

Why HCP?

High-Performance Computing: CCMAR's CETA & National Infrastructure

January 2026, David Palecek



Laptop to HPC

High-Performance Computing (HPC) is not about “faster computers”
It’s about making certain problems possible at all

- HPC removes artificial constraints on ideas
- Modern research & data workflows exceed laptop limits

Laptops break down when:

- Datasets exceed RAM
- Analyses take days or week
- You need reproducibility at scale

HPC is about possibility, not luxury



Memory vs Compute

If your data doesn't fit in memory, computation fails, not just slows

- Large matrices, graphs, genomes, images
- In-memory analytics (pandas, numpy, R)
- Slow disk swapping and crashes
- Genome assembly

More CPU doesn't just make things faster:

- Enables higher resolution models
- Parameter space sampling
- Batch processing of many samples at once
- Makes "load once, analyze many times" possible

Understand your bottleneck = smarter HPC use



Warning: most programs are single-threaded

- Many standard tools use only 1 CPU core
- More cores \neq faster by default
- HPC does not magically parallelize your code

Common examples:

- Many bioinformatics tools
- Default Python / R scripts
- Legacy scientific software

You must choose parallel tools or parallelize workflows



Common patterns

1. Many independent jobs
 - Run 1,000 single-core jobs at once
 - BLAST
2. Multithreaded programs
 - One job using many cores
3. Distributed workflows
 - Pipelines, task schedulers, workflow engines
 - Nextflow, Snakemake, CWL



CETA

5 nodes

- 128 GB
- 54 CPUs

1 BIG memory node

- 900 GB
- 54 CPUs

Submission via job queue system, SLURM. Any other compute from the login node is **FORBIDDEN**.

The more resources you ask for, the longer you will wait to get your job started (on commercial Cloud or EU/national infrastructure), you get billed by what you allocated, not by what you used).

Ask for help, whenever unsure.