**中山大学计算机院本科生实验报告**

**（2024学年春季学期）**

**课程名称：并行程序设计 批改人：**

|  |  |  |  |
| --- | --- | --- | --- |
| **实验** | **10-CUDA并行矩阵乘法** | **专业（方向）** | **计算机科学与技术** |
| **学号** | **21307174** | **姓名** | **刘俊杰** |
| **Email** | **liujj255@mail2.sysu.edu.cn** | **完成日期** | **2024/5/29** |

# **实验目的**

**CUDA实现并行通用矩阵乘法，并通过实验分析不同线程块大小，访存方式、数据/任务划分方式对并行性能的影响。**

**输入：三个整数，每个整数的取值范围均为[128, 2048]**

**问题描述：随机生成的矩阵及的矩阵，并对这两个矩阵进行矩阵乘法运算，得到矩阵.**

**输出：三个矩阵，及矩阵计算所消耗的时间。**

**要求：使用CUDA实现并行矩阵乘法，分析不同线程块大小，矩阵规模，访存方式，任务/数据划分方式，对程序性能的影响。**

# **实验过程和核心代码**

**2.1 全局内存 每个线程负责计算C中一个位置的值**

## **1**

**2.1.1 实验思路**

**①首先根据blockID、blockDim和threadIDx计算出每个线程在全局上的索引,同时也是其负责计算C位置上的值的索引。**

**②检查负责的索引是否在矩阵C的范围内。**

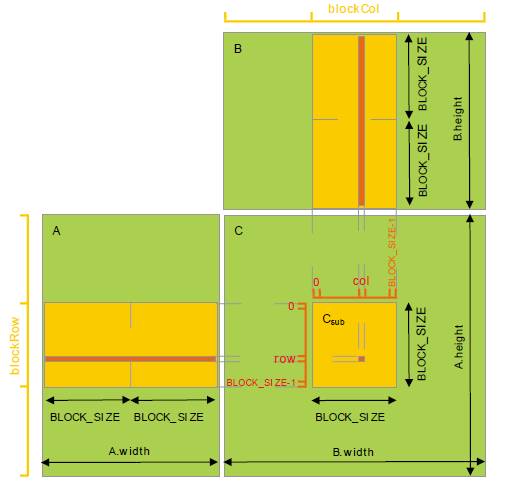
**③若在,则该线程计算出该位置的值,并赋值给C对应的位置。**

**2.1.2 核心代码**

**核心代码如下:**

## 

**2.2 共享内存**

****

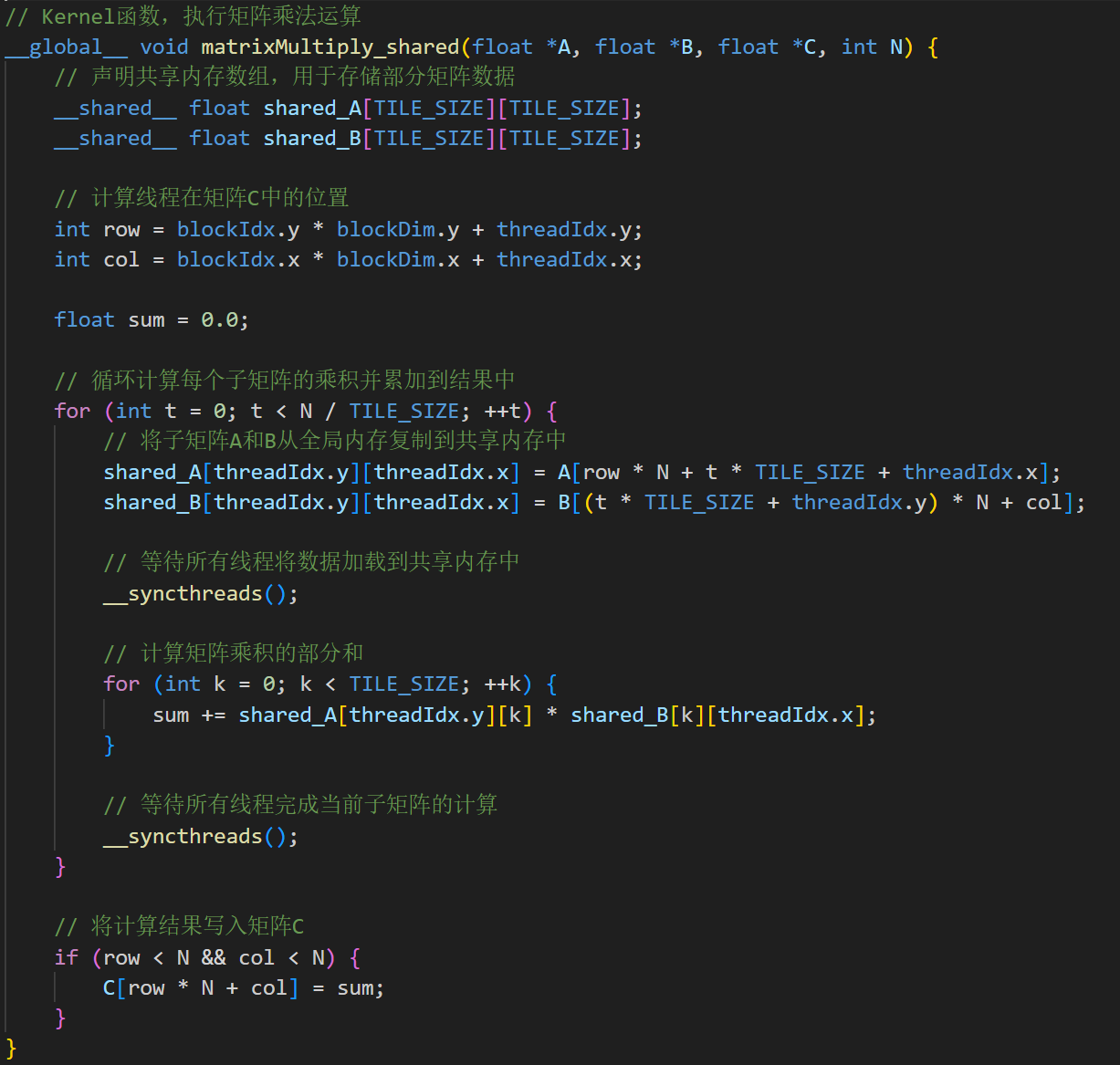
**2.2.1 实验思路**

**使用共享内存优化的基本思想是利用数据的局部性，通过将数据从全局内存读取到共享内存中，减少每次访问全局内存所需的延迟时间。**

**在矩阵乘法中，可以将每个线程块所需要的矩阵子块数据（例如 BLOCK\_SIZE × BLOCK\_SIZE 大小的子矩阵）从全局内存中读取到共享内存中。然后，每个线程在计算时可以从共享内存中读取数据，避免了每次计算都需要从全局内存中读取数据的高延迟。**

