EuroHPC Development Access Proposal – MiniWeather Modernization

Group: Sofía Gonzalez, Hibba Kamas, Gabriela Vega, Cayetana Hinostroza, Diego Oliveros, Sabina Bacaoanu, Diana Cordovez

1. Abstract and Objectives

MiniWeather is a compact 3D stencil workload distilled from operational weather models. We request EuroHPC Development Access to:

1. Port and optimize the MPI + OpenACC code base for next-generation GPU nodes.
2. Quantify strong and weak scaling across 1–64 nodes, with and without hybrid threading.
3. Introduce algorithmic improvements (communication overlap, cache/blocking strategies) guided by Nsight Systems/Compute and LIKWID/PAPI profiling.
4. Deliver reproducible artifacts (containers, Slurm scripts, plots) and a public report feeding into a larger atmospheric modeling project at IE University.

2. State of the Art

- MiniWeather is widely used in tutorials but lacks published large-scale scaling data on EuroHPC-class machines. Prior work focuses on single-node performance.
- Current IE cluster offers only two GPU nodes (1 GPU each), limiting experimentation to 2 GPUs.
- Communication-avoiding stencil variants (e.g., halo fusion, redundant computation) have yielded 1.4–2× speedups in literature; we aim to validate similar techniques in our fork.
- Tooling: we already rely on perf, Nsight Systems/Compute, and custom metrics logging; this will extend to LIKWID/PAPI or Intel VTune depending on target hardware.

3. Current Code Base and TRL

- Repository: `MiniWeatherApp` (public GitHub once sanitized). Contains serial, OpenMP, MPI, hybrid, and MPI+OpenACC implementations plus comprehensive Slurm workflows (`slurm/`), data scaffolding, and plot scripts.
- Technology Readiness Level (TRL): 4–5 (researched and validated in lab) – code runs reliably on the local cluster but has not been ported or optimized for large EuroHPC systems.
- Recent additions: automated plotting (`scripts/plot_scaling.py`), sacct capture script, and a growing bottleneck analysis document.

4. Target System and Software Stack

- Preferred machine: **LUMI-G** (AMD Instinct MI250X GPUs) or **Leonardo Booster** (NVIDIA A100). We require at least 64 GPU nodes for scaling sweeps.
- Toolchain:
    - MPI: Cray MPICH (LUMI) or OpenMPI (Leonardo).
    - Compilers: Cray/ROCm (HIP) for LUMI, NVHPC + CUDA for Leonardo.
    - Programming models: MPI + OpenMP for CPU-only runs, MPI + HIP/OpenACC for GPUs. We plan to maintain both CUDA/HIP branches for portability.
    - Profilers: rocprof, Nsight Systems/Compute, LIKWID/PAPI, perf.
    - Libraries: optional use of NVSHMEM/HIPSHMEM for halo exchange experiments; BLAS not required.
- Packaging: Apptainer recipe (`env/project.def`) will be adapted per target, ensuring consistent builds.

5. Work Plan, Milestones, and Risks

Phase	Month	Activities	Milestones	Risks / Mitigation
P1 – Port & Validate	1	Adapt build system to Cray/NVHPC environments, run baseline correctness on 1 node.	Container + sbatch templates verified.	Risk: compiler incompatibility → mitigate via vendor support tickets + simplified kernels.
P2 – Profiling & Baseline Scaling	1–2	Collect perf/LIKWID/Nsight/rocprof traces on 1,2,4 nodes; establish METRICS logs + sacct data.	Profiling reports archived in `results/profiling/`.	Risk: queue wait times → plan night runs, request reservation if needed.
P3 – Optimization Sprint	2–3	Implement overlapping communication, cache blocking, GPU stream concurrency,	Demonstrated >1.5× speedup over baseline on ≥16 nodes.	Risk: limited developer availability → share tasks among IE +

Phase	Month	Activities	Milestones	Risks / Mitigation
		and hierarchical partition keys for data layout.		partner institution.
P4 – Scaling to 64 Nodes	3	Run strong/weak scaling sweeps up to requested limit, generate plots + bottleneck summary.	Report + plots ready for short paper submission.	Risk: hardware failures → schedule contingency buffer.
P5 – Documentation & Dissemination	4	Finalize short paper, EuroHPC reports, GitHub release, FAIR data package.	Publish PDF + deposit dataset in Zenodo.	Risk: missing GPU data → ensure Phase 2 success before moving on.

6. Resource Justification

- CPU nodes: 2000 core-hours for baseline + hybrid scaling (perf, LIKWID runs included).
- GPU nodes: 1500 GPU-hours (64-node strong/weak sweeps + profiling overhead). Each experiment logs METRICS + sacct output; we will reuse built binaries per parameter set.
- Storage: <50 GB for logs, Nsight/rocprof traces, and plot outputs.
- Support: brief onboarding with site application experts (profiling counter selection, high-throughput submission best practices).

7. Data Management, FAIR, and Ethics

- All scripts, CSVs, and plots live in a public Git repository (MIT license) with DOIs minted for major releases (via Zenodo).
- Raw Nsight/rocprof files and sacct logs (<50 GB) will also be shared when permissible, or summarized with reproducible parsing scripts if vendor EULAs restrict redistribution.
- Metadata (grid sizes, compiler flags, job IDs) stored in CSV/JSON to guarantee reuse.
- No personal data involved; only synthetic weather fields. Ethics considerations limited to responsible resource usage and transparent reporting.

8. Expected Impact

- Provides a vetted reference implementation + workflow for weather-model kernels on EuroHPC systems.
- Demonstrates best practices for packaging, profiling, and scaling mid-size atmospheric workloads, easing future project onboarding.
- Delivers documented bottlenecks + optimization recipes applicable to larger codes (ICON, IFS, FV3) that also rely on stencil kernels.
- Builds collaboration between IE University and EuroHPC centers, paving the way for subsequent Pilot or Benchmark access proposals focused on full weather models.