ilifu Online Training – Introduction to `slurm`

Dane Kennedy

Bioinformatics Support, ilifu
University of Cape Town, 3 September 2024

Job scheduling & cluster management tool

- Framework : Login node & Compute nodes

- **Login node**
  - Accessed via ssh (`$ ssh <username>@slurm.ilifu.ac.za`)
  - Submit jobs and manage work directories

- **Compute nodes**
  - Where processes/code runs
  - via singularity containers or modules

- **Partitions / Queues**

`Main, Jupyter, Devel` 32 core, ~232GiB RAM	`GPU` 32 core, ~232GiB RAM, Nvidia GPUs	`HighMem` 32 core, 503GiB RAM 96 core, 1.5TiB RAM
85 + 12 Nodes	7 nodes	3 nodes

SLURM - Use cases

Login node
Run SLURM & bash commands
cd, mkdir, ls, etc

Jupyter/Dev. node
Development space
New code / workflows / routines
Debugging / testing software

Main partition
Stable, computationally
heavy processing

HighMem/GPU
For single-high memory
jobs that can't be split
into multiple jobs for MPI

SLURM - user commands

```
$ sinfo                      → shows partitions and resources

$ squeue                     → shows all jobs in SLURM queue/partition

$ squeue -u $USER            → shows your jobs

$ sbatch slurm_job_script.sh → submit job to SLURM

$ sbatch --help              → information on job submission parameters

$ scancel <jobid>            → cancel running / pending job
```

SLURM - running a job

```
#!/bin/bash

module add python/3.11.2

python hello_world.py
```

SLURM - defaults

```
–time=0-03:00:00   # 3 hours
–mem=3G            # 3 GiB
–ntasks=1          # one task
–nodes=1           # one node
–partition=Main
–account=<your default>
```

SLURM - running a job

https://github.com/ilifu/ilifu_user_training/tree/main/introduction/tutorial2

```
#!/bin/bash
#SBATCH --job-name=tutorial2_R_container
#SBATCH --time=00-00:01:00
#SBATCH --mem=4G
#SBATCH --partition=Main
#SBATCH --output=R_container-%j.stdout
#SBATCH --error=R_container-%j.stderr
#SBATCH --mail-user=YOUR_EMAIL_ADDRESS
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT_80
#SBATCH --account=ACCOUNTING_GROUP
```

Describe **job** parameters / resources

```
singularity exec /software/common/containers/RStudio2023.06.1-524-R4.3.1.sif Rscript hello_world.R
```

container

software

SLURM - defaults and maximums per partition

https://docs.ilifu.ac.za/#/tech_docs/resource_allocation?id=maximum-allocation

Partition	Node names	Default CPUs	Max CPUs	Default Memory (GiB)	Max Memory (GiB)	Default wall-time	Max wall-time
Main	compute-[002-021]	1	32	3	232	3 hours	14 days
Main	compute-[101-105]	1	48	3	232	3 hours	14 days
Main	compute-[201-260]	1	32	3	251	3 hours	14 days
HighMem	highmem-[001-002]	1	32	15	503	3 hours	14 days
HighMem	highmem-003	1	96	15	1508	3 hours	14 days
GPU	gpu-[001-004]	1	32	7	232	3 hours	14 days
GPU	gpu-005	1	24	7	232	3 hours	14 days
GPU	gpu-006	1	48	7	354	3 hours	14 days
GPU	gpu-007	1	48	7	354	3 hours	14 days
Devel	compute-001	1	32	-	-	3 hours	12 hours

DEMO TIME!

https://github.com/ilifu/ilifu_user_training/tree/main/introduction/tutorial2

SLURM - Best practices

Do's :
- Run jobs using sbatch rather than interactive jobs
- Identify job resources requirements:
  - No. of nodes and CPUs, amount of RAM and wall-time.
- Remove files that aren't needed
  - `/scratch3` folder after data processing is complete
  - Old raw data, temporary products , etc.
- Use Singularity (cannot install software on nodes)
- Use <`username`>@`transfer.ilifu.ac.za` for data transfers

Don't:
- Don't run software/heavy processes on login node
- Don't place large files in your home directory (`/users`)
- Don't transfer using `scp`/`rsync` on the login node
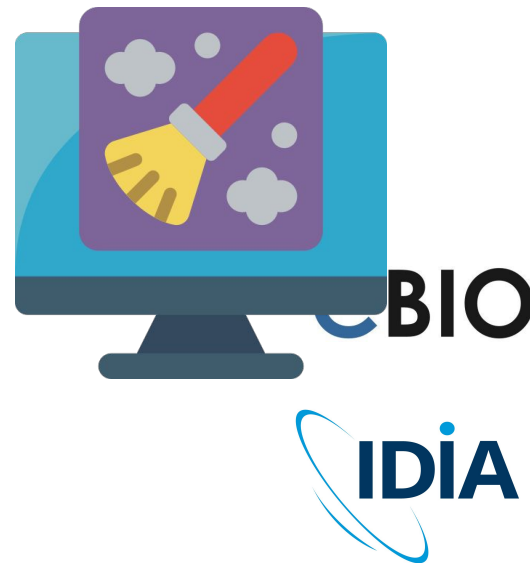
Thank you
for coming and for your time.

https://docs.ilifu.ac.za/

support@ilifu.ac.za