**具体实现时，可以将矩阵 A 的子块在行向上滑动，将矩阵 B 的子块在列向上滑动，以便每个线程块中的每个线程都能够计算完所有元素的乘累加。这样，每个线程块中的线程都可以重复利用共享内存中的数据，避免了频繁访问全局内存的开销。**

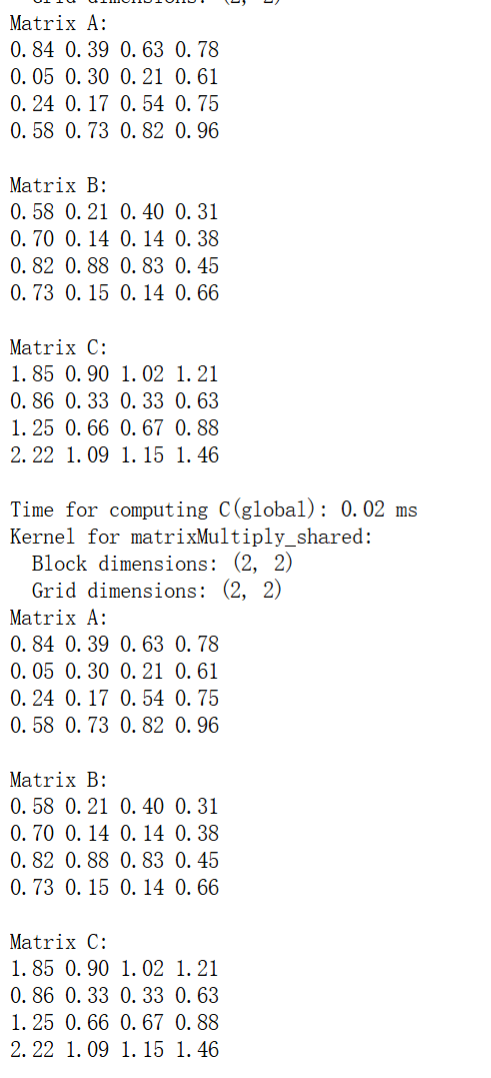
