Lab4 CUDA Advance

Nov, 2023 Parallel Programming
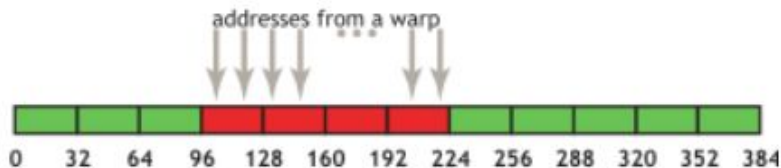
Overview

❖ Techniques that can further optimize a CUDA program

 ❖ Coalesced Memory Access

 ❖ Lower Precision

 ❖ Shared Memory

❖ Lab4

Coalesced Memory Access

❖ In short,
  ➢ Concurrent memory accesses in a warp should be continuous

addresses from a warp

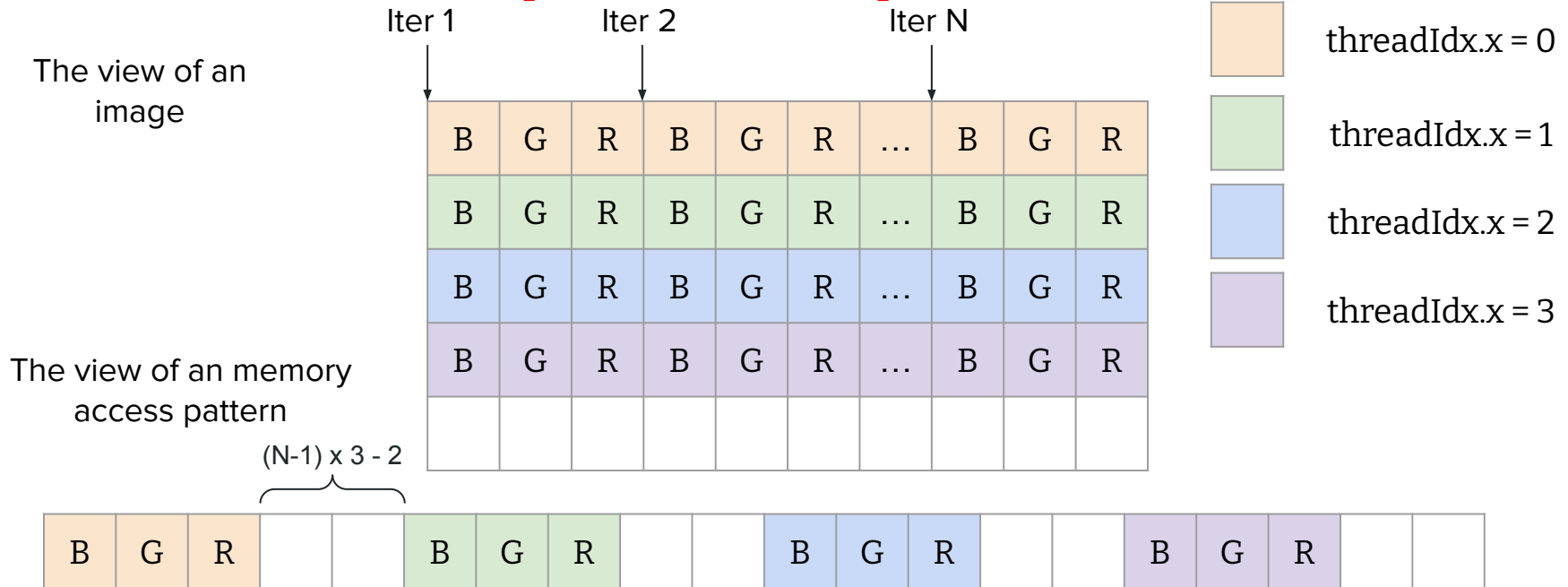| | | | | | | | | | | | |
|0|32|64|96|128|160|192|224|256|288|320|352|384|

