并行与分布式计算基础 2021 秋季第一次作业
MPI 实现 In-place AllReduce 算子

叶子凌锋　罗昊

2021 年 11 月 1 日

1 问题介绍

全规约 (AllReduce) 是一种集合通信原语。对于一个满足交换律和结合律的算子，它的语义是将每个进程的结果进行规约并确保每个进程都得到规约后的结果，见图 1。这一通信算子在当下大量应用于分布式深度学习的数据并行训练中，人们使用该算子将每次迭代后向计算得到的梯度值求平均并规约到数据并行进程组内的每一个进程中，保证每个模型拷贝更新使用的梯度值都是相同的。

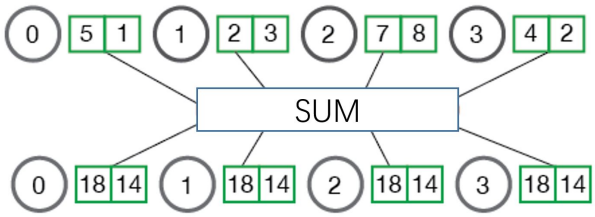

图 1: In-place AllReduce **示意**。以求和为例子，对于待规约的每一个位置的值进行求和，最终每个进程都能获得求和后的值，并存储在原来的待规约数组的位置上。

虽然 In-place AllReduce 的语义并不难理解，但是在分布式训练的场景下，人们常常面对复杂的网络环境：同一节点内不同加速卡（例如 GPU）之间互联的带宽较大，而跨节点的网络带宽往往较小……为了避免带宽的短板对整个集合通信效率的影响，人们对 AllReduce 的实现算法进行了很多改进的尝试。本次作业我们将通过使用 MPI 基本的 Send/Recv 原语实现 In-place 的 AllReduce 算法，探究其实现的优劣，加深对集合通信和消息传递的分布式编程模型的理解。

2 任务描述

使用 MPI 的 Send/Recv 原语实现一个 In-place 的 AllReduce 算子，即要求算子的输入和输出存储在同一块空间，并撰写报告。

- 基于数院集群，最多使用 2 个节点，各 8 个 CPU 核心；
- 要求使用 MPI 的 Send/Recv 原语实现至少**两种** AllReduce 的算法；
- 阻塞、非阻塞的 Send/Recv 接口都可以使用，MPI_Sendrecv 也可以使用；
- 要求完成正确性验证和提供测试脚本，注意正确性验证不应假设进程数和数据长度是任何数的整数倍；

- 要求在报告中给出 AllReduce 实现算法的通信拓扑，并从理论上分析其优劣，例如算法时间延迟、带宽瓶颈等等，讨论其适用的场景；
- 鼓励提供更多的实现，鼓励在不同数据量大小的情况下进行实验，并分析讨论其表现的差异。

注意：为了方便正确性检测，同学们可以将作业代码 allreduce.c 复制若干份，分别在不同的 c 源文件中实现不同的算法。如有其他疑问请咨询课程助教。

3 提交要求与评分标准

将代码和文档打包后上传至 http://mu2.davidandjack.cn:8888/ ，压缩包命名为 “学号-hw1”，在 **11 月 30 日 23 点 59 分**前提交。每位同学提交次数限制为 3 次。

评分标准：（1）基本要求：至少完成两种 AllReduce 算法，代码运行结果正确，在报告中完成相应的理论分析；（2）主要加分项：实现更多的 AllReduce 算法，在不同数据大小下进行实验测试，进行不同算法间的比较分析并讨论适用场景；（3）其他加分项：根据文档内容丰富程度、代码质量酌情加分。

参考文献

[1] A. Gibiansky, “Bringing hpc techniques to deep learning,” Baidu Research, Tech. Rep., 2017.

[2] A. Sergeev and M. Del Balso, “Horovod: fast and easy distributed deep learning in tensorflow,” arXiv preprint arXiv:1802.05799, 2018.

[3] Z. Cai, Z. Liu, S. Maleki, M. Musuvathi, T. Mytkowicz, J. Nelson, and O. Saarikivi, “Synthesizing optimal collective algorithms,” in Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 2021, pp. 62–75.