



中山大學

SUN YAT-SEN UNIVERSITY

并程序设计与算法实验

Lab2-基于 MPI 的并行矩阵乘法 (进阶)

姓名 林隽哲

学号 21312450

学院 计算机学院

专业 计算机科学与技术

2025 年 4 月 2 日

1 实验目的

- 掌握 MPI 集合通信在并行矩阵乘法中的应用
- 学习使用 MPI_Type_create_struct 创建派生数据类型
- 分析不同通信方式和任务划分对并行性能的影响
- 研究并行程序的扩展性和性能优化方法

2 实验内容

- 使用 MPI 集合通信实现并行矩阵乘法
- 使用 MPI_Type_create_struct 聚合进程内变量后通信

3 实验结果

3.1 性能分析

根据运行结果，填入下表以记录不同线程数量和矩阵规模下的运行时间：

表 1: 用 MPI 集合通信实现

进程数	矩阵规模				
	128	256	512	1024	2048
1	0.009239	0.072149	0.670923	6.924118	54.375005
2	0.004372	0.044670	0.344490	3.466872	30.935780
4	0.002561	0.029424	0.216383	2.023358	18.362430
8	0.034996	0.019056	0.190298	1.327365	11.270439
16	0.050508	0.010465	0.131064	1.200326	9.370801

表 2: 用 MPI_Type_create_struct 聚合进程内变量后通信

进程数	矩阵规模				
	128	256	512	1024	2048
1	0.007141	0.059567	0.532740	5.003699	52.291290
2	0.003733	0.029872	0.280494	2.591877	26.501993
4	0.098602	0.120185	0.144042	1.723110	15.240982
8	0.027828	0.057403	0.231252	1.130620	10.078613
16	0.080558	0.122373	0.127154	1.051537	8.649403

4 实验分析

根据运行时间，分析程序并行性能及扩展性：

4.1 MPI 集合通信实现

表 3: MPI 集合通信实现的加速比

进程数	矩阵规模				
	128	256	512	1024	2048
1	1.00	1.00	1.00	1.00	1.00
2	2.11	1.61	1.95	2.00	1.76
4	3.61	2.45	3.10	3.42	2.96
8	0.26	3.79	3.53	5.22	4.83
16	0.18	6.89	5.12	5.77	5.80

- 小规模矩阵（128 256）：
 - 进程数为 2 和 4 时，加速比随进程数增加而提高
 - 进程数为 8 和 16 时，由于通信开销增加，加速比反而下降
 - 256×256 矩阵在 16 进程时达到最高加速比 6.89
- 中大规模矩阵（512 2048）：
 - 随着矩阵规模增大，加速比整体呈上升趋势
 - 2048×2048 矩阵在 16 进程时达到最高加速比 5.80
 - 进程数增加带来的性能提升逐渐趋于稳定

4.2 MPI_Type_create_struct 实现

表 4: MPI_Type_create_struct 实现的加速比

进程数	矩阵规模				
	128	256	512	1024	2048
1	1.00	1.00	1.00	1.00	1.00
2	1.91	1.99	1.90	1.93	1.97
4	0.07	0.50	3.70	2.90	3.43
8	0.26	1.04	2.30	4.42	5.19
16	0.09	0.49	4.19	4.76	6.05

- 小规模矩阵 (128 256):
 - 进程数为 2 时获得较好的加速比 (1.91 1.99)
 - 进程数超过 2 后, 由于结构体通信开销, 加速比显著下降
 - 256×256 矩阵在 2 进程时达到最高加速比 1.99
- 中大规模矩阵 (512 2048):
 - 随着矩阵规模增大, 结构体通信的开销影响逐渐减小
 - 512×512 矩阵在 16 进程时达到最高加速比 4.19
 - 2048×2048 矩阵在 16 进程时达到最高加速比 6.05

4.3 扩展性分析

- MPI 集合通信实现:
 - 使用 Bcast/Scatter/Gather 进行数据分发和收集
 - 在小规模矩阵下, 扩展性能随进程数增加而下降
 - 在大规模矩阵下表现出良好的扩展性
- MPI_Type_create_struct 实现:
 - 使用 Bcast/Scatterv/Gather 进行数据分发和收集
 - 在小规模矩阵下, 扩展性能同样随进程数增加而下降
 - 在大规模矩阵下同样表现出良好的扩展性