

CUDA. Агрегация данных

Ахтямов Павел

МФТИ

Мотивация

- Мы можем сложить массивы

Мотивация

- Мы можем сложить массивы
- Но можем ли мы посчитать сумму элементов в массиве?

Задача Reduction

- Есть массив A

Задача Reduction

- Есть массив A
- Необходимо посчитать сумму элементу в массиве

Задача Reduction

- Есть массив A
- Необходимо посчитать сумму элементов в массиве
- Классическое решение: N операций сложения, 1 поток

Задача Reduction

- Есть массив A
- Необходимо посчитать сумму элементу в массиве
- Классическое решение: N операций сложения, 1 поток
- Параллельное решение на CPU: $N/C + C$ циклов по регистрам, C потоков

Задача Reduction

- Есть массив A
- Необходимо посчитать сумму элементу в массиве
- Классическое решение: N операций сложения, 1 поток
- Параллельное решение на CPU: $N/C + C$ циклов по регистрам, C потоков
- Решение на GPU: $C = 4352$?

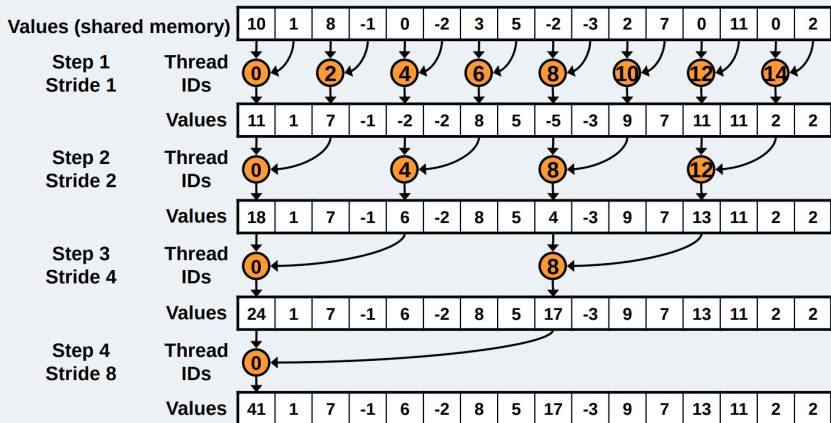
Задача Reduction

- Есть массив A
- Необходимо посчитать сумму элементу в массиве
- Классическое решение: N операций сложения, 1 поток
- Параллельное решение на CPU: $N/C + C$ циклов по регистрам, C потоков
- Решение на GPU: $C = 4352$?
- Не совсем!

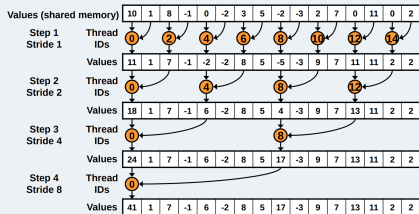
Решение задачи!

Решение задачи!

Каждый блок можно обработать за $O(\log BS)$

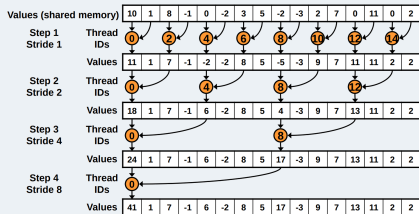


А что с warp-ами? $BS = 256$



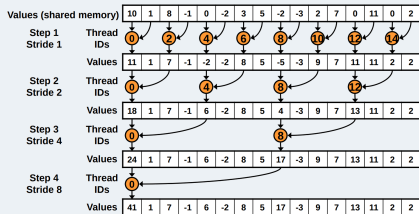
А что с warp-ами? $BS = 256$

- 1 шаг (0, 2, 4, ...) - 8



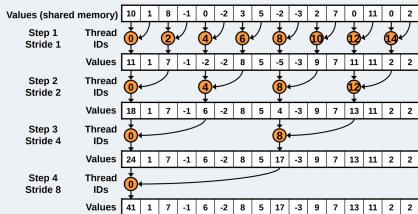
А что с warp-ами? $BS = 256$

- 1 шаг (0, 2, 4, ...) - 8
- 2 шаг (0, 4, 8, ...) - 8



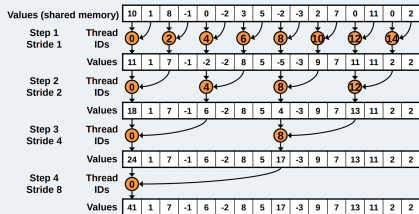
А что с warp-ами? $BS = 256$

- 1 шаг (0, 2, 4, ...) - 8
- 2 шаг (0, 4, 8, ...) - 8
- 3 шаг (0, 8, 16, ...) - 8

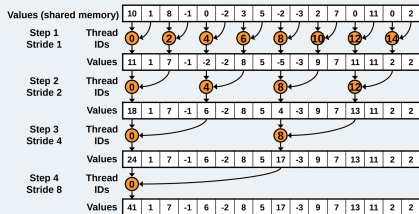


А что с warp-ами? $BS = 256$

- 1 шаг (0, 2, 4, ...) - 8
- 2 шаг (0, 4, 8, ...) - 8
- 3 шаг (0, 8, 16, ...) - 8
- 4 шаг (0, 16, 32, ...) - 8

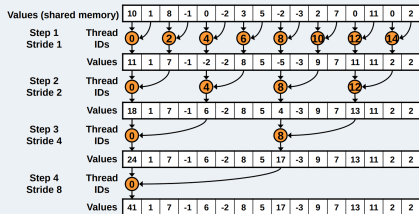


А что с warp-ами? $BS = 256$



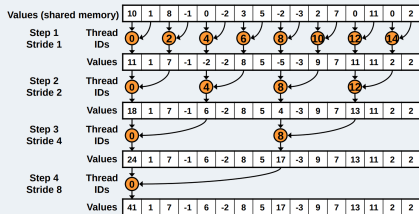
- 1 шаг (0, 2, 4, ...) - 8
- 2 шаг (0, 4, 8, ...) - 8
- 3 шаг (0, 8, 16, ...) - 8
- 4 шаг (0, 16, 32, ...) - 8
- 5 шаг (0, 32, 64, ...) - 8

А что с warp-ами? $BS = 256$



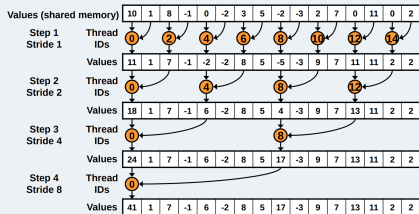
- 1 шаг (0, 2, 4, ...) - 8
- 2 шаг (0, 4, 8, ...) - 8
- 3 шаг (0, 8, 16, ...) - 8
- 4 шаг (0, 16, 32, ...) - 8
- 5 шаг (0, 32, 64, ...) - 8
- 6 шаг (0, 64, 128, ...) - 4

А что с warp-ами? $BS = 256$



- 1 шаг (0, 2, 4, ...) - 8
- 2 шаг (0, 4, 8, ...) - 8
- 3 шаг (0, 8, 16, ...) - 8
- 4 шаг (0, 16, 32, ...) - 8
- 5 шаг (0, 32, 64, ...) - 8
- 6 шаг (0, 64, 128, ...) - 4
- 7 шаг (0, 128) - 2

А что с warp-ами? $BS = 256$

