高性能计算程序设计（1） 秋季2024

**提交格式说明**

按照实验报告模板填写报告，需要提供源代码及代码描述至https://easyhpc.net/course/212。实验报告模板使用PDF格式，命名方式为高性能计算程序设计\_学号\_姓名。如果有问题，请发邮件至[zhudp3@mail2.sysu.edu.cn、liux276@mail2.sysu.edu.cn](mailto:zhudp3@mail2.sysu.edu.cn、liux276@mail2.sysu.edu.cn)询问细节。

1. **通过MPI实现通用矩阵乘法**

通过MPI点对点通信的方式实现通用矩阵乘法（Lab1），MPI并行进程（rank size）从1增加至8，矩阵规模从512增加至2048.

通用矩阵乘法（GEMM）通常定义为：

输入：M , N, K三个整数（512 ~2048）

问题描述：随机生成M\*N和N\*K的两个矩阵A,B,对这两个矩阵做乘法得到矩阵C.

输出：A,B,C三个矩阵以及矩阵计算的时间

1. **基于MPI的通用矩阵乘法优化**

分别采用MPI点对点通信和MPI集合通信实现矩阵乘法中的进程之间通信，并比较两种实现方式的性能。如有余力，可以进一步尝试用mpi\_type\_create\_struct聚合MPI进程内变量后通信。

1. **将 “实验0” 改造成矩阵乘法库函数**

将Lab0的单进程矩阵乘法改造为一个标准的库函数 matrix\_multiply（函数实现文件和函数头文件），输入参数为三个完整定义矩阵（A,B,C），定义方式没有具体要求，可以是二维矩阵，也可以是struct等。在Linux系统中将此函数编译为.so文件，由其他程序调用。

1. **构造MPI版本矩阵乘法加速比和并行效率表**

参考下图，分别构造MPI版本的标准矩阵乘法和优化后矩阵乘法的加速比和并行效率表格。并分类讨论两种矩阵乘法分别在强扩展和弱扩展情况下的扩展性。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Comm\_size (num of processes) | Order of Matrix (Speedups, milliseconds) | | | | |
| 128 | 256 | 512 | 1024 | 2048 |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 16 |  |  |  |  |  |