

并行与分布式计算导论 作业 1

PDC 2025 Spring Homework 1

截止日期 2025 年 4 月 20 日 23:59

DDL: 2025 Apr. 20 23:59 (GMT+8)

归约 (Reduction) 是并程序设计中常用的算子之一，用于将一个数组中的多个元素通过给定的运算符合并成单个元素。合法的归约运算符需要满足结合律，因此，归约操作的最终结果可以通过部分结果汇总得到。常用的归约运算符包括+、*、-、&、|、^、&&、||、min 和 max。

给定长度为 n 的数组 $X = \{x_0, x_1, \dots, x_{n-1}\}$ ，以及定义在 X 元素上满足结合律的二元运算符 \oplus ，归约操作可以定义为：

$$\text{result} = \bigoplus_{i=0}^n x_i.$$

本次作业要求使用 OpenMP 和 CUDA 对归约求和（即规约运算符为加法）进行并行加速，请实现以下两个任务，并按要求撰写书面报告。将书面报告和代码源码¹共同打包为压缩包上传至教学网。

任务 1

通过 OpenMP 对规约求和进行加速。要求使用**自定义 reduction 子句**²对求和进行并行计算，并实现 **SIMD 批量操作**。例如，针对 32 位整型数据的归约，利用 256 位 AVX 的 SIMD 指令，实现批量计算的优化。

任务 2

阶段 1: 使用 CUDA 对归约操作进行加速。参考 reduction.zip 中给出的代码框架，需同时优化以下方面：**使用共享内存并合并内存访问**：确保线程对全局内存的访问模式高效；**避免共享内存的 bank 冲突**：优化共享内存的使用方式；**利用 Warp Shuffle 指令**：通过 Warp 内部的高效数据交换优化归约操作。

¹ 或者在书面报告中指明 Github 等代码仓库地址，并且授权助教访问权限（助教 GitHub ID 为 jieran-zhang，其他代码仓库可微信单独联系助教）。

² <https://www.openmp.org/spec-html/5.0/openmpsu107.html> 中 2.19.5.7 declare reduction Directive 等可供参考；此外，由于北大未名集群 gcc OpenMP 版本过低，不支持自定义 reduction，因此该任务需要用个人电脑进行编译。

阶段 2: 使用 Nsight Compute³对你写的 kernel 进行分析，尝试通过 Nsight Compute 的报告数据证明阶段 1 的优化，如通过低 Hit Rate 证明 kernel 实现了内存合并访问。

书面报告要求

对于以上两个任务，分别从**并行算法设计、编程实现思路、测试结果分析**三个方面撰写报告，报告要求尽量**简洁**。**不要粘贴代码截图**。最终在教学网将**实验报告与源代码**打包提交。写代码和报告时可以使用大模型辅助，但是最终提交版本的内容需要本人理解并确认。

³ 可考虑在北大未名集群中获取报告文件，在本地下载 Nsight Compute 图形界面进行查看。