

Regulus HPC cluster – User Guide

1. Introduction

The Regulus HPC cluster is managed using the Slurm Workload Manager scheduler (version slurm-wlm 23.11.4). It consists of 48 compute nodes, all running **Ubuntu Server 24.04 LTS**, with heterogeneous CPU slots and memory sizes.

You can use **Ganglia** to monitor the current load, memory usage, and cluster status:

<http://regulus.uibk.ac.at/ganglia>

Content of this guide

1. Introduction
 2. Getting an account
 3. Accessing the cluster
 4. Running programs
 5. Organization and storage
 6. Best practices
 7. Managing Slurm jobs
 8. Support
-

2. Getting an account

Accounts on the Regulus HPC cluster can be created for ITP students and researchers. To request an account:

1. Find the official user authorization (“Benutzungsbewilligung”) that was sent to you by ZID, which provides the license to use the ZID services along with registration data (including username, initial password, and email address).
2. Share a copy of this document with the system admin.

User authentication is managed via the Network Information Service (NIS) hosted on Mungo and the Regulus master node. The initial password is printed on your “Benutzungsbewilligung”.

To change your password: (1) log into the central server (node name: Mungo; IP: 138.232.67.8) and run `yppasswd`, (2) insert your current password, then (3) insert your new password (twice, with no visual feedback). Password updates automatically propagate to all nodes and do not affect other ZID services.

3. Accessing the cluster

3.1 Login node

The Regulus login/master node is intended only for interactive work such as editing scripts and submitting jobs. Do not run heavy computations directly on the login node. All production workloads must be submitted via the batch system using the commands described in Sections 4 and 7 (to ensure fair resource sharing and stable operation).

3.2 Connecting to the Regulus cluster

IP address: 138.232.67.20

For security reasons, the Regulus HPC cluster is accessible only from UIBK's IP addresses via SSH. Access from other networks requires a VPN such as Cisco Secure Client (see [ZID's guide](#)).

3.2.1 Connecting using a VPN

Cisco Secure Client is easy to set up and use, as described in the above guide. You can download a free version from UIBK's software download area. Note that the VPN client routes traffic through the UIBK network, so you will appear to be connecting from UIBK and may experience performance degradation with external services.

3.2.2 Connecting via SSH

Both Linux/macOS and Windows users can access the Regulus HPC cluster via SSH.

- **Connecting from Linux/macOS**

In Linux, run this command in your local shell: `ssh username@regulus.uibk.ac.at`

- **Connecting from Windows**

You need a terminal emulator that supports the SSH protocol. We recommend installing PuTTY from UIBK's free software download area.

To connect to the cluster, (1) open PuTTY's graphical interface, (2) enter the IP or `regulus.uibk.ac.at` into the "Host Name" field, (3) click "Open" to get access to a new Terminal window where (4) you are asked to enter your username and password.

4. Running programs

4.1 Software modules

Software on the Regulus HPC cluster is managed using **Environment Modules** in the two-level structure `<software>/<version>`. Software is periodically updated (up to three versions per software are available). New software can be installed upon request (see Sec. 8).

Below is a list of commands to navigate the environment modules:

Task	Command
List all available modules	<code>module avail</code>
See what a module does to your env	<code>module display <software>/<version></code>
Load a module	<code>module load <software>/<version></code>
Unload a module	<code>module unload <software>/<version></code>
List all currently loaded module	<code>module list</code>
Unload all modules at once	<code>module purge</code>

4.2 Submitting jobs

As noted in Sec. 3.1, the Regulus login/master node is intended only for interactive work such as editing job scripts, compiling code, and submitting and monitoring jobs. Do not run heavy computations directly on the login node. All production workloads must be managed via the batch system (see Sec. 7) to ensure fair resource sharing and stable operation.

4.2.1 Basic tips

- Prefer **job scripts with directives** instead of command-line submissions.
- Always request **realistic resources** (time, CPUs, memory).
Over-requesting delays scheduling.
- Keep job scripts self-contained and reproducible.

