



中山大學

SUN YAT-SEN UNIVERSITY

并程序设计与算法实验

Lab3-Pthreads 并行矩阵乘法与数组求和

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 9 日

1 实验目的

- Pthreads 程序编写、运行与调试
- 多线程并行矩阵乘法
- 多线程并行数组求和

2 实验内容

- 掌握 Pthreads 编程的基本流程
- 理解线程间通信与资源共享机制
- 通过性能分析明确线程数、数据规模与加速比的关系

2.1 并行矩阵乘法

- 使用 Pthreads 实现并行矩阵乘法
- 随机生成 $m \times n$ 的矩阵 A 及 $n \times k$ 的矩阵 B
- 通过多线程计算矩阵乘积 $C = A \times B$
- 调整线程数量 (1-16) 和矩阵规模 (128-2048)，记录计算时间
- 分析并行性能 (时间、效率、可扩展性)

2.2 并行数组求和

- 使用 Pthreads 实现并行数组求和
- 随机生成长度为 n 的整型数组 A ， n 取值范围 [1M, 128M]
- 通过多线程计算数组元素和 $s = \sum_{i=1}^n A_i$
- 调整线程数量 (1-16) 和数组规模 (1M-128M)，记录计算时间
- 分析并行性能 (时间、效率、可扩展性)

3 实验结果

3.1 并行矩阵乘法

表 1: 并行矩阵乘法在不同线程数下的运行时间

矩阵规模	1 线程	2 线程	4 线程	8 线程	16 线程
128×128	0.010	0.006	0.007	0.011	0.010
256×256	0.058	0.032	0.026	0.021	0.024
512×512	0.567	0.281	0.168	0.119	0.124
1024×1024	5.671	2.902	1.602	1.011	0.951
2048×2048	82.575	43.418	21.989	14.054	13.390

3.2 并行数组求和

表 2: 数组求和不同线程数下的运行时间

数组规模	1 线程	2 线程	4 线程	8 线程	16 线程
1M	0.002	0.002	0.005	0.005	0.011
4M	0.009	0.025	0.010	0.013	0.013
16M	0.033	0.044	0.058	0.032	0.034
64M	0.149	0.322	0.203	0.109	0.103
128M	0.255	0.320	0.456	0.326	0.151

4 实验分析

4.1 并行矩阵乘法

- 线程数量对性能的影响分析：
 - 对于小规模矩阵，多线程带来的性能提升不明显，甚至有时会出现性能下降，这是因为线程创建和管理的开销超过了并行计算带来的收益。
 - 对于大规模矩阵，多线程的优势更加明显，可以从 2048×2048 的矩阵乘法的结果中看出，单线程需要 82.575 秒，而使用 16 个线程时仅需 13.390 秒，加速比约为 6.17。
- 矩阵规模对并行效率的影响：

- 当矩阵规模较小时，线程间的通信开销和线程管理开销占比较大，导致并行效率较低。
- 矩阵规模越大，并行效率越高。对于 128×128 的矩阵，16 线程的加速比仅为 1.0，而对于 2048×2048 的矩阵，加速比达到 6.17。
- 可扩展性分析：
 - 从 2 线程到 16 线程，性能提升呈现递减趋势，说明系统存在一定的可扩展性限制。
 - 这种限制可能来自多个方面：内存带宽限制、CPU 缓存竞争、线程调度开销以及矩阵规模等。
 - 对于 2048×2048 的矩阵，从 8 线程增加到 16 线程时，性能提升仅为 5%，说明对于 2048×2048 的矩阵，16 线程已经接近性能瓶颈。如果继续增加线程数，线程调度开销将超过并行计算带来的收益。

4.2 并行数组求和

- 线程数量对性能的影响分析：
 - 对于小规模数组（1M），多线程反而导致性能下降，这是因为线程创建和同步的开销超过了并行计算带来的收益。
 - 对于大规模数组（128M），16 线程相比单线程有约 1.69 倍的加速比，但性能提升不如矩阵乘法明显。
 - 在某些情况下（如 64M 数组），2 线程的性能反而比 4 线程差，这可能是由于线程调度和内存访问模式的影响。
- 数组规模对并行效率的影响：
 - 数组规模越大，并行效率越高。对于 1M 的数组，多线程几乎没有性能提升，而对于 128M 的数组，16 线程有显著加速。
 - 当数组规模较小时，线程间的通信开销和线程管理开销占比较大，导致并行效率较低。
 - 大规模数组可以更好地分摊线程创建和同步的开销，使得并行计算更加高效。
- 可扩展性分析：
 - 数组求和的并行效率明显低于矩阵乘法，这是因为数组求和的计算密度较低，内存访问模式更简单。
 - 从 2 线程到 16 线程，性能提升呈现不稳定的趋势，说明其可扩展性较差。

- 这种限制主要来自内存带宽和缓存竞争，因为数组求和主要是内存密集型操作。