



**Hewlett Packard  
Enterprise**

# Practical sessions

Comprehensive General LUMI Course

April 23–26, 2024



# Materials Locations

---

- Slides will be uploaded soon after talks  
`/project/project_465001098/Slides/HPE`  
`Files *.pdf`
- Exercise notes and files (should include PDFs or Readme.md with instructions)  
`/project/project_465001098/Exercises/HPE`
- Copy exercise files into your `$HOME` directory
  - If needed, unpack the exercise tar files with  
`tar xf <file>.tar`  
`tar xf <file>.tar.gz`



# Setup

- A reservation is setup for use during the training

- Use the following flags in the SLURM commands:

**-A project\_465001098 --reservation=LUMItraining\_C**  
**(or -reservation=LUMItraining\_G)**

- To run the examples either use above options with sbatch/srun/salloc or you can also set SLURM environment variables, e.g.

```
export SLURM_ACCOUNT=project_465001098
export SLURM_RESERVATION=LUMItraining_G
```

(to be repeated for variables with prefix **SLURM\_**, **SBATCH\_**, **SALLOC\_**)

- For convenience, we provide a script to setup your environment (copy from **/project/project\_465001098/Exercises/HPE**):

- **source lumi\_c.sh # LUMI-C**
  - **source lumi\_g.sh # LUMI-G**

➔ It will change the prompt accordingly, remember to run `exit` before you switch environment



# Exercises – Day 1 Introduction

- Exercise notes and files (should include PDFs or Readme.md with instructions)  
**/project/project\_465001098/Exercises/HPE/day1**
- Directory: **ProgrammingModels**
  - Session 1:  
Run on the system and get familiar with the SLURM commands
  - Session 2:  
Try different compilers and flags
- Directory: **libsci\_acc**
  - Session 3:  
Test with LibSci\_ACC, check the different interfaces and environment variables

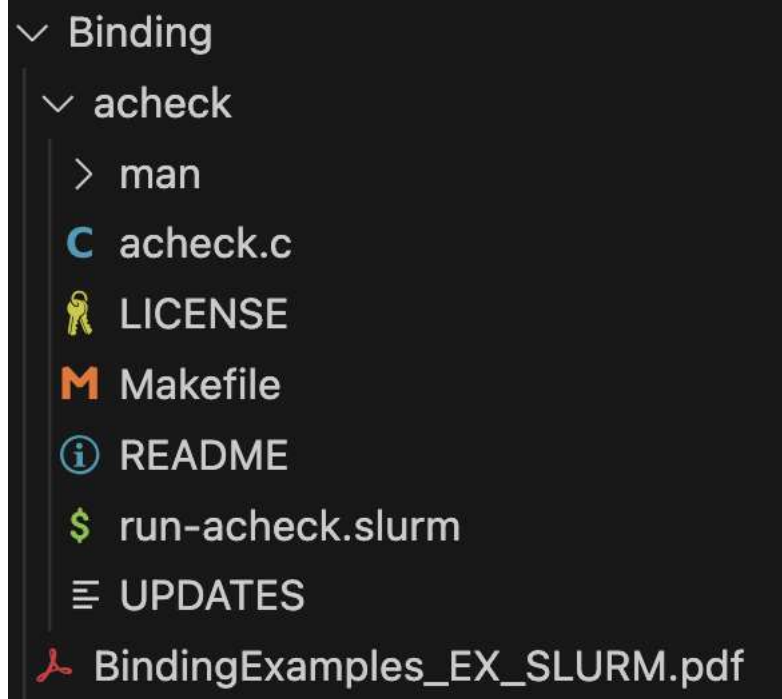


# Exercises – Day 2 Introduction

- Exercise notes and files (should include PDFs or Readme.md with instructions)  
**/project/project\_465001098/Exercises/HPE/day2**
- Directories: **debugging** (within directories)
  - Session 4:  
Try the debugging tools in debugging sub-directory
- Directories: **Binding, gpu\_perf\_binding**
  - Session 5:  
Try different binding options for CPU execution (look at slides and use envvars to change order and display the order) and for GPU execution (gpu\_perf\_binding, see next slide)



# CPU binding : HPE/day2/Binding/



Get the right environment:

- `source ../../lumi_c.sh`

Compile acheck :

- `cd acheck && make`
  - Default modules produces *acheck-cray*

Look at the sbatch example

- `cat run-acheck.slurm`
  - test different bindings
- `sbatch run-acheck.slurm`



# GPU binding : HPE/day2/gpu\_perf\_binding/

Get the right environment:

- `source ../../lumi_g.sh`
- `source gpu_env.sh`

Compile the two applications:

- `cd hello_jobstep && make`
- `cd himeno && make`

Start with hello\_job\_step example

- `cd hello_jobstep`
- `sbatch job.slurm`
  - test different bindings
  - `gpu_bind variables...`

Launch Himeno from the root directory

- `cd himeno`
- `sbatch job.slurm`
  - test different bindings

```
✓ gpu_perf_binding
  > hello_jobstep
  > himeno
  ◆ .gitignore
  $ gpu_env.sh
  ⓘ README.md
  $ select_gpu_naive.sh
  $ select_gpu.sh
```



# Exercises – Day 3 Introduction

- Exercise notes and files (should include PDFs or Readme.md with instructions)  
**/project/project\_465001098/Exercises/HPE/day3**
- Directories: **perftools-lite**, **perftools-lite-gpu**
  - Session 7:  
Follow the Readme.md description and get familiar with the perftools-lite commands and outputs
    - subdirectory perftools-lite needs lumi\_c.sh to be sourced
    - subdirectory perftools-lite-gpu needs lumi\_g.sh to be sourced
- Directories: **perftools** (within directories)
  - Session 8:  
Follow the Readme.md description (per each directory) and get familiar with the perftools commands and outputs
    - subdirectories perftools, perftools-api, perftools-hwpc, perftools-python, and perftools-apa need lumi\_c.sh to be sourced
    - subdirectories perftools-for-hip and perftools-for-omp-offload need lumi\_g.sh to be sourced
- Directories: **ProgrammingModels**
  - Session 9:  
Test the Pi example with MPI or MPI/OpenMP on 4 nodes and 4 tasks  
Show where the ranks/threads are running by using the appropriate MPICH environment variable  
Use environment variables to change this order (rank-reordering)