4.2.2 Basic job script example

Create a script (e.g. `job.sh`) with Slurm directives (marked by `#SBATCH`) at the top:

```
#!/bin/bash -l
#SBATCH -J <descriptive job name>
#SBATCH -t 4-00:00:00
#SBATCH --mail-type=END,FAIL
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=30
#SBATCH --mem-per-cpu=1G
#SBATCH -w node09
#SBATCH -o /home/<username>/test_%j.out
#SBATCH -e /home/<username>/test_%j.err

module load anaconda3/2024.10
python3 $FILE {parameters}
```

- `-J`
Job name shown in the queue.
- `-t`
Walltime limit (days-hours:minutes:seconds) when jobs are terminated.
- `--ntasks`
Number of tasks (MPI ranks). Slurm will automatically launch the MPI runtime.
- `--cpus-per-task`
Number of CPU cores per task (threads).
- `--mem-per-cpu`
Memory allocated per CPU core.
- `-o, -e`
Standard output and error files for debug. %j expands to the job ID.
- `--mail-type`
Email notifications.
- `--exclusive`
Prevents other jobs from sharing the same node. Use it only *if necessary*, as it reduces cluster throughput and may entail longer queue times.
- `-w`
You may request specific nodes *if necessary* (only for debugging or benchmarking).

Jobs are submitted through a batch system using the *sbatch* command. Once submitted, Slurm assigns to them a job ID, which can be used for monitoring and troubleshooting. As of 2025, there is currently only **one queue** for all users and jobs. Additional queues may be introduced depending on usage and documented here.

4.2.3 Requesting resources from the job script

You can choose the method that best matches your job's requirements.

- **CPUs** can be requested using:
 - `--ntasks`
 - `--cpus-per-task`
 - **Memory** can be requested using:
 - `--mem`
 - `--mem-per-cpu`
-

5. Organization and storage

Files and directories in the Regulus HPC Cluster are shared (from the central server Regulus) across all compute nodes. As of 2025, files in `/home/` are *not* externally backed up.

Filesystem overview

/home – All home directories `/home/<username>` on Regulus are shared with each compute node via [NFS](#). For large or temporary files, please use the `/scratch` folder described below.

/scratch – Shared scratch folder for temporary storage. `/scratch` resides on Regulus and is available on all nodes at the same path `/scratch`. It provides a large, centralized location for storing and accessing data across the cluster.

A summary of the above filesystems is shown below:

Mount point	Purpose	Backup policy (as of 2025)
<code>/home</code>	Login directory	No automatic backup
<code>/scratch</code>	Shared storage	No automatic backup

6. Best practices

- Always test new workflows with small jobs.
- Avoid hard-coding node names (`-w`) unless strictly necessary.
- Clean up large or temporary output files regularly.
- Monitor your jobs and cancel those no longer needed.
- If a job is **pending**, check:
 - the requested resources (time, memory, CPUs)
 - the cluster status and load via Ganglia
 - the job details
- If a job **fails immediately**, check:
 - the error file (`.err`)
 - module availability
 - memory and time limits
- If a job is **killed unexpectedly**, check:
 - the cluster status and load via Ganglia
 - if the job exceeded memory or walltime limits

7. Managing Slurm jobs

Task	Command
Submit a job	<i>sbatch job.sh</i>
View current job queue	<i>squeue</i>
See your jobs only	<i>squeue -u <username></i>
Detailed job info while pending	<i>scontrol show job <job id></i>
Cancel a job	<i>scancel <job id></i>
Cancel all your jobs	<i>scancel -u <username></i>
Put a job on hold	<i>scontrol hold <job id></i>
Release a held job	<i>scontrol release <job id></i>
Show finished job resource usage	<i>sacct -j <job id> --format=...</i>

8. Support

Getting started with Linux – Internet is full of great resources for Linux, but it's easy to get lost in the process! Tools are also rapidly changing and books are being complemented by platforms and chatbots. If you look for a good place to start, don't hesitate to ask around or discuss it with the [system admin](#) (ITP, 3rd floor, office 4S14).

For issues with Slurm jobs, please contact the [system admin](#) and include (at least)

- Job ID
- Job script (path and/or snippets)
- Output and error files (if applicable)

To report issues, request **software updates**, or for general enquires, please also contact the [system admin](#) or just stop by at the office above.