体系结构Lab5 数据级并行

PB19000015 贾欣宇

CPU

三种实现的性能差异

三种方法进行规模为 $64 * 64$ 至 $2048 * 2048$ 的矩阵乘法用时如下表，其中 AVX 分块方法使用大小为 $32 * 32$ 的块.

规模	基础方法用时	AVX 方法用时	AVX 分块方法用时
64	0.000 606 s	0.000 134 s	0.000 219 s
128	0.004 963 s	0.001 120 s	0.001 367 s
256	0.047 524 s	0.010 030 s	0.009 994 s
512	0.540 867 s	0.111 728 s	0.085 713 s
1024	4.338 142 s	0.894 085 s	0.692 586 s
2048	95.078 06 s	17.859 58 s	6.108 356 s

可见 AVX 方法和 AVX 分块方法性能均优于基础方法；当数据规模较小时，AVX 方法优于 AVX 分块方法，数据规模较大时，AVX 分块方法优于 AVX 方法.

AVX 方法和 AVX 分块方法性能较优的原因是它们实现了一定并行处理；当数据规模较小时，数据规模接近于 AVX 分块方法的块大小，此时 AVX 分块方法的加速程度较小，无法弥补增加的预处理时间；数据规模较大时 AVX 分块方法对 cache 局部性利用充分的优势得以显现，性能也优于 AVX 方法.

分块参数对 AVX 分块方法性能的影响

以 AVX 分块方法在块大小为 $16 * 16$ 至 $128 * 128$ 情况下进行规模为 $1024 * 1024$ 的矩阵乘法用时如下表：

块大小	用时
16	0.710 957 s
32	0.692 586 s
64	0.763 671 s
128	0.950 402 s

可见分块过小或过大都会降低性能.

分块过小时，虽然可以较为充分的利用 cache 的局部性，但为此在每次循环前后增加的处理用时很长，得不偿失；分块过大时，不能很好的利用 cache 局部性，性能较差.

其他矩阵乘法的优化方式

- 循环展开：循环展开可以产生更让多指令级并行的机会，提高性能；
- 多处理器：让程序运行在多个处理器上，随着处理器线程数增加，程序性能也会提高.

GPU

三种实现的性能差异

三种方法进行规模为 $128 * 128$ 至 $2048 * 2048$ 的矩阵乘法用时如下表，其中 GPU 分块方法使用大小为 $8 * 8$ 的块.

规模	基础方法用时	GPU 方法用时	GPU 分块方法用时
128	4.963 ms	0.053 120 ms	0.045 088 ms
256	47.524 ms	0.251 583 ms	0.092 385 ms
512	540.867 ms	1.215 33 ms	0.334 75 ms
1024	4338.142 ms	1.977 46 ms	1.953 10 ms
2048	95078.06 ms	7.824 3 ms	7.866 2 ms

可见 GPU 方法和 GPU 分块方法性能均优于基础方法；当数据规模较小时，GPU 分块方法优于 GPU 方法，数据规模较大时，二者较为接近.

两种 GPU 方法均实现了并行处理，因此效率优于基础方法；GPU 分块方法利用了访存更快的 `shared memory`，效率更高.

GPU 方法参数对性能的影响

为了保证计算的可进行性，`gridsize` 取决于 `blocksize` 和数据规模. 以 GPU 方法在 `blocksize` 为 $8 * 8$ 至 $64 * 64$ 情况下进行规模为 $1024 * 1024$ 的矩阵乘法用时如下表：

`blocksize`	**用时**
8	1.954 29 ms
16	1.977 46 ms
32	7.656 55 ms
64	9.519 88 ms

可见`blocksize`过大时性能显著下降.

`blocksize`过大时，可能超出硬件设备支持的范畴，导致部分退化为串行处理，显著降低性能.

GPU 分块方法参数对性能的影响

为了保证计算的可进行性，`gridsize`取决于`blocksize`和数据规模. 以 GPU 分块方法在`BLOCK`大小为 $8*8$，`blocksize`为 $8*8$ 至 $64*64$ 情况下进行规模为 $1024*1024$ 的矩阵乘法用时如下表：

`blocksize`	**用时**
8	1.961 80 ms
16	1.953 10 ms
32	1.341 30 ms
64	1.358 70 ms

以 GPU 分块方法在`blocksize`为 $32*32$，`BLOCK`大小为 $8*8$ 至 $32*32$ 情况下进行规模为 $1024*1024$ 的矩阵乘法用时如下表：

`BLOCK` **大小**	**用时**
8	1.341 30 ms
16	1.549 30 ms
32	6.457 21 ms

可见`blocksize`增大时性能有所上升，而`BLOCK`大小过大时性能显著下降.

`BLOCK`大小过大时，数据可能超出`shared memory`能容纳的范围，部分退化为串行处理，导致性能降低；`blocksize`与`BLOCK`大小接近时，分块带来的性能提升可能被预处理带来的消耗所抵消，使性能有所下降.