Onboarding
Julia on HLRS Cluster/Laptops

September 10, 2024, HLRS

HLRS Laptops

Most of the course can be completed on the laptop.

- Equipped with NVIDIA GPU
- Jupyter + VS Code
  - `jupyter lab`
  - `code`
- Course materials
  - `$HOME/JuliaHLRS24`

HLRS Training Cluster

The training cluster has two kinds of compute nodes.

- CPU-only nodes “skl”
  - 2x Intel Skylake
  - 40 cores total
- GPU nodes “clx-ai”
  - 2x Intel Cascade Lake
  - 36 cores total
  - 8x **NVIDIA V100**

HLRS uses the PBS Pro scheduler.

- Submit a job:
  - `qsub job_script.sh`
- See your queued/running jobs:
  - `qstat -rnw`

VS Code → HLRS Cluster

Run VS Code on a cluster node via SSH

Login node

- Works fine, just connect to (more later)
  - `accountname@training.hlrs.de`

Compute node

- Possible but inconvenient
  - `SetEnv PBS_JOBID=2297805.hawk-pbs5`
  - `SSH ProxyJump`

Side comment: “remote tunnels” instead of SSH

Not at HLRS!

On the target node

- Download the code CLI and run
  - `code tunnel --verbose`

Locally

- Remote Tunnels: Connect to Tunnel

Julia on the Cluster

Use a system module or standard Julia binaries.

- No need to compile Julia from source
- Binaries from
  - juliaup or julialang.org
- System module on training.hlrs.de
  - `module use julia`

Put the Julia depot on the parallel file system.

- Why not `$HOME`?
  - Quotas
  - Not always writable from compute jobs
- On HLRS clusters use Workspaces
  - `ws_allocate jlhpc 10`
- JULIA_DEPOT_PATH environment variable

Use a Julia wrapper for the Julia VS Code extension.

- Julia: Executable Path should point to a wrapper script, like this one:

```
#!/bin/bash
[...]

# Make julia available
module load julia

# Pass on all arguments to julia
exec julia "${@}"
```

Let’s try it!