并行计算 II 2023 春季第二次作业 MPI 优化 Attention

丁明朔

1 Scaled Dot-Product Attention

参考文献 [VSP+17] 中提出了 Transformer 架构，其中的核心算法是 Scaled Dot-Product Attention，定义如下：

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V}$$

其中，$\boldsymbol{Q} \in \mathbb{R}^{N \times d_q}$，$\boldsymbol{K} \in \mathbb{R}^{N \times d_k}$，$\boldsymbol{V} \in \mathbb{R}^{N \times d_v}$，softmax 函数是矩阵的每个行向量做 softmax，定义如下：

$$\text{softmax}(\boldsymbol{A})_{i,j} = \frac{e^{\boldsymbol{A}_{i,j}}}{\sum_k e^{\boldsymbol{A}_{i,k}}}$$

作业要求基于给定的 Attention 串行算法实现 (attention.cpp)，使用 MPI 实现 Scaled Dot-Product Attention 算法，并提供优化报告。

2 作业要求

根据附件中给定的 Attention 串行算法实现 (attention.cpp)，使用 MPI 进行优化。具体要求如下：

- 输入矩阵文件为 input1.in 和 input2.in，代码需在两个算例下运行通过；
- 可改变读入方式，此部分无需进入耗时；
- 串行实现中除 `reduce_the_sum` 及 `check` 函数外，其他均可以改动；
- 基于数院集群，最多使用 4 个节点，进程数与线程数不限，允许使用 OpenMP 进行线程级的并行；
- 不限制编程语言、编译选项和编译器版本 (推荐 C/C++ 实现)；
- 提交完整的并行程序代码，要求程序中包括 MPI 并行代码/计时模块/正确性验证，提供与非优化版本的性能加速比较和分析；
- 提交报告中要求说明进程数目，划分方法以及每个进程的任务负载；
- 鼓励报告中提供分阶段优化或者多版本优化的分析和比较。

3 提交要求和评分标准

- **提交：**将代码和报告打包后发邮件至：parcoii2023@163.com，邮件主题为“学号-第 X 次作业”，在 **6 月 30 日 23 点 59 分**前提交，可更新结果至多 3 次。
- **评分：**（1）基本要求：基于串行实现完成 Attention 算法的并行程序，代码运行通过，在报告中给出并行划分方法、运行时间和正确性验证；(2) 分析对比优化前后，哪些策略带来了明显性能收益，**代码实际加速效果越好分数越高**；（3）其他加分项：根据报告内容丰富程度、代码质量酌情加分。

References

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.