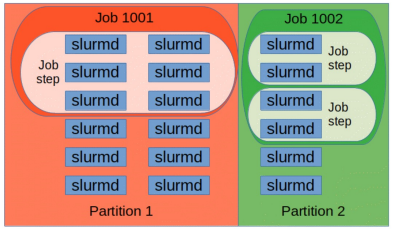


Figure 2. Slurm entities

The things that you should know !

Job

- The unit submitted by the user; the unit for which
- Represents the work to be executed on compute n

Job Step

- A job can include one or more job steps.
- Typically consists of job commands. (sinfo ...)

Task

- An instance of a job execution, serving as the basi
- A job step can include one or more tasks.
- Usually corresponds to a process. (one task ~=

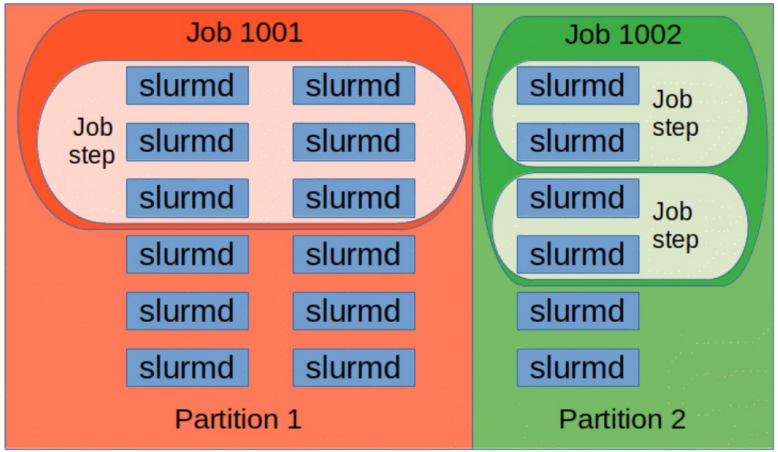


Figure 2. Slurm entities

The things that you should know !

Node

- A compute node is a node where slurmd is installed and correctly configured.
- A node with slurmctld installed is known as the Slurm Master node (Head node) .

Partition/Queue

- A collection of multiple nodes grouped based on hardware configuration, job runtime, job type, user permissions, etc.
- If compute nodes have different hardware configurations, partitions help users select the appropriate node.
- Each partition has its unique configuration and restrictions, such as maximum runtime, user access limits, or node count limits.

The things that you should know !

Workflow

Request resources from Slurm. -> Specify resource requirements using parameters.

-> Execute the job -> Run the job within the allocated environment.

Common commands (monitor job)

- sinfo : Displays the existing nodes and partitions available in the cluster.

- squeue : Shows all jobs under Slurm management that are either waiting in the queue or currently running.

- scontrol : A management tool used to view and modify Slurm configurations; most commands require root privileges.

- sacct : Lists the current state of jobs related to the active account, such as Pending, Running, Completed, Failed, etc.

- sstat : Monitors the status of a running job, including details like CPU usage, physical memory consumption , and virtual memory (VM)

Common commands (Launch jobs)

- salloc : Allocates resources for interactive jobs and opens a new shell.
  When you exit the shell, the job terminates automatically (be sure to exit!).
  (Suitable for development and debugging of computational programs that require frequent testing.)
- srun : Directly executes tasks under the Slurm management environment and waits for the command to complete
  Can be used with both sbatch (non-interactive) and salloc (interactive)
  (Suitable for debugging tasks that require real-time output.)
- sbatch : Submits a job in batch mode by specifying resources and commands in a job script.
  (Ideal for multi-node computational tasks)
- scancel : Used to cancel one or more running jobs.
  Specify the job ID to cancel, or use `-u $USER` to remove all jobs belonging to the current user.

Parameters

-A, --account : Specify the job's account (wallet) ID.

-p, --partition : Specify the queue (partition) to which the job will be submitted.

-J, --job-name : Set the job's name.

-N, --nodes : Specify the number of nodes required for the job. partition 內的幾個 node

-n, --ntasks : Specify the total number of tasks for the job.

-c, --cpus-per-task : Specify the number of CPUs required per task.

-o, --output : Specify the path to the output file.

-e, --error : Specify the path to the error output file.

Parameters

--mem : Specify the memory requirement for each node.

--gres : Request generic resources, such as GPUs or Infiniband.

--gpus-per-node : Specify the number of GPUs required per node.

--pty : Run the command in terminal mode.

-t, --time : Set the maximum runtime for the job.

--exclusive : Request exclusive access to nodes, ensuring that resources are not shared with other jobs.

--constraint : Request compute nodes with a specific hardware configuration.

03 How to launch jobs in Slurm

Non-interactive Jobs (sbatch)

1.Create a file called { xxxxxxx.sh }

2. Using `sbatch hello_world.sh` to submit job.

```
#!/bin/bash
#SBATCH -A my_account # -A, --account : Specify the job's account (wallet) ID
#SBATCH -p my_partition # -p, --partition : Specify the job queue (partition)
#SBATCH -J HelloWorld # -J, --job-name : Set the job name
#SBATCH -N 1 # -N, --nodes : Specify the number of nodes (1 node)
#SBATCH -n 4 # -n, --ntasks : Specify the total number of tasks (4 tasks)
#SBATCH -c 1 # -c, --cpus-per-task : Specify the number of CPUs per task (1 CPU per task)
#SBATCH -o hello_world.out # -o, --output : Specify the output file path
#SBATCH -e hello_world.err # -e, --error : Specify the error output file path

# Simple operation:
echo "Hello world" to a file named hello_output.txt echo "Hello world" > hello_output.txt
```

Interactive Jobs (srun, salloc)

– Use salloc or srun to launch jobs through the terminal.

Interactive Session with Two Separate Commands

a. Request an interactive allocation using salloc:

```
salloc -A my_account -p my_partition -N 1 -n 4 -c 1
```

requests an allocation of 1 node with 4 tasks (each with 1 CPU) under the specified account and partition. Once granted, you'll get an interactive shell on the allocated node(s).

b. Run the command using srun.

```
srun echo "Hello world" > hello_output.txt
```

This runs the command echo "Hello world" on the allocated resources and redirects the output to a file named hello_output.txt

Summary

How to submit a Interactive job ?

- salloc { parameters }
- srun { parameters } < your program >

How to submit a batch job ?

- write a job script (xxxxx.sh)
- sbatch { parameters } < your job script >

04 Hands-on time !

It’s your turn!

SSH to the login node

```
ssh {user}@{hostname}
```

Course Reference

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

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

https://genomicsaotearoa.github.io/Workshop-Bash_Scripting_And_HPC_Job_Scheduler/5_working_with_job_scheduler/

https://jhpce.jhu.edu/slurm/about-jobs/