



中山大學

SUN YAT-SEN UNIVERSITY

## 并程序设计与算法实验

### Lab9-CUDA 矩阵转置

姓名 \_\_\_\_\_ XXX

学号 \_\_\_\_\_ XXX

学院 \_\_\_\_\_ 计算机学院

专业 \_\_\_\_\_ XXX

2025 年 5 月 21 日

## 1 实验目的

- 熟悉 CUDA 线程层次结构 (grid、block、thread) 和观察 warp 调度行为。
- 掌握 CUDA 内存优化技术 (共享内存、合并访问)。
- 理解线程块配置对性能的影响。

## 2 实验内容

### 2.1 CUDA 并行输出

1. 创建  $n$  个线程块，每个线程块的维度为  $m \times k$ 。
2. 每个线程均输出线程块编号、二维块内线程编号。例如：
  - “Hello World from Thread (1, 2) in Block 10!”
  - 主线程输出 “Hello World from the host!”。
  - 在 main 函数结束前，调用 `cudaDeviceSynchronize()`。
3. 完成上述内容，观察输出，并回答线程输出顺序是否有规律。

### 2.2 使用 CUDA 实现矩阵转置及优化

1. 使用 CUDA 完成并行矩阵转置。
2. 随机生成  $N \times N$  的矩阵  $A$ 。
3. 对其进行转置得到  $A^T$ 。
4. 分析不同线程块大小、矩阵规模、访存方式 (全局内存访问，共享内存访问)、任务/数据划分和映射方式，对程序性能的影响。
5. 实现并对比以下两种矩阵转置方法：
  - 仅使用全局内存的 CUDA 矩阵转置。
  - 使用共享内存的 CUDA 矩阵转置，并考虑优化存储体冲突。

## 3 实验结果与分析

### 3.1 CUDA Hello World 并行输出

#### 3.1.1 实验现象

描述实验观察到的现象，例如线程输出的顺序等。可以粘贴部分关键的运行截图或输出文本。

- 回答：

#### 3.1.2 结果分析

线程输出顺序是否有规律？为什么？结合 CUDA 线程调度机制进行解释。

- 回答：

### 3.2 CUDA 矩阵转置及优化

#### 3.2.1 不同实现方法的性能对比

1. 展示不同矩阵转置实现（仅全局内存、使用共享内存、优化共享内存访问）在不同矩阵规模 (N) 和不同线程块大小下的运行时间。可以根据你的实验设置更改表格的矩阵规模、线程块大小。

表 1: 矩阵转置性能对比 (时间单位: ms)

矩阵规模 (N)	线程块大小	全局内存版本	共享内存版本	优化共享内存版本
512	8×8			
	16×16			
	32×32			
1024	8×8			
	16×16			
	32×32			
2048	8×8			
	16×16			
	32×32			

#### 3.2.2 结果分析

1. 根据实验结果，总结线程块大小、矩阵规模对程序性能的影响。哪种配置下性能最优？为什么？

回答：

2. 讨论任务/数据划分和映射方式对性能的影响。

回答：

注：实验报告格式参考本模板，可在此基础上进行修改；实验代码以 zip 格式另提交；最终提交内容包括实验报告 (pdf 格式) 和实验代码 (zip 压缩包格式)