

Report

代码实现

代码中共实现了三个GPU Kernel函数，分别为 `center`，`cross`，`whole`。块的大小为 32×32 。

`center` 函数用于在第 p 轮的中心块运行Floyd-Warshall算法。该Kernel函数分配一个线程块，一个线程块中有 32×32 个线程。使用一个 32×32 个4-byte的共享内存来存储中心块的数据。在每一轮中，每个线程负责计算一个中心块的一个元素的Floyd-Warshall算法。

`cross` 函数用于在第 p 轮与第 p 个中心块同行或同列的块运行Floyd-Warshall算法（一个块多次做Floyd-Warshall不会改变最终的正确性）。该Kernel函数分配 $\lceil \frac{n}{32} \rceil \times 2$ 个线程块，每个线程块有 32×32 个线程。使用个 32×32 个4-byte的共享内存来存储分别中心块的数据和当前块的数据。在每一轮中，每一个线程块负责计算一个中心块与当前块的的Floyd-Warshall算法。

`whole` 函数用于在第 p 轮的所有块运行Floyd-Warshall算法。该Kernel函数分配 $\lceil \frac{n}{32 \times 6} \rceil \times \lceil \frac{n}{32 \times 6} \rceil$ 个线程块，每个线程块有 32×32 个线程。这是由于每个线程块支持 $48KB$ 的共享内存，故每个线程块内有两个 $6 \times 32 \times 32$ 的共享内存，分别记录在 `cross` 函数中已经计算的到的块中与当前块同行或同列的块的数据。由于当前线程块处理的块大小为 $(6 \times 32) \times (6 \times 32)$ ，因此需要记录两个 $6 \times 32 \times 32$ 的数据。

运行结果

以下为代码在 $n = 1000, 2500, 5000, 7500, 10000$ 时的运行结果：

n	运行时间 (ms)	朴素运行时间 (ms)	加速比
1000	1.616211	15.258736	9.441
2500	13.954964	377.812076	27.073
5000	88.775329	2986.758598	33.644
7500	295.208918	10051.515541	34.048
10000	652.662816	22908.779042	35.100