

# 实验二：全源最短路

## 实现方法

在我的实现中，BLOCK\_SIZE 设为 32，BATCH\_SIZE\_PHASE2 和 BATCH\_SIZE\_PHASE3 均设为 6，使用了分块的思想来优化 Floyd-Warshall 算法。具体来说，首先将图邻接矩阵划分为多个大小为 BLOCK\_SIZE × BLOCK\_SIZE 的块，每轮迭代处理第 k 个对角块（pivot block）及其相关行列块（phase 1 & 2），然后处理剩余非对角块（phase 3）。在每个阶段中，使用共享内存来存储当前的最短路径信息，以减少全局内存访问的次数。

相比于助教提供的基础实现，我的实现引入了两级分块机制：将若干个 BLOCK\_SIZE × BLOCK\_SIZE 的小块再组合成大块，每次处理多个小块，批量加载到共享内存中。这种方法进一步压缩了线程块的数量，提高了吞吐率。

整个算法分为三个阶段：

- Phase 1（对角块处理）：只使用一个线程块，负责处理第 k 个轮次中的对角块 B(k, k)，即图对角线上某个 BLOCK\_SIZE × BLOCK\_SIZE 子矩阵。该阶段只需一个线程块读取该子矩阵进共享内存，在本地更新最短路径。
- Phase 2（行块和列块处理）：使用  $\lceil \frac{n}{BATCH\_SIZE\_PHASE2 \times BLOCK\_SIZE} \rceil \times 2$  个线程块，分别对应对角块所在的整行和整列。每个线程块处理 BATCH\_SIZE\_PHASE2 个 BLOCK\_SIZE × BLOCK\_SIZE 小块组成的大块（横向或纵向），将对角块与相关子块加载到共享内存中，由 BLOCK\_SIZE × BLOCK\_SIZE 个线程分别更新自己负责的块元素。批量处理可以在共享内存中重用对角块数据，从而减少重复读写。
- Phase 3（非对角块处理）：处理除对角块及其所在行列以外的所有块。使用  $\lceil \frac{n}{BATCH\_SIZE\_PHASE3 \times BLOCK\_SIZE} \rceil \times \lceil \frac{n}{BATCH\_SIZE\_PHASE3 \times BLOCK\_SIZE} \rceil$  个线程块，每个线程块处理一个 BATCH\_SIZE\_PHASE3 × BATCH\_SIZE\_PHASE3 的小块矩阵。当 BATCH\_SIZE\_PHASE3 = 6 时，每个线程块加载的共享内存空间为 48KB，恰好填满一个 SM 的共享内存上限，利用率较高。

## 运行时间

首先根据助教给出的思路初步实现了基于分块思想的 Floyd-Warshall 算法优化方法，得到运行时间数据如下。

| $n$   | 运行时间（ms）    | 朴素实现运行时间（ms） | 加速比   |
|-------|-------------|--------------|-------|
| 1000  | 3.744969    | 15.442393    | 4.12  |
| 2500  | 34.254069   | 377.835219   | 11.03 |
| 5000  | 244.743859  | 2986.590461  | 12.20 |
| 7500  | 808.646755  | 10050.433585 | 12.43 |
| 10000 | 1893.072071 | 22837.625252 | 12.06 |

进行了一些寄存器优化，减少对共享内存的访问。

| $n$   | 运行时间 (ms)   | 朴素实现运行时间 (ms) | 加速比   |
|-------|-------------|---------------|-------|
| 1000  | 2.679856    | 15.442393     | 5.76  |
| 2500  | 22.591037   | 377.835219    | 16.73 |
| 5000  | 156.343829  | 2986.590461   | 19.10 |
| 7500  | 513.028327  | 10050.433585  | 19.59 |
| 10000 | 1192.183992 | 22837.625252  | 19.16 |

为了更充分地利用 Shared Memory，在第二阶段和第三阶段使用了让每个 block 处理 6×6 个块的方法。

| $n$   | 运行时间 (ms)  | 朴素实现运行时间 (ms) | 加速比   |
|-------|------------|---------------|-------|
| 1000  | 2.638259   | 15.442393     | 5.85  |
| 2500  | 18.822631  | 377.835219    | 20.07 |
| 5000  | 122.125867 | 2986.590461   | 24.46 |
| 7500  | 398.027312 | 10050.433585  | 25.25 |
| 10000 | 928.647928 | 22837.625252  | 24.59 |

进行进一步优化，在第三阶段分类讨论完全在边界内的块和部分在边界外的块，减少了条件判断的次数，同时针对循环中的变量进行了一些优化。

| $n$   | 运行时间 (ms)  | 朴素实现运行时间 (ms) | 加速比   |
|-------|------------|---------------|-------|
| 1000  | 1.930875   | 15.442393     | 8.00  |
| 2500  | 14.339964  | 377.835219    | 26.35 |
| 5000  | 88.366294  | 2986.590461   | 33.80 |
| 7500  | 285.393124 | 10050.433585  | 35.22 |
| 10000 | 660.559218 | 22837.625252  | 34.57 |