

Understanding the Power of Supercomputers

Environmental Modules, SLURM Job Submission, and Lustre FS Commands

An Introduction for Beginners

Dr A.Khalatyan

Researcher at eScience/Supercomputing/IT

Leibniz-Institut für Astrophysik Potsdam (AIP), Germany

AIP, 2025

Environmental Modules

What are Environmental Modules?

- **Purpose:** Manage and configure the user environment dynamically.
- **Functionality:** Load, unload, and switch between different software packages and their dependencies.

Module Command Overview

Basic Commands

- Load a Module

```
module load <module_name>
```

- Unload a Module

```
module unload <module_name>
```

Module Command Overview

Basic Commands

- List Loaded Modules

```
module list
```

- Search for Available Modules

```
module avail
```

- Show Module Information

```
module show <module_name>
```

Example Usage

Loading and Unloading

1. Load Python Module

```
module load gcc
```

2. Verify Loaded Modules

```
module list
```

3. Unload Python Module

```
module unload gcc
```

Try it yourself

ssh **username@141.33.4.143**

- check what modules are loaded in your terminal on the **Newton21**
- load Intel compiler
- What error do you see?
- Follow instructions

SLURM Job Submission

Introduction to SLURM

- What is SLURM?
 - Simple Linux Utility for Resource Management.
 - Manages job scheduling and resource allocation on HPC clusters.

Submitting Jobs with SLURM

sbatch Command

- Basic Submission

```
sbatch <script.sh>
```

- Example Script (`script.sh`)

```
#!/bin/bash
#SBATCH --job-name=my_job
#SBATCH --output=output.txt
#SBATCH --time=01:00:00
#SBATCH --partition=debug

module load gcc
./my.exec
```


SLURM Script Directives

Common Directives

- `#SBATCH --job-name=<name>`
- `#SBATCH --output=<file>`
- `#SBATCH --error=<file>`
- `#SBATCH --time=<HH:MM:SS>`
- `#SBATCH --partition=<partition>`

SLURM Script Directives

Common Directives

- `#SBATCH --nodes=<number>`
- `#SBATCH --ntasks=<number>`
- `#SBATCH --mem=<memory>`
- `#SBATCH --mail-type=<type>`
- `#SBATCH --mail-user=<email>`

Detailed Explanation of SBATCH Directives

#SBATCH --time=00:01:00

- **Purpose:** Specifies the maximum run time for the job.
- **Format:** HH:MM:SS
- **Example:**

```
#SBATCH --time=00:01:00
```

- **Meaning:** The job is allowed to run for 1 minute.

#SBATCH --job-name=<name>

- **Purpose:** Assigns a name to the job for easy identification.
- **Example:**

```
#SBATCH --job-name=my_analysis
```

#SBATCH --output=<file>

- **Purpose:** Redirects the standard output (stdout) of the job to a specified file.
- **Example:**

```
#SBATCH --output=results.out
```

#SBATCH --error=<file>

- **Purpose:** Redirects the standard error (stderr) of the job to a specified file.
- **Example:**

```
#SBATCH --error=errors.err
```

#SBATCH --partition=<partition>

- **Purpose:** Specifies the partition (queue) to submit the job to.
- **Example:**

```
#SBATCH --partition=compute
```

#SBATCH --nodes=<number>

- **Purpose:** Requests a specific number of nodes for the job.
- **Example:**

```
#SBATCH --nodes=2
```


#SBATCH --ntasks=<number>

- **Purpose:** Specifies the number of tasks/processes to run.
- **Example:**

```
#SBATCH --ntasks=16
```

#SBATCH --mem=<memory>

- **Purpose:** Requests a certain amount of memory per node.
- **Format:** Can specify in MB (e.g., 500M), GB (e.g., 2G), etc.
- **Example:**

```
#SBATCH --mem=8G
```

#SBATCH --mail-type=<type>

- **Purpose:** Specifies when to send email notifications.
- **Common Types:**
 - **BEGIN** : When the job starts.
 - **END** : When the job finishes.
 - **FAIL** : If the job fails.
 - **ALL** : All of the above.
- **Example:**

```
#SBATCH --mail-type=END,FAIL
```

#SBATCH --mail-user=<email>

- **Purpose:** Specifies the email address to send notifications to.
- **Example:**

```
#SBATCH --mail-user=user@example.com
```

AIP Example for hybrid jobs

```
#!/bin/bash
#SBATCH --partition=tiny
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=96
#SBATCH --cpus-per-task=1
#SBATCH --time=00:01:00
module load intel
module swap hdf5 phdf5
module load gsl
module load mkl
module load hypre
module load fftw
module list
sleep 1
prun test-hybrid.x > a.log
```

Managing Jobs

Checking Job Status

- **queue Command**

```
queue -u <username>
```

Canceling a Job

- **scancel Command**

```
scancel <job_id>
```

Filesystems

- Home for source and important small data:
 - spacial with snapshotting every hour
- Lustre massively parallel filesystem
 - for large data
 - backup only on demand

Lustre Filesystem Commands

Introduction to Lustre FS

- **What is Lustre?**
 - High-performance parallel file system commonly used in HPC environments.
- **Key Features**
 - Scalability, reliability, and high throughput.

Ifs Command Overview

- General Usage

```
ifs <subcommand> [options]
```

Common Subcommands

- `ifs find` : Search for files in Lustre filesystem.
- `ifs setstripe` : Configure stripe settings on directories/files.
- `ifs getstripe` : Retrieve current stripe configuration.

Using `lfs find`

Syntax

```
lfs find <path> [options]
```

Example

- Find All `.dat` Files

```
lfs find /lustre/project -name "*.dat"
```

Managing Stripes with `lfs`

Setting Stripe Parameters

- Set Stripe Count and Size

```
lfs setstripe -c <count> -S <size> <path>
```

- Example

```
lfs setstripe -c 4 -S 1M /lustre/project/data
```

Getting Stripe Information

- Retrieve Stripe Settings

```
lfs getstripe <path>
```

AIP Filesystem

Quota

- myquota command

```
(dask2) [arm2arm@nnewl3 ~]$ myquota
::: HOME STORAGE :::
Disk quotas for user arm2arm (uid 1266):
  Filesystem    space    quota    limit    grace    files    quota    limit    grace
192.168.111.201:/xhome
                  33158M  51200M  56320M          377k         0         0
::: DATA STORAGE :::
Disk quotas for user arm2arm (uid 1266):
  Filesystem    used    quota    limit    grace    files    quota    limit    grace
  /lustre  58.77T*  46.57T  51.22T      -  8223656  10000000  10010000  -
(dask2) [arm2arm@nnewl3 ~]$
```

Try yourself

ssh **username@141.33.4.143**

- mkdir /lustre/yourusername/excosm25
- cd /lustre/yourusername/excosm25
- create a "run.bat" file with basic jobscript
- qsub run.bat
- ls and explore folder
- run myquota

Summary

- **Environmental Modules:** Efficiently manage software environments.
- **SLURM:** Submit and manage computational jobs effectively.
- **Lustre FS:** Utilize high-performance filesystem commands for optimal storage management.

Questions?