第二次作业说明

陈健达

2025 年 10 月 31 日

1 作业背景

MiniWeather 是在一个简单实现的大气动力学模拟程序包。该程序包采用有限体积法实现大气动力过程显式积分仿真，是一个典型的模板计算类程序。同时，该程序包已经完成多种语言实现和并行实现，包括 C、C++ 和 Fortran 实现，以及串行、MPI 并行、OpenMP 并行实现等。该程序包也支持多种配置选择，包括不同实验算例、不同模拟空间分辨率与可变的模拟时长等。

该程序包的编译说明、实验设置说明、正确性检验说明和数值计算方法说明详见 miniweather 目录下的 README 部分 (原项目 github 网页为 (https://github.com/mrnorman/miniWeather)。

2 作业描述

2.1 作业要求

1. 每位同学独立完成。

2. 提交文件夹命名格式为学号 + 作业编号 + 姓名, 如第二次作业: 2025000000_h2_name，其中包含报告 report.pdf。

3. 注意 DDL，以网络学堂发布的为准。

2.2 作业任务: MiniWeather 并行优化

在CPU 集群上对 MiniWeather 的特定实验算例进行性能优化。本次作业中所提供的源代码对 MiniWeather 原始仓库的代码进行了删减，而

README 文件保留了原始的 README 文件，因此请大家酌情阅读。有需要的同学可以参考原项目 github 网页。

特定实验算例为“colliding thermals”（对应于参考代码中 cmake_hpc.sh 的 DATA_SPEC 参数，代码中已经设置好），网格设置：x 方向网格数 nx_glob = 3200；z 方向网格数 nz_glob = 1600；模拟时长：sim_time = 300s；IO 输出频率为每 50s（可通过 cmake_hpc.sh 文件进行设置，参考 README 的“Altering the Code's Configurations”章节）。

解压 miniWeather.zip 后，首先参考 env_build.md 中的说明在课程集群上完成相应依赖库的装载。然后参考 README 的 “Compiling and Running the Code”部分在实验集群环境中建立基线程序版本（c/build/文件夹下），并确保其正确运行。建议采用 MPI+OpenMP 版本作为基线，进一步进行性能优化，可以采用编译优化、代码优化等。

2.3 任务要求

1. 测试结果以程序运行的端到端时间为评判标准，并建议分别记录初始化时间、积分计算时间（不含 IO）和 IO 时间，计时方法和计时结果请在报告中详细说明。

2. 参考 README 中“Checking for Correctness”部分所提供的质量守恒和能量守恒方法检查性能优化实现的正确性，并在遵循其正确性判断标准的前提下开展优化。

3. 绘制并报告性能曲线：在固定问题规模下，记录端到端时间随核数（核数定义为 MPI 进程数 × 每进程 OpenMP 线程数）的变化曲线（time vs 核数），并计算加速比 Speedup = T_1/T_p（其中 T_1 为单核时间，T_p 为 p 核时间），绘制 speedup vs 核数曲线。请至少覆盖核数：1、2、4、8、16、32。

2.4 作业评分

1. 通过质量和能量守恒的正确性检测 (15%)（正确性测试方法为在 c/build 文件夹运行 srun -n 1 make test）；

2. 评测 MiniWeather 的端到端的性能结果进行打分 (40%)；

3. 在报告中**详细描述**采取的优化手段，代码对应的部分，以及对应的实验结果（包括并行可扩展性、时间 breakdown 等），来解释目前取得的性能结果 (30%)；

4. 在报告中**详细展示**不同变量的模拟结果（参考 README 中"Viewing the Output" 章节）以及优化前后的偏差情况，说明优化后模拟结果的正确性 (15%)。

2.5 作业提示

1.OpenMP 优化的过程中要考虑变量的共享或者私有属性，负载的分配方式，以及 NUMA 效应带来的影响。

2.MPI 优化的过程中，要考虑通讯的开销，鼓励进行计算通信隐重叠等设计。

3. 调试过程中建议设置较粗的分辨率、较短的运行时间，节省调试成本。

4. 与助教和老师及时交流作业和实现方法遇到的问题。

3 参考资料

[1]~ [2] 是和性能模型有关的部分工作。

MiniWeather 涉及大量的模版计算，[3] 是最近的针对模板计算问题进行性能自动调优的工作。[4] 是 intel 在 CPU 和 GPU 上对 3.5-D 模版计算问题的优化。

大家可以阅读以上的文献，也可以在网上自行查找其他的相关文献。

参考文献

[1] David Culler, Richard Karp, David Patterson, Abhijit Sahay, Klaus Erik Schauser, Eunice Santos, Ramesh Subramonian, and Thorsten Von Eicken. Logp: Towards a realistic model of parallel computation. In *Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 1–12, 1993.

[2] Benjamin C Lee, David M Brooks, Bronis R de Supinski, Martin Schulz, Karan Singh, and Sally A McKee. Methods of inference and learning for performance modeling of parallel applications. In *Proceedings of the 12th ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 249–258, 2007.

[3] Mingzhen Li, Yi Liu, Hailong Yang, Yongmin Hu, Qingxiao Sun, Bangduo Chen, Xin You, Xiaoyan Liu, Zhongzhi Luan, and Depei Qian. Automatic code generation and optimization of large-scale stencil computation on many-core processors. In *50th International Conference on Parallel Processing*, pages 1–12, 2021.

[4] Anthony Nguyen, Nadathur Satish, Jatin Chhugani, Changkyu Kim, and Pradeep Dubey. 3.5-d blocking optimization for stencil computations on modern cpus and gpus. In *SC '10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–13, 2010.