❖ Why
  ➢ GPU has L2 (32 bytes), L1 (128 bytes) cache
  ➢ If memory accesses in a warp are continuous, it can
    ■ merge memory requests from all threads into a single memory request
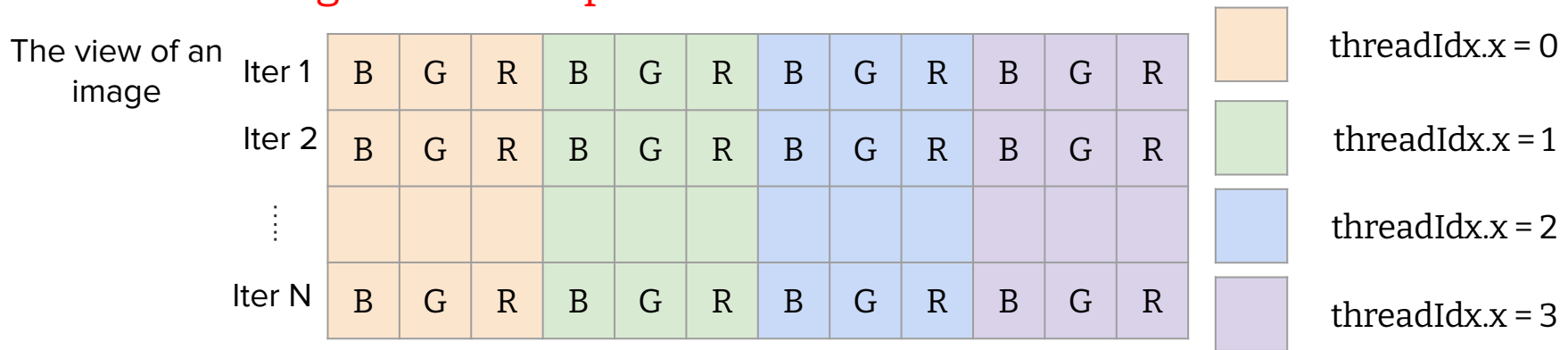   ■ utilize the cache
❖ Details
  ➢ CUDA Best Practices

Access without Coalesced Memory

❖ **If each thread compute a single row ->**
   Failed to combine requests into one request

Iter 1   Iter 2   Iter N

The view of an image

B	G	R	B	G	R	...	B	G	R
B	G	R	B	G	R	...	B	G	R
B	G	R	B	G	R	...	B	G	R
B	G	R	B	G	R	...	B	G	R

threadIdx.x = 0

threadIdx.x = 1

threadIdx.x = 2

threadIdx.x = 3

The view of an memory access pattern

(N-1) x 3 - 2

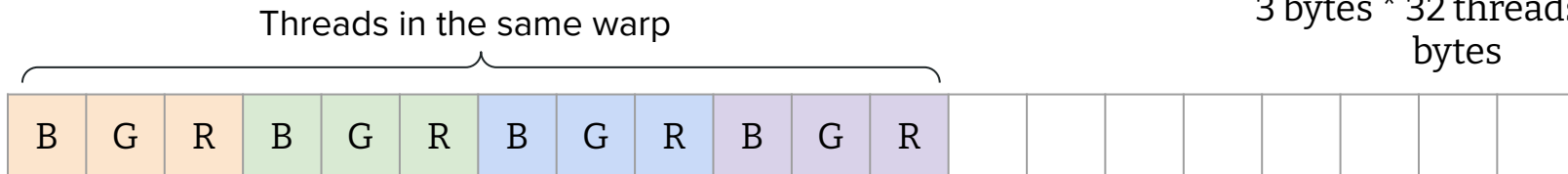| B	G	R			B	G	R			B	G	R			B	G	R		

Coalesced Memory Access

❖ The accesses can be combined into a single request if we change the access pattern

The view of an image

Iter 1	B	G	R	B	G	R	B	G	R	B	G	R
Iter 2	B	G	R	B	G	R	B	G	R	B	G	R
⋮												
Iter N	B	G	R	B	G	R	B	G	R	B	G	R

threadIdx.x = 0

threadIdx.x = 1

threadIdx.x = 2

threadIdx.x = 3

The view of an memory access pattern

Threads in the same warp

3 bytes * 32 threads = 96 bytes

| B	G	R	B	G	R	B	G	R	B	G	R									

Better Access Pattern

- ❖ In thread level, we should parallel x-axis
  - ➢ Different with CPU
- ❖ How to parallel y-axis and x-axis
  - ➢ Use block to parallel y
  - ➢ Launch 2D block
  - ➢ Combine both

