高性能计算导论 第六次小作业报告

管思源 2021012702

**一、不同的参数设置对 global memory 性能的影响**

以下是设置Stride分别为1、2、4、8时，程序测试GPU访问global memory的带宽：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stride | 1 | 2 | 4 | 8 |
| Bandwidth | 530.158 | 182.486 | 91.9932 | 46.2864 |
| Sector # | 4 | 8 | 16 | 32 |

观察发现，随访问数据间隔的变大，带宽也迅速下降。这是因为global memory存储在主存DRAM中，GPU访问这些数据需要先通过L1/L2 cache与DRAM之间传输，这一步的带宽是性能的主要限制因素。而这两者传输数据的最小单元是sector (32 Bytes)，当数据间隔越大，在访问同等数据量的情况下连续访问的sector个数就越多（如上表所示），实际传输的数据量越大，因此运行时间也就线性地增加。

此外，我们也观察到当Stride为1（即访问数据连续时），带宽得到了超线性的提升。这主要是硬件的合并访存功能起到了作用，它自动合并了warp中线程对连续地址的访存，节省了反复执行load指令的开销，进一步地降低了运行时间。

**二、不同的参数设置对 shared memory 性能的影响**

以下是分别改变数据位宽Bitwidth和数据间隔Stride时，程序测试GPU访问shared memory的带宽：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bandwidth | | Bitwidth | | |
| 2 | 4 | 8 |
| Stride | 1 | 4250.2 | 8515.11 | 8637.79 |
| 2 | 4084.01 | 4300.49 | 4339.32 |
| 4 | 2156.58 | 2029.26 | 2173.5 |
| 8 | 829.465 | 1016.8 | 1087.67 |
| 16 | 427.298 | 509.998 | 544.069 |
| 32 | 214.856 | 250.927 | 544.067 |

固定Bitwidth进行分析，我们发现访问带宽均随数据间隔Stride增大而近线性减小，这主要是shared memory的bank硬件机制引发的bank conflict导致的。当数据间隔按2的幂增大时，每个warp中线程访问的数据位于同一个bank的个数增加（见下表），这些访问指令变为了顺序执行，相应地使运行时间成倍增加。

这个趋势在数据位宽Bitwidth为4Bytes时表现地最为典型，这是因为shared memory访存的单位就是4Bytes。当数据位宽为2Bytes时，数据间隔为1或2都不会发生冲突，因此带宽基本一致；同时因为每个访问总是传输4Bytes而非所需的2Bytes，所以带宽整体比位宽为4Bytes的情况要小一半。而当数据位宽为8Bytes时，除了bank conflict路数比4Bytes增大了一倍、以及本身数据量大了一倍，这两者综合导致其与4Bytes情况带宽基本一致之外；在Stride为16和32时，由于warp中所有线程都已经冲突了，已经是最坏情况，因此带宽没有进一步下降。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N-way Bank Conflict  (1 for no conflict) | | Bitwidth | | |
| 2 | 4 | 8 |
| Stride | 1 | 1 | 1 | 2 |
| 2 | 1 | 2 | 4 |
| 4 | 2 | 4 | 8 |
| 8 | 4 | 8 | 16 |
| 16 | 8 | 16 | 32 |
| 32 | 16 | 32 | 32 |