HPC at Monash Robotics

A Practical Guide to High-Performance Computing

M3 (formerly known as MASSIVE)

MAVERIC (Monash AdVanced Environment for Research and Intelligent Computing)

Monash Robotics Group Meeting · 30 April 2026
Lingheng Meng | CSIRO Data61 / Monash University

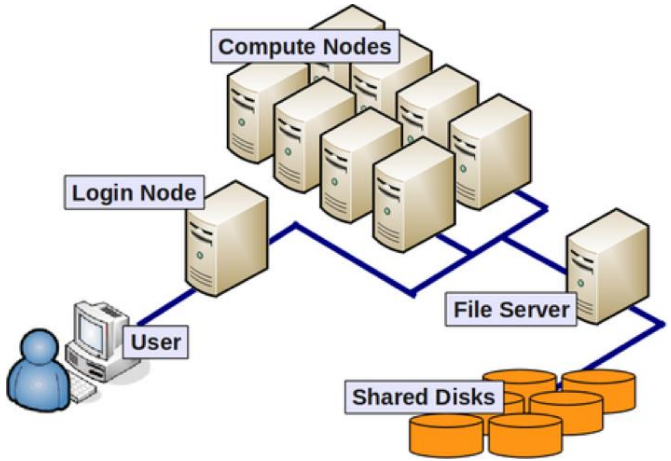
Agenda

- **Introduction**
  - What is HPC? · Why use it?
  - Computing resources overview
- **Monash Resources**
  - Monash M3 (MASSIVE)
  - Monash Robotics Kubernetes Cluster
- **National HPCs**
  - NCI Gadi · Pawsey Setonix
- **HPC Skills**
  - SLURM job scheduling
  - Apptainer containers
  - IsaacSim / IsaacLab on HPC
- **Tips & Tricks**
  - AI coding tools
  - Job script generation · W&B
  - Data transfer (Globus) · rclone
- **Context**
  - CSIRO HPCs · Canada vs Australia

What is HPC?

- **The concept**
  - Linux-based OS — familiar terminal environment
  - Shared resource — no sudo privileges
  - SLURM scheduler manages the job queue
  - Submit a job → runs when resources free up

- **Think of it as a very powerful computer,**
- **shared by many researchers, accessed via a queue**



- **Architecture**
  - Login node → you SSH here, edit files
  - sbatch job.sh ↓
  - SLURM Scheduler → queue manager
  - allocates resources ↓
  - Compute nodes → CPUs, GPUs, memory
  - Shared storage → /home, /scratch, /project

HPC Key Concepts

- **No sudo**
  - Cannot install system packages
  - Use conda/pip, modules, or containers instead

- **SLURM (job scheduler)**
  - Write a script: CPUs, GPUs, memory, time limit
  - sbatch job.sh → waits in queue → runs
  - squeue, scancel, sinteractive

- **Modules**
  - Software loaded via module load <name>
  - Pre-installed, version-managed
  - e.g. module load miniforge3 / cuda/12.2

- **Storage tiers**
  - Home: small quota (20 GB), backed up
  - Project: large, fast — use for jobs
  - Scratch: very large, NOT persistent
  - Scratch flushed periodically!

Why HPC for Robotics Research?

- **RL / Sim training**
  - Policy training in IsaacLab can take hours–days locally
  - Run on multiple GPUs simultaneously on HPC
- **Hyperparameter sweeps**
  - Test 50 configs simultaneously, not sequentially
  - Each config → one SLURM job → all run in parallel
- **Dataset processing**
  - Large datasets: video, point clouds, demonstrations
  - CPU + memory-intensive preprocessing at scale
- **Sim-to-real experiments**
  - Many simulation seeds / domain-randomization configs
  - Run in parallel → statistically robust results faster
- **Bottom line**
  - Your workstation runs 1 experiment; HPC runs 50

Computing Resources — What's Available for Monash Robotics

Resource	Type	GPU	Isaac Sim	Access
Robotics Cluster	University HPC	NVIDIA (various)	Yes	Lab members (Kubernetes)
Monash M3	University HPC	A40 / H100 / A100 / L4	Yes (5.1.0)	All Monash researchers — join project mh42
NCI Gadi	National HPC	H200 / A100 / V100	Yes	Apply / join project pg06
Pawsey Setonix	National HPC	AMD ROCm — NO CUDA	No	Apply at pawsey.org.au (CPU sweeps only)
CSIRO Virga/Petrichor	Internal HPC	NVIDIA CUDA	Yes (4.5.0)	CSIRO staff / affiliated students only
Alliance (Canada)	National HPC	NVIDIA (various)	—	Canadian accounts (comparison context)