**2.2.2 核心代码**

****

# **实验结果**

**3.1 验证实验正确性**

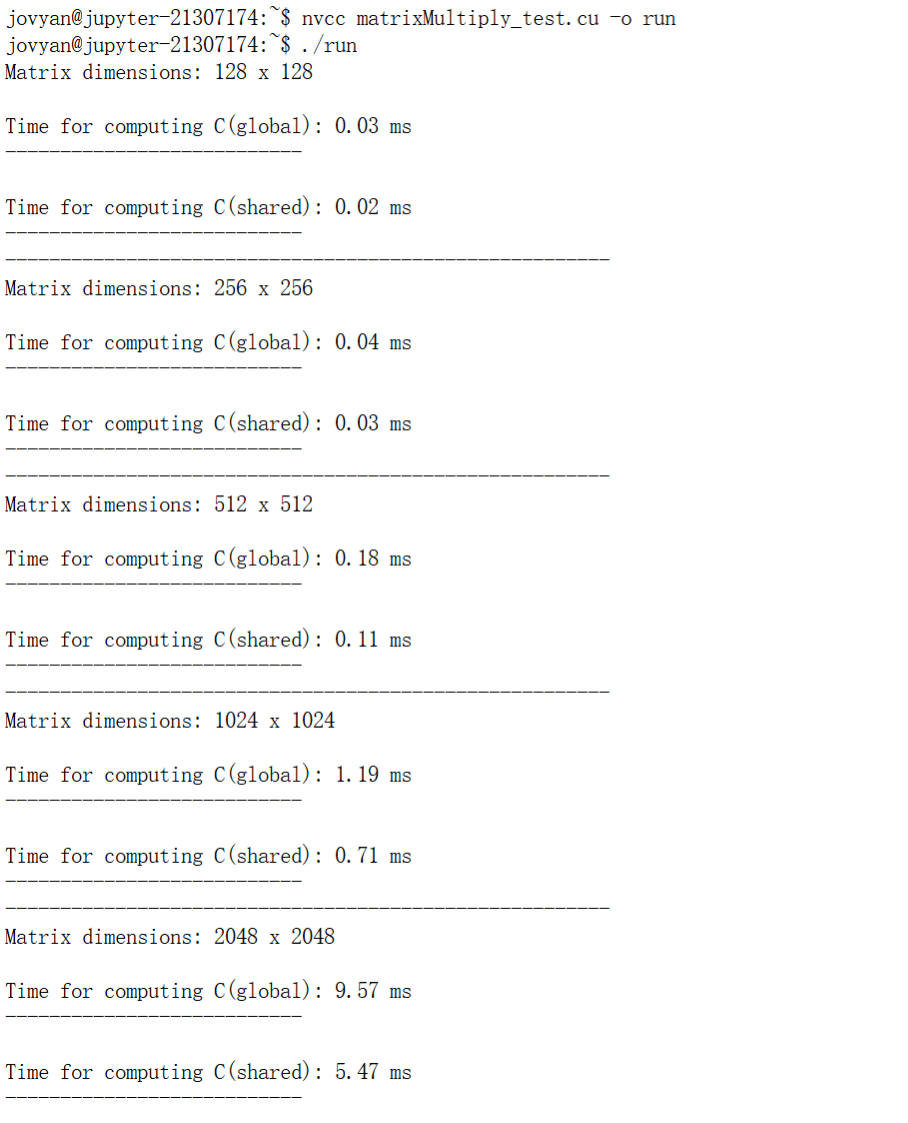
## **运行结果:**

****