Mixed-Precision

❖ Lower the precision of variables could reduce the computing time and also the computing accuracy

❖ Try to
  ❖ Use float to replace double
  ❖ Use fp16 to replace float
  ❖ Make sure using lower precision does not corrupt the results

Shared Memory

❖ Shared memory can greatly reduce the access time of a reused data item

Required data by t0

Required data by t1

Required data by t31

Shared data by t0 ~ t31
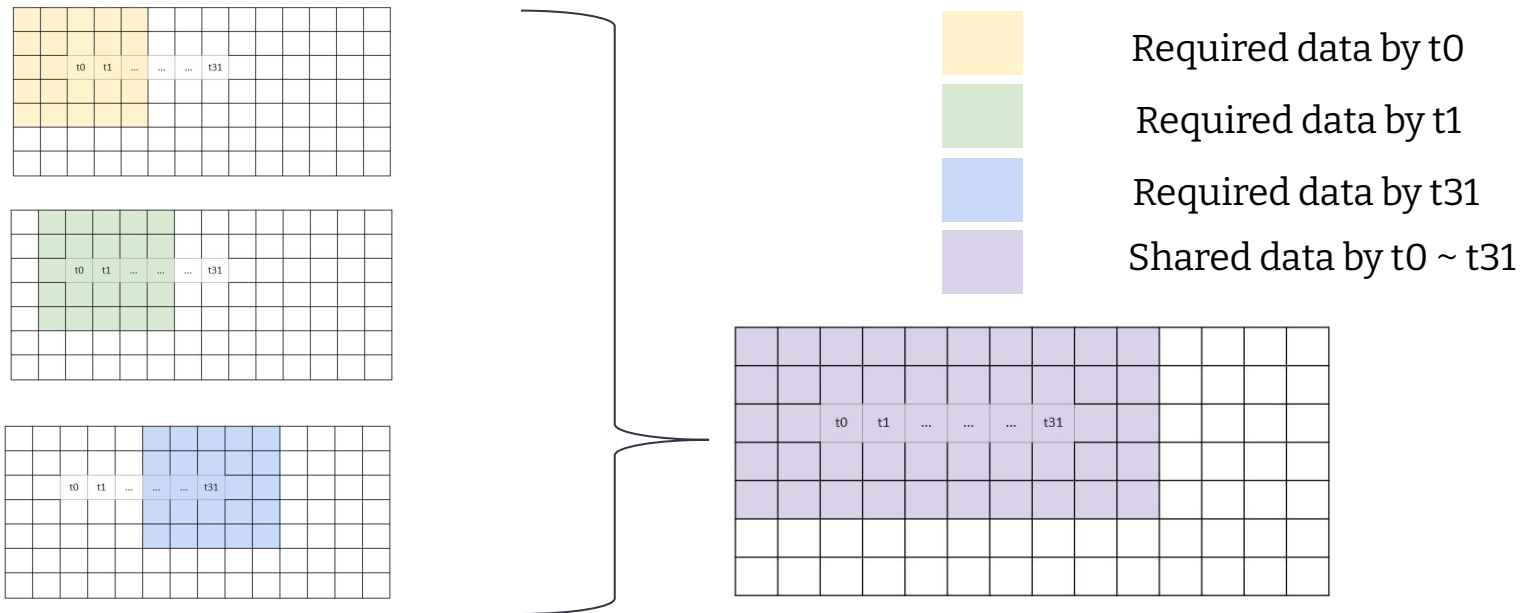
Using Shared Memory in Sobel

- ❖ Move the required data into shared memory
- ❖ Compute
- ❖ Update shared memory

		t0	t1	...	...	...	t31					

Lab4

❖ Optimize the sobel operator with the following
- ❑ Coalesced Memory
- ❑ Lower Precision
- ❑ Shared Memory

❖ TAs provided a sample CUDA program
- ❑ optimize it to be at least **13x faster**
- ➢ Materials are under **/home/pp23/share/lab4**

❖ Name your kernel as "sobel"

❖ We accept little pixel errors

Submission

- ❖ Finish it before 12/4 23:59
- ❖ Submit your code and Makefile (optional) to eeclass
- ❖ You can use **lab4-judge** for pre-check