M3 join: my.massive.org.au | NCI portal: my.nci.org.au | Pawsey: pawsey.org.au

Monash M3 — Getting Started

- **Step 1 — Get an account**
  - Register at my.massive.org.au
  - Join project mh42
    - ➢ (Machine Learning for HRI)
  - Approval ≈ 1 week
    - ➢ Project admin approves → M3 admin provisions
  - Ping your co-supervisor after submitting

- **Project ID: mh42**
- **Docs: docs.erc.monash.edu**
  - → Getting Started → Join Existing Project

- **Step 2 — Connect**
  - Option A — SSH (terminal)
    - ➢ ssh {m3_id}@m3.massive.org.au
  - Option B — Strudel (browser GUI)
    - ➢ strudel.hpc.monash.edu
      - ▪ Isaac Sim · Jupyter · RViz
  - Interactive GPU session (testing)
    - ➢ sinteractive --partition=gpu --gres=gpu:A40:1 --time=02:00:00

Register / join: my.massive.org.au | Docs: docs.massive.org.au | Strudel GUI: strudel.hpc.monash.edu

Monash M3 — Storage Layout

Location	Quota	Persistent?	Use for
/home/{m3_id}/	20 GB	Yes	Config, dotfiles, small scripts ONLY
/home/{m3_id}/mh42/	500 GB (shared)	Yes	Project files, conda envs, containers
/home/{m3_id}/mh42_scratch/	Large	**NO**	Temporary — flushed every 1–3 months
/home/{m3_id}/mh42_scratch2/	Large	**NO**	Temporary — flushed every 1–3 months

⚠ **Home = 20 GB only. Save envs, containers, datasets to mh42/ not home. Check quota: user_info**

```
/home/{m3_id}/mh42/FYP2026S1_3487/stack_yourname/    ← your personal stack
```

Docs: docs.massive.org.au | Quota cmd: user_info (run on M3)

Monash Robotics Cluster — Kubernetes-based GPU cluster

Alternative to M3 — custom Docker workflows, no job time limit

When to use the Robotics Cluster

Custom Docker image needed

No time limit on jobs (M3 max = 7 days)

Need persistent GPU reservation

Docs: github.com/MonashRobotics/cluster-training

Workflow overview

```
# 1. Build Docker image locally
docker build -t registry/myimage:v1 .

# 2. Push to lab registry
docker push mu00120825.../myimage:v1

# 3. Submit Kubernetes manifest
kubectl apply -f job.yaml

# 4. Monitor
kubectl get pods
kubectl logs <pod-name>
```

Training guide: github.com/MonashRobotics/cluster-training

National HPCs — NCI Gadi & Pawsey Setonix

- **NCI Gadi**
  - Login: gadi.nci.org.au
  - Portal: my.nci.org.au
  - GPUs: H200 (141 GB HBM3) / A100 / V100
  - Isaac Sim: Yes — full CUDA support
  - Project: pg06 (Lead CI: Dana Kulić)
  - Allocation: 117 kSU Q2 2026, 1 TiB scratch
  - GUI: are.nci.org.au (ARE)
- **→ Contact Lingheng to join pg06**

- **Pawsey Setonix**
  - Login: setonix.pawsey.org.au
  - Portal: portal.pawsey.org.au
  - GPU: AMD Instinct MI250X (ROCm)
  - CUDA: NO — AMD only
  - Isaac Sim: NOT supported
  - Best for: CPU RL sweeps (MuJoCo/Gym)
  - Expires: 2026-06-30 (current allocation)
- **→ CPU-only — no CUDA, no Isaac Sim**

NCI portal: my.nci.org.au | NCI ARE: are.nci.org.au | Pawsey: portal.pawsey.org.au

SLURM Basics — Submitting and Managing Jobs

A minimal GPU job script (job.sh)

Resources

```
#!/bin/bash
#SBATCH --job-name=my_training
#SBATCH --partition=gpu
#SBATCH --gres=gpu:A40:1       # 1× A40 GPU
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=8
#SBATCH --mem=32G
#SBATCH --time=08:00:00        # max wall time  HH:MM:SS
#SBATCH --output=logs/%j.out   # %j = job ID
```

Modules & Env.

```
module load miniforge3
conda activate myenv
```

Script

```
python train.py --config config.yaml
```

Key Commands

Submit	`sbatch job.sh`
Queue	`squeue -u $USER`
Cancel	`scancel <job_id>`
Interactive	`sinteractive ...`
Partitions	`sinfo`
Quota (M3)	`user_info`
Quota (NCI)	`nci_account`

Tip: only ask for what you need to get resources allocated quickly.