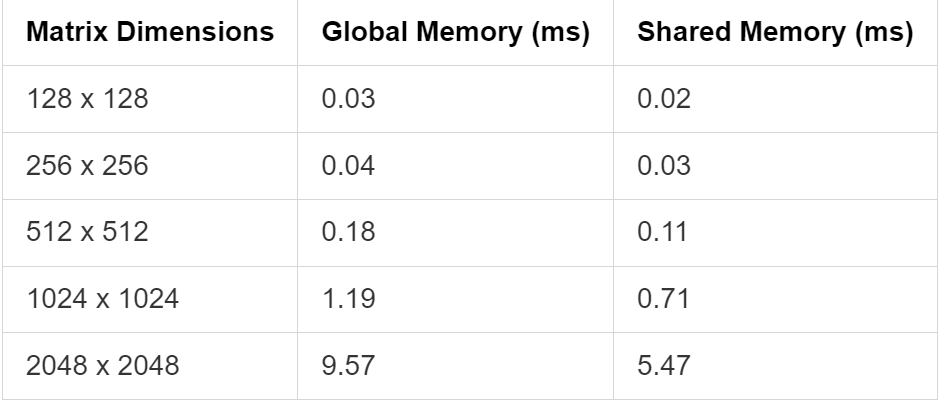
**可以看到使用共享内存和全局内存计算的结果都是正确的。**

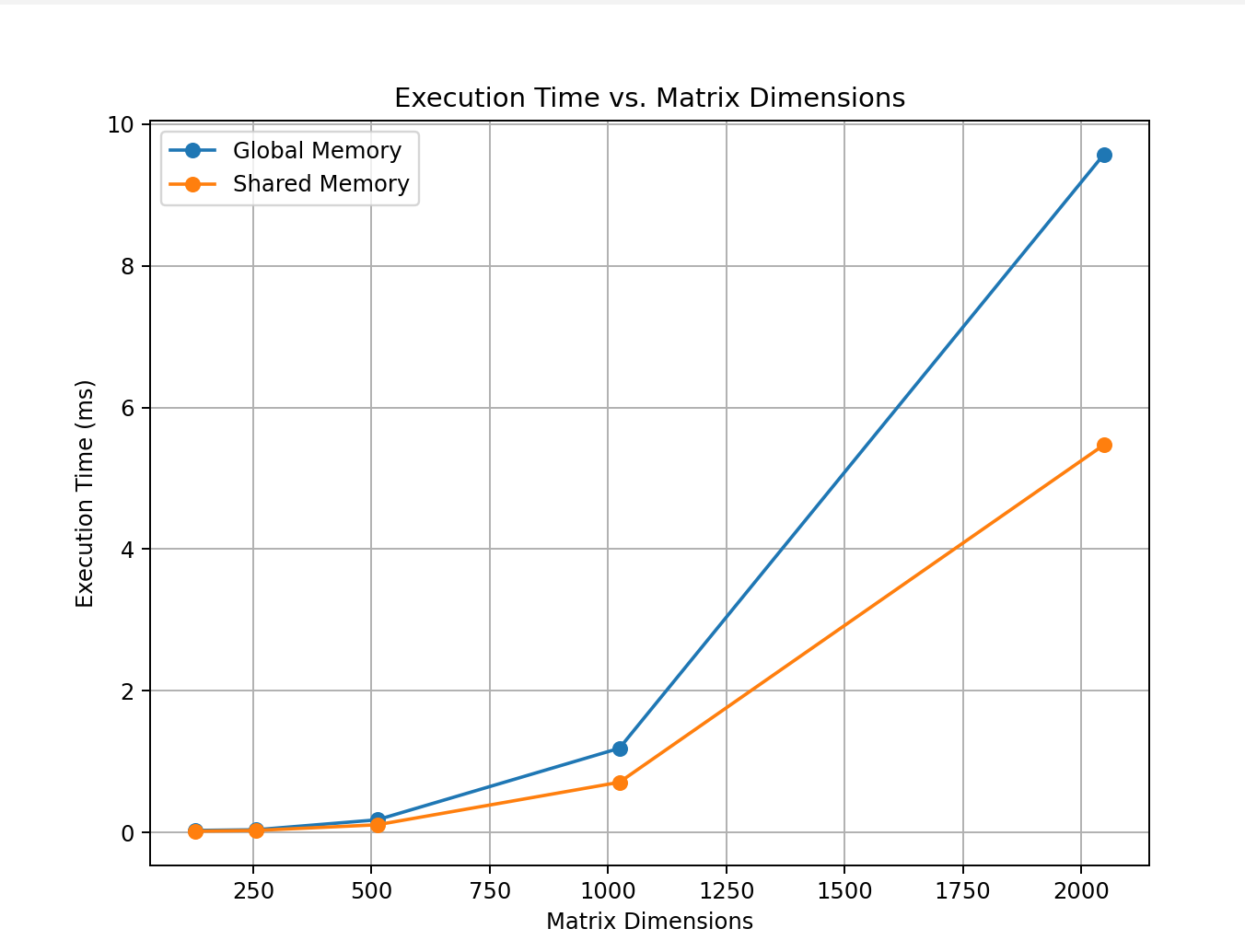
**3.2 比较共享内存与全局内存对程序性能的影响(16 X 16的线程块大小)**

