



中山大學

SUN YAT-SEN UNIVERSITY

并行程序设计 with 算法实验

Lab4-Pthreads 并行方程求解与蒙特卡洛

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 16 日

1 实验目的

- 深入理解 Pthreads 同步机制：条件变量、互斥锁
- 评估多线程加速性能

2 实验内容

- 多线程一元二次方程求解
- 多线程圆周率计算

2.1 方程求解

- 使用 Pthread 多线程及求根公式实现并行一元二次方程求解。
- 方程形式为 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ ，其中判别式的计算、求根公式的中间值分别由不同的线程完成。
- 通过条件变量识别何时线程完成了所需计算，并分析计算任务依赖关系及程序并行性能。

2.2 蒙特卡洛方法求圆周率

2.2.1 不同线程数量和随机点数量下的性能比较

蒙特卡洛方法估计圆周率是一种典型的可并行化问题。通过在单位正方形内随机生成点，计算落在单位圆内的点的比例，可以估计 π 值。本实验测试了不同线程数量和随机点数量下算法的性能和精度。

表 1: 执行时间（秒）随线程数和点数的变化趋势

点数 / 线程数	1 线程	4 线程	8 线程	16 线程
10,000	0.000515	0.000607	0.000518	0.001028
20,000	0.000509	0.000000*	0.000523	0.000657
40,000	0.000521	0.000526	0.000525	0.000511
65,536	0.000677	0.000658	0.000524	0.001629

^{0*} 注：20,000 点和 4 线程的时间为 0.000000，可能是计时精度限制或测量误差。

表 2: π 的近似值随线程数和点数的变化趋势

点数 / 线程数	1 线程	4 线程	8 线程	16 线程
10,000	3.1428000000	3.1204000000	3.1312000000	3.1380000000
20,000	3.1444000000	3.1498000000	3.1624000000	3.1680000000
40,000	3.1391000000	3.1410000000	3.1299000000	3.1419000000
65,536	3.1423339844	3.1420898438	3.1372070312	3.1404418945

表 3: 与标准 π 的差异随线程数和点数的变化趋势（误差）

点数 / 线程数	1 线程	4 线程	8 线程	16 线程
10,000	0.0012073464	0.0211926536	0.0103926536	0.0035926536
20,000	0.0028073464	0.0082073464	0.0208073464	0.0264073464
40,000	0.0024926536	0.0005926536	0.0116926536	0.0003073464
65,536	0.0007413308	0.0004971902	0.0043856223	0.0011507591

2.2.2 实验结果分析

比较不同线程数量和随机点数量下圆周率估计的准确性和计算速度： 从表格中可以观察到，随着点数的增加，圆周率估计的准确性总体上有所提高。例如，在单线程情况下，当点数从 10,000 增加到 65,536 时，误差从 0.0012073464 减小到 0.0007413308。最高精度出现在 40,000 点和 16 线程的组合，误差仅为 0.0003073464。

在计算速度方面，不同配置的执行时间差异不明显，大多数情况下在 0.0005-0.0008 秒范围内。这表明在当前实验规模下，计算开销较小，线程创建和管理的开销可能占据主要部分。

增加线程数量对计算效率和精度的影响： 从实验数据来看，增加线程数量并不总是能提高计算效率。例如：

- 在 10,000 点测试中，从 1 线程到 4 线程，执行时间从 0.000515 秒增加到 0.000607 秒
- 在 65,536 点测试中，16 线程配置的执行时间（0.001629 秒）明显高于其他线程配置

这种现象可能是由以下因素导致的：

- 线程创建和管理的开销在小规模计算中占比较大
- 线程间同步（互斥锁）产生的额外开销
- 当前问题规模较小，并行化带来的收益被线程管理开销抵消

线程数量对精度的影响不一致，在某些情况下（如 40,000 点，16 线程）提供了最高精度，而在其他情况下（如 20,000 点，16 线程）反而产生了较大误差。这可能与随机数生成在多线程环境中的分布特性有关。

计算效率和精度趋势：

- 精度趋势：总体而言，随着点数增加， π 估计的精度通常会提高，但这种提升不是线性的。
- 效率趋势：在实验范围内，线程数增加对计算时间没有显著改善，某些情况下甚至造成性能下降。
- 最佳配置：从精度角度看，40,000 点配合 16 线程提供了最佳结果（误差 0.0003073464）；从效率与精度平衡的角度看，65,536 点配合 4 线程可能是较好选择（时间 0.000658 秒，误差 0.0004971902）。

实验过程中遇到的问题及解决策略：

- 同步问题：使用互斥锁保护全局计数器是必要的，但在高线程数下可能成为性能瓶颈。改进策略包括：使用原子操作替代互斥锁；减少同步频率，采用批量更新方式；或实现无锁算法。
- 负载均衡问题：当前实现通过处理余数（最后一个线程处理额外的点）保证了工作负载的基本均衡。更好的策略可能是动态负载均衡或采用工作窃取（work stealing）机制。
- 线程开销问题：对于小规模问题，线程创建和管理的开销可能超过并行计算带来的收益。使用线程池或 OpenMP 等更高级别的抽象可能比直接管理 pthread 更有效率。
- 随机数质量问题：当前实现使用简单的线性同余随机数生成器，并通过为每个线程分配不同种子来避免冲突。使用更高质量的随机数生成器（如 Mersenne Twister）可能提高估计精度。
- 计时精度问题：实验中观察到的计时结果波动较大，某些情况下甚至出现 0 时间。使用更高精度的计时方法或增加重复测试次数取平均值可以提高测量准确性。

通过这些改进，可以进一步优化蒙特卡洛方法在并行环境下的性能和精度，特别是在更大规模的点数和更复杂的计算环境中。