

GPU Computing



规约算法

Hui Liu

Email: hui.sc.liu@gmail.com

Reduction #3: Sequential Accesses

Values (shared memory)

| | | | | | | | | | | | | | | | |
|----|---|---|----|---|----|---|---|----|----|---|---|---|----|---|---|
| 10 | 1 | 8 | -1 | 0 | -2 | 3 | 5 | -2 | -3 | 2 | 7 | 0 | 11 | 0 | 2 |
|----|---|---|----|---|----|---|---|----|----|---|---|---|----|---|---|

Step 1
Stride 8

Thread
IDs

0 1 2 3 4 5 6 7

Values

| | | | | | | | | | | | | | | | |
|---|----|----|---|---|---|---|---|----|----|---|---|---|----|---|---|
| 8 | -2 | 10 | 6 | 0 | 9 | 3 | 7 | -2 | -3 | 2 | 7 | 0 | 11 | 0 | 2 |
|---|----|----|---|---|---|---|---|----|----|---|---|---|----|---|---|

Step 2
Stride 4

Thread
IDs

0 1 2 3

Values

| | | | | | | | | | | | | | | | |
|---|---|----|----|---|---|---|---|----|----|---|---|---|----|---|---|
| 8 | 7 | 13 | 13 | 0 | 9 | 3 | 7 | -2 | -3 | 2 | 7 | 0 | 11 | 0 | 2 |
|---|---|----|----|---|---|---|---|----|----|---|---|---|----|---|---|

Step 3
Stride 2

Thread
IDs

0 1

Values

| | | | | | | | | | | | | | | | |
|----|----|----|----|---|---|---|---|----|----|---|---|---|----|---|---|
| 21 | 20 | 13 | 13 | 0 | 9 | 3 | 7 | -2 | -3 | 2 | 7 | 0 | 11 | 0 | 2 |
|----|----|----|----|---|---|---|---|----|----|---|---|---|----|---|---|

Step 4
Stride 1

Thread
IDs

0

Values

| | | | | | | | | | | | | | | | |
|----|----|----|----|---|---|---|---|----|----|---|---|---|----|---|---|
| 41 | 20 | 13 | 13 | 0 | 9 | 3 | 7 | -2 | -3 | 2 | 7 | 0 | 11 | 0 | 2 |
|----|----|----|----|---|---|---|---|----|----|---|---|---|----|---|---|

Reduction #4: Read two elements and do the first step

- Original: Each thread reads one element

```
// each thread loads one element from global to shared mem
unsigned int tid = threadIdx.x;
unsigned int i = blockIdx.x*blockDim.x + threadIdx.x;
sdata[tid] = g_idata[i];
__syncthreads();
```

- Read two and do the first reduction step:

```
// each thread loads two elements from global to shared mem
// end performs the first step of the reduction
unsigned int tid = threadIdx.x;
unsigned int i = blockIdx.x* blockDim.x * 2 + threadIdx.x;
sdata[tid] = g_idata[i] + g_idata[i + blockDim.x];
__syncthreads();
```

Performance for 4M element reduction

| | Time (2^{22} ints) | Bandwidth | Step Speedup | Cumulative Speedup |
|--|-----------------------|--------------------|--------------|--------------------|
| Kernel 1: interleaved addressing with divergent branching | 8.054 ms | 2.083 GB/s | | |
| Kernel 2: interleaved addressing non-divergent branching | 3.456 ms | 4.854 GB/s | 2.33x | 2.33x |
| Kernel 3: sequential addressing | 1.722 ms | 9.741 GB/s | 2.01x | 4.68x |
| Kernel 4: first step during global load | 0.965 ms | 17.377 GB/s | 1.78x | 8.34x |

THANK YOU

