并行计算基础第三次作业说明

黄浩鹏

2025 年 4 月 6 日

1 作业背景

Attention 机制最初用于提升序列到序列模型（Seq2Seq）在处理长文本时的效果，通过允许模型动态聚焦输入序列的不同部分，解决了传统方法的瓶颈。它通过计算查询（Q）、键（K）和值（V）之间的相似度来加权每个值，从而生成新的表示。Attention 被广泛应用于 NLP、计算机视觉等领域，尤其是 Transformer 模型中。随着模型应用的扩展，优化 Attention 计算的效率变得尤为重要，常见的优化技巧包括稀疏化计算、低秩近似、多机分布式计算以及 GPU 加速等方法，以减少计算和内存开销，提升效率。

2 作业描述

2.1 作业目标

掌握单 GPU 的并行编程和优化方法，了解 Attention 机制的基本原理。

2.2 作业要求

1. 独立完成代码实现与优化。

2. 可以参考矩阵乘法相关的外部库（如 BLAS 或 BLIS 等数学库）的实现与优化思路，但禁止直接使用外部库作为作业成果。

3. 可以使用 AI 工具（如 ChatGPT、DeepSeek、GitHub Copilot 等）辅助编程，但必须确保对代码的理解。提交内容应基于自己的理解与实现，禁止直接抄袭他人代码或让 AI 工具生成完整的作业代码。

4. 推荐仅在登录节点上编译程序，并使用 srun 或 sbatch 提交到计算节点运行程序。请勿在登录节点上直接运行程序，否则可能影响系统响应，相关进程会被强制终止。请大家自觉遵守。

5. 注意作业截止时间，以网络学堂发布的为准。

6. 提交单个压缩包，命名格式如：2024000000_h3_name.zip（学号、作业编号、姓名），其中包含代码文件夹 code，和报告 report.pdf。

2.3 作业任务：方阵的 Attention 机制计算

2.3.1 任务描述

完成 Attention 算子在GPU 单卡上的实现与优化。Attention 算子:

$$Y = \text{Softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{N}}\right)V,$$

其中，Q，K，V 均为 $N \times N$ 的单精度**行优先**存储的稠密矩阵。

其中 Softmax 的定义为

$$\text{Softmax}([z_1, ..., z_n]) = [\frac{e^{z_1}}{\sum_{j=1}^{n} e^{z_j}}, ..., \frac{e^{z_n}}{\sum_{j=1}^{n} e^{z_j}}].$$

在 Attention 计算中，Softmax 的计算是逐行进行的，具体算法实现可以参考“code/cpu/naive.c”中的代码。在熟悉代码核心后，优化版本的程序实现在“code/gpu/opt.cu”中。

2.3.2 正确性验证

1. 采用 **float 单精度浮点数据类型**进行运算，运算结果通过作业基础代码中的正确性验证，eps 为机器的单精度极小值，约为 10^{-6} 左右。

$$||custom\_attention(n, Q, K, V) - Label|| < 100n^2\epsilon$$

其中，$||\cdot||$ 表示将矩阵逐元素取绝对值求和（即向量的 1 范数）。

2.Attention 的**主要计算开销在两次矩阵乘法**，其计算复杂度为 $O(N^3)$，在计算性能指标的时候采用 $(4N^3)$ 计算，如果采用了一些非 $O(N^3)$ 算法而导致通过不了正确性测试，这种情况可以适当且合理地放宽精度的要求，但需要在作业报告中指出。

3. 开展必要的性能分析，比如某些矩阵规模性能出现明显的降低，可以采用性能分析工具进行性能分析。

2.3.3 运行方法

代码可以从网络学堂上获取，其中附有 example.txt 日志可供参考。课程集群的登录和程序运行请参考网络学堂文件《课程集群使用手册》。code 文件夹中，有两个子目录 cpu 和 gpu，分别对应于两种不同的平台。其中 cpu 子目录下包含 naive.c 样例，作为最简单的实现参考，gpu 子目录中的 opt.cu 作为本次作业要实现的代码。

2.4 作业评分

2.4.1 GPU Attention 算子并行优化 100%

1. 通过正确性检验（20%）。

2. 评测 Attention 算子在不同输入（共 102 个测例）下的性能结果，按照提交后的性能排序结果，以及代码质量进行打分 (50%)。

3. **详细描述**在实现 Attention 算子中采取的优化手段，代码对应的部分，以及对应的实验结果，可以采用性能工具或者模型来解释目前取得的性能结果 (20%)。

4. 给出一张完整的实验结果图，描述当前算法的性能，横坐标为矩阵规模，纵坐标为 $Gflop/s$(10%)。

2.5 作业提示

1. 在保证正确性的前提下，可对计算流程中的某些冗余部分进行删减。

2. 本作业的核心模块是两次矩阵乘法，可以借鉴第一次作业 SGEMM 的优化思路，在 GPU 上优化好矩阵运算的子模块。

3. 有任何问题欢迎与助教和老师交流。

3 参考资料

[1, 2, 3] 是 GPU 上矩阵乘法计算的参考文献。针对 Attetion 的优化可以参考 FlashAttention 的实现，如 [4]。

参考文献

[1] Kayvon Fatahalian, Jeremy Sugerman, and Pat Hanrahan. Understanding the efficiency of gpu algorithms for matrix-matrix multiplication. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 133–137, 2004.

[2] Junjie Li, Sanjay Ranka, and Sartaj Sahni. Strassen's matrix multiplication on gpus. In *2011 IEEE 17th international conference on parallel and distributed systems*, pages 157–164. IEEE, 2011.

[3] Guangming Tan, Linchuan Li, Sean Triechle, Everett Phillips, Yungang Bao, and Ninghui Sun. Fast implementation of dgemm on fermi gpu. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–11, 2011.

[4] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.