**课程实验报告**

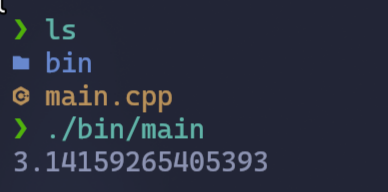
**题目：并行计算π深入讨论**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **班级** | 辅修 | **学号** | 202226910925 | **姓名** | 杨俊科 |

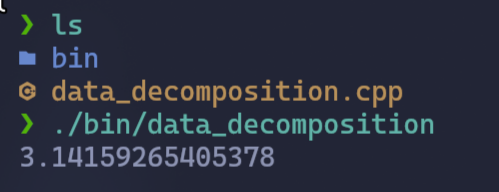
**实验内容：**

1. 在并行计算π作业中，观察单线程（串行）实现和多线程（2线程或以上）实现过程中，最终计算出来π的结果是否一致？请解释此现象。（首先可运行程序并截图，展现出以上两者的结果，证明是有不一样。然后利用自己的知识，解释此现象，可图文并茂进行解释。不少于200字）

串行：



多线程：



两者结果不一致，可能是因为浮点数计算在不同线程中可能会因为累积顺序不同而产生细微的精度误差。串行计算时，所有计算按照严格的顺序执行，而并行计算可能会在多个线程中同时进行，最终结果会因为不同顺序累积而产生差异。这种现象在使用浮点数时尤其明显，因为浮点数的表示是近似的，尤其是在多次累加时，误差会逐渐放大。至于线程间竞争的问题，由于设置了互斥锁，故而影响因素不大，主要归于于浮点数精度问题。

1. 在并行计算π作业中，如果使用数据分解把数轴分为两个数据块：[1,Int.Max/2)，[Int.Max/2, Int.Max)。请在两个数据块并行计算中，加入计时器（即循环开始前时刻，循环结束后时刻，当前线程运行时间为以上两个时刻之差），对比两个数据块所需运行时刻，将会出现前一个数据块运行时间比后一个数据块要多。

问题（1）：请解释此现象

问题（2）：请给出解决方案，并提交优化后的实现代码

在计算π时，前半段数据块的运行时间通常较长，这主要是由于数据密集度、CPU缓存与分支预测以及浮点运算的累积误差。较小的数据块涉及更多有效计算操作，且由于每一项对最终结果的贡献逐渐减小，处理小数字时需要更高的计算精度。此外，现代处理器依赖缓存和分支预测，前半段数据块需要更多内存访问，导致性能下降。相对而言，后期较大的数值变化对结果影响较小，计算效率更高。最后，浮点运算在小范围内的精度要求高，导致计算耗时增加，而后半段的误差容忍度更高，从而使计算速度加快。