**运行结果:**

****

**对比结果:**

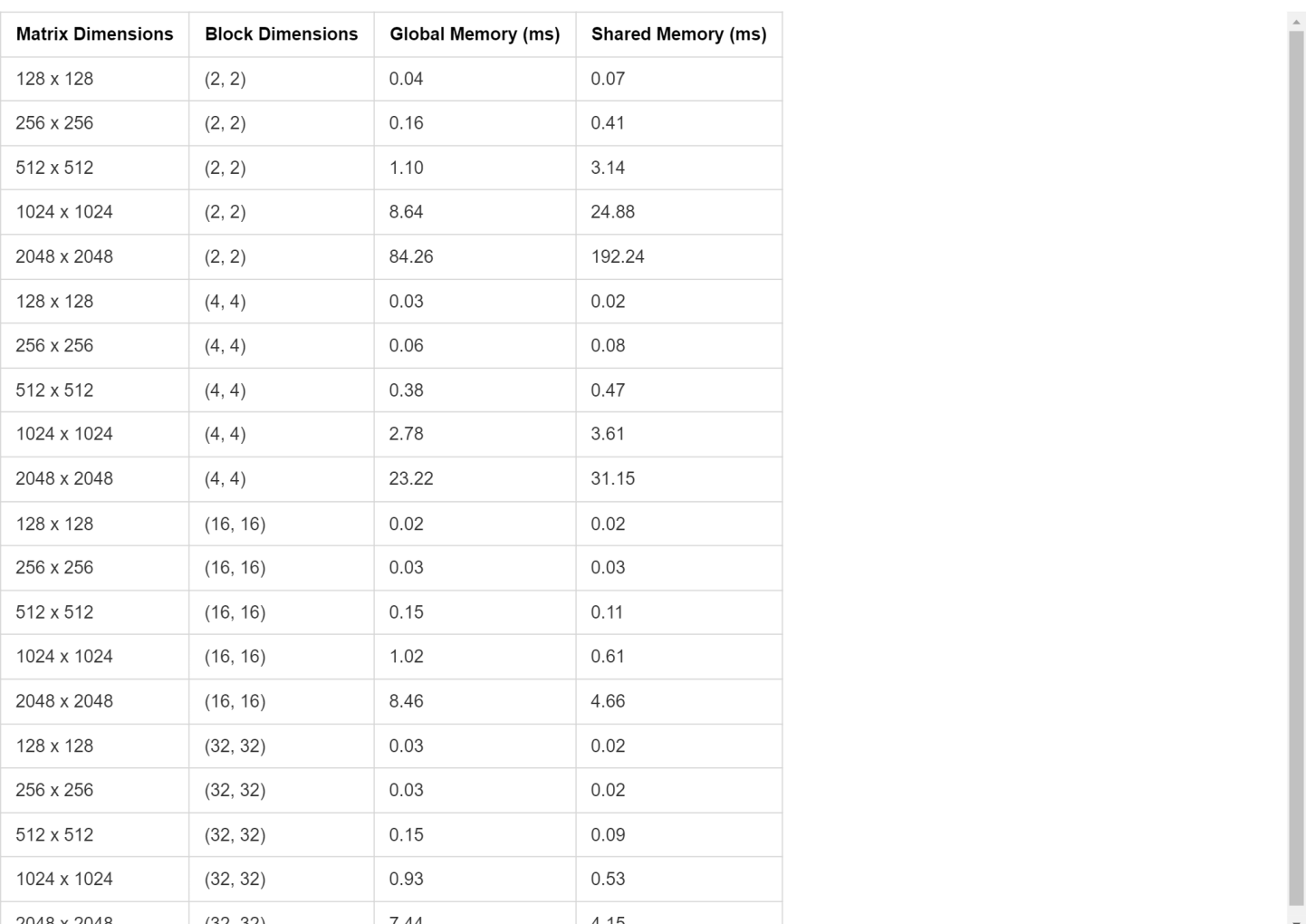
****

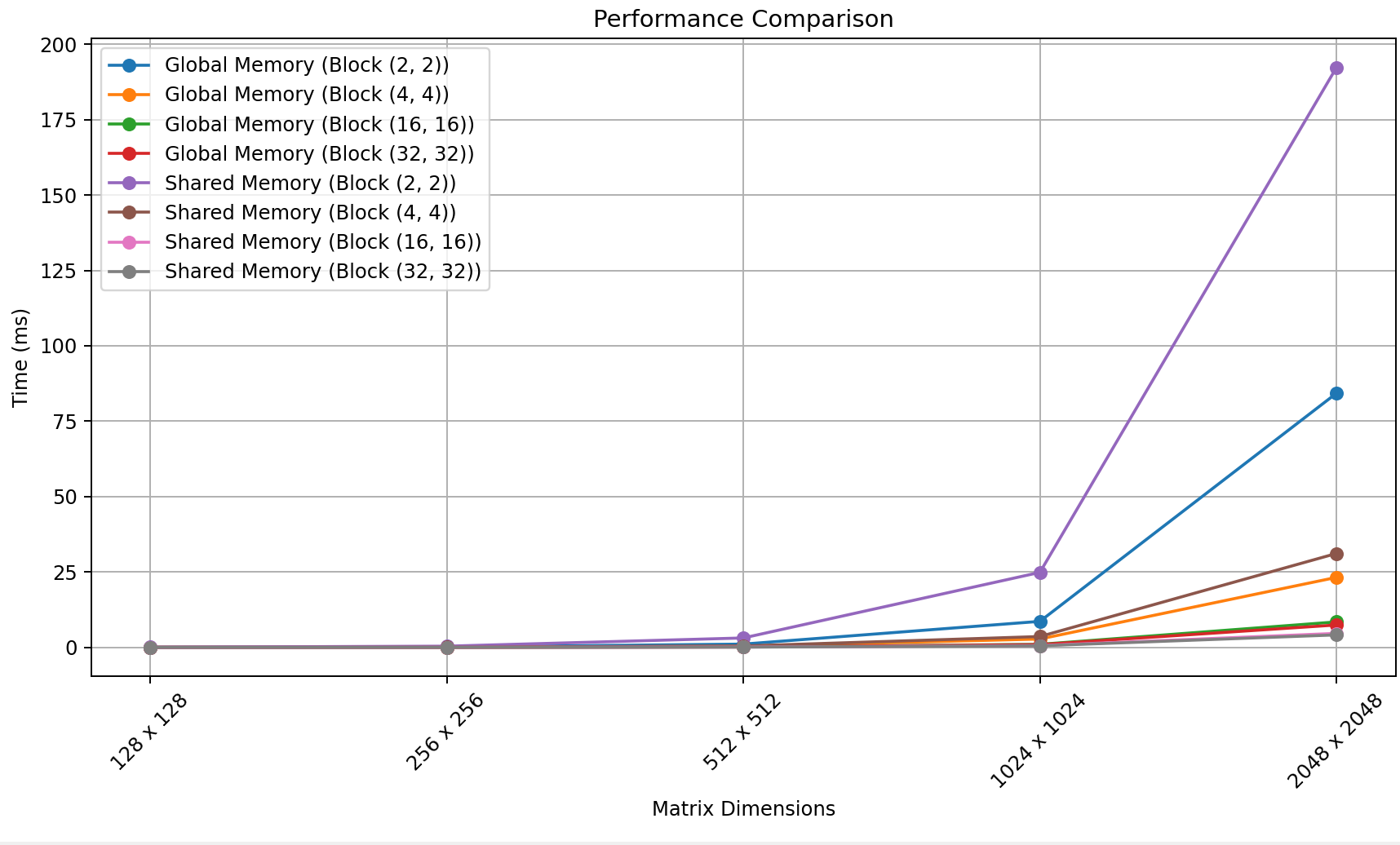
****

**可以从实验结果中看到,使用共享内存的运行时间小于使用全局内存的运行时间,同时随着矩阵规模的增大,共享内存的加速效果越明显。**

**3.3 不同大小的线程块对程序性能的影响**

**实验结果对比:**

****

****

**可以看到使用不同的线程块大小对程序性能的影响是显著的,使用过小的线程块，程序的运行时间长,甚至当线程块大小过小时，全局内存的效果优于共享内存;从实验结果总体上看来,线程块大小越大,对程序性能的加速结果越明显,但当线程块过大时加速的提升就不是很大了。**

# **实验感想**

**4.1 问题即解决方案**

## **在实验过程中,因为对threadIdx.x和threadIdx.y的理解错误导致矩阵乘法的计算结果一直不正确,这是因为误将C++中vector nums中nums[x][y]中的x、y的方向认为是与CUDA threadIIx.x和threadIdx.y是一致的。