M3 SLURM docs: docs.massive.org.au/M3/slurm/simple-batch-jobs.html

Containers on HPC — Apptainer

- **Why not Docker?**
  - Docker needs root — not allowed on shared HPCs
- **Apptainer (formerly Singularity)**
  - Runs as your user (no root needed)
  - Open-source, available on M3 / NCI / Pawsey / CSIRO
- **Convert Docker → Apptainer in one command**
  - apptainer pull myimage.sif
  - docker://nvcr.io/nvidia/cuda:12.2-base
- **Existing Docker images work — no rewrite needed**
- **Isaac Sim 5.1.0 on M3 uses an Apptainer .sif**

- **Run a command inside the container**
  - apptainer exec --nv myimage.sif python train.py --nv = pass through NVIDIA GPU
- **Interactive shell**
  - apptainer shell --nv myimage.sif
- **In a SLURM script**
  - #!/bin/bash
  - #SBATCH --gres=gpu:A40:1
  - apptainer exec --nv /path/myimage.sif python train.py

Apptainer docs: apptainer.org/docs/

IsaacSim / IsaacLab on HPC

Most HPC GPUs are for AI/ML compute — NOT 3D rendering
Isaac Sim ray-tracing requires RT Cores. Many HPC GPUs (A100, H100, V100) do NOT have them.

GPU	RT Cores?	Isaac Sim?	Available on
A40	Yes	Yes	Monash M3 ← recommended for Isaac Sim
H200	No	No	NCI Gadi
A100	No	No	M3, NCI Gadi
H100	No	No	M3
V100	No	No	NCI Gadi
L4	Yes	Not tested	M3
T4	No	No	M3

IsaacLab: isaac-sim.github.io/IsaacLab | M3 GPUs: docs.massive.org.au/M3/slurm/gpu-jobs.html

AI Coding Tools — A Game Changer for HPC Workflows

- **Claude Code (Anthropic)**
  - Terminal agent in VS Code
  - Works over VS Code SSH Remote — runs on M3
  - Reads files, writes scripts, fixes errors
  - Persistent memory across sessions

- **GitHub Copilot**
  - Inline code completion as you type
  - Context-aware — great for SLURM boilerplate

- **ChatGPT / Claude.ai**
  - Web-based — explain errors, debug logic
  - Cannot access your files directly

- **Typical workflow with Claude Code**
  - 1. Develop & test locally
  - 2. Push to git
  - 3. Pull on HPC (git pull)
  - 4. Open VS Code SSH Remote → M3
  - 5. Run Claude Code on M3 directly

- **Claude reads your code, writes SLURM scripts,**
- **debugs errors — while running on M3**

Claude Code: claude.ai/code | Copilot: github.com/features/copilot

Job Script Generation & Experiment Tracking

Python / Jupyter: generate many job

Generate 20 configs, submit in one go:

```
import subprocess

lrs     = [1e-4, 3e-4, 1e-3]
batches = [64, 128, 256]

for lr in lrs:
  for bs in batches:
    script = f'''#!/bin/bash
#SBATCH --gres=gpu:A40:1
#SBATCH --time=04:00:00
python train.py --lr {lr} --batch {bs}
'''
    fname = f'job_lr{lr}_bs{bs}.sh'
    with open(fname,'w') as f: f.write(script)
    subprocess.run(['sbatch', fname])
```

Weights & Biases (W&B)

Cloud experiment tracking — runs, metrics,

```
import wandb

wandb.init(project='my-rl',
           config={'lr': 3e-4})

for step, (loss, reward) in enumerate(train()):
    wandb.log({'loss': loss,
               'reward': reward})
```

- Live dashboard in browser
- Compare runs side by side
- Built-in sweep support (wandb.agent)
- HPC jobs send metrics to cloud
- Free tier sufficient for research

W&B: wandb.ai | W&B sweeps: docs.wandb.ai/guides/sweeps

Data Transfer & Visualization

Globus — reliable large data transfer

- app.globus.org or CLI — no extra SSH setup
- Resumes automatically if connection drops
- Checksum verification — no silent corruption
- HPC-to-HPC directly (no local hop needed)
- Supported: M3, NCI, Pawsey, most HPCs

rclone — browse HPC files locally

```
# 1. Install: rclone.org/install/
# 2. Configure (run once):
rclone config  # sftp, host: m3.massive.org.au

# 3. Mount as Z: (Windows) or ~/mnt (Linux/Mac):
rclone mount m3:/home/{m3_id}/mh42/ Z: --vfs-cache-
mode writes
```

Mount /mh42 as a drive in Explorer/Finder. Uses existing SSH key.