**

**而在 CUDA 编程中:**

**threadIdx.x 表示线程在其所属线程块中的 X 方向索引。**

**threadIdx.y 表示线程在其所属线程块中的 Y 方向索引。**

## **4.2 实验感悟**

**在实现并行通用矩阵乘法并进行实验分析的过程中，我深刻体会到了CUDA并行计算的强大性能和灵活性。通过调整线程块大小、矩阵规模、访存方式，我发现这些因素对程序性能有着显著的影响。**

**首先，在调整线程块大小方面，我发现合适的线程块大小能够充分利用GPU的并行计算能力，提高计算效率。**

**其次，在矩阵规模方面，矩阵大小的选择会直接影响到并行计算的效率。较大的矩阵规模能够充分利用GPU的并行计算资源，提高计算效率，但也会增加访存和通信的开销。**

**再者，在访存方式方面，合理的内存访问模式能够减少内存访问延迟，提高计算效率。使用共享内存可以减少全局内存的访问次数，加速矩阵乘法运算。而优化的存储布局和内存访问模式也能够提高数据的局部性，减少内存访问冲突。**

**通过实验分析不同因素对并行性能的影响，我对CUDA并行计算的原理和优化技巧有了更深入的理解，也对如何优化并行程序性能有了更清晰的认识。在今后的工作中，我将继续探索并行计算领域，不断优化并行程序，提高计算效率。**