Visualization on HPC

HPCs are SSH-only by default — no display. All major HPCs have a remote desktop / GUI option.

Monash M3 Strudel	*strudel.hpc.monash.edu*
NCI Gadi ARE	*are.nci.org.au*
Pawsey Nimbus / VNC	*see Pawsey docs*
CSIRO Virga OOD portal	*rms.csiro.au*

Use cases: Isaac Sim · RViz · matplotlib · Jupyter · TensorBoard

Globus: app.globus.org | rclone: rclone.org | Strudel: strudel.hpc.monash.edu | NCI ARE: are.nci.org.au

CSIRO HPCs — Virga & Petrichor

- **Internal — CSIRO-affiliated only**
  - Context: for potential Monash–CSIRO collaboration
- **Virga**
  - Login: virga.hpc.csiro.au
  - Portal: rms.csiro.au
  - GPUs: NVIDIA CUDA — SLURM
  - Isaac Sim: 4.5.0 via Apptainer
  - Storage: /datasets/work/hri-isaaclab-storage/
- **Both use SLURM — same workflow as M3 / NCI**
- **Petrichor**
  - Login: petrichor.hpc.csiro.au
  - Portal: rms.csiro.au
  - CPU-focused
  - Access: CSIRO account required
- **CSIRO HRI Lab (Clayton)**
  - Franka arms · Jetson Thor
  - RTX 6000 workstations
  - MSI GB300
  - HPC access
  - Open to joint projects
- **→ Talk to Lingheng if interested**

CSIRO HPC portal: rms.csiro.au

HPC Ecosystems — Canada vs Australia

- **Canada — Digital Research Alliance**
  - One account → ALL national HPCs
  - Nibi, Fir, Cedar, Graham, Narval, Béluga
  - Consistent paths everywhere:
  - /home /scratch /project
  - Same modules → same script runs anywhere
  - unchanged across systems
  - Apply once: ccdb.computecanada.ca

- **Australia — Fragmented**
  - Separate account per system:
  - NCI Gadi · Pawsey · Monash M3 ...
  - Inconsistent storage paths:
  - NCI: /scratch/pg06/
  - Pawsey: different layout entirely
  - Different modules → scripts need edits
  - Separate application processes
  - NCI: 2×/yr · Pawsey: quarterly · M3: ongoing

Alliance (Canada): ccdb.computecanada.ca | Alliance docs: docs.alliancecan.ca

Summary & Next Steps

- **Start with M3**
  - Join project mh42 at my.massive.org.au
  - Most Monash Robotics people start here
- **Isaac Sim needs A40**
  - Always request --gres=gpu:A40:1 on M3
  - Other GPUs lack RT Cores
- **Use Apptainer**
  - Convert your Docker image → .sif in one command
  - Works everywhere: M3 · NCI · Pawsey
- **NCI Gadi for more compute**
  - H200/A100 via project pg06
  - Contact Lingheng or Dana
- **AI tools save time**
  - Claude Code + Copilot — write & debug faster
- **Python sweeps + W&B**
  - Generate 50 job scripts programmatically
  - Log everything to Weights & Biases
- **Happy to help you get started after the talk**

Useful Links

Monash M3

Docs: docs.massive.org.au

Join / login: my.massive.org.au

Strudel GUI: strudel.hpc.monash.edu

NCI Gadi

Portal / join: my.nci.org.au

Docs: opus.nci.org.au/display/Help

ARE (GUI): are.nci.org.au

Pawsey Setonix

Portal: portal.pawsey.org.au

Docs: pawsey.org.au/systems/setonix

Robotics Cluster

Training guide: github.com/MonashRobotics/cluster-training

Tools

Globus: app.globus.org

rclone: rclone.org

AI Tools

Claude Code: claude.ai/code

GitHub Copilot: github.com/features/copilot

Questions & Discussion

Slides will be shared to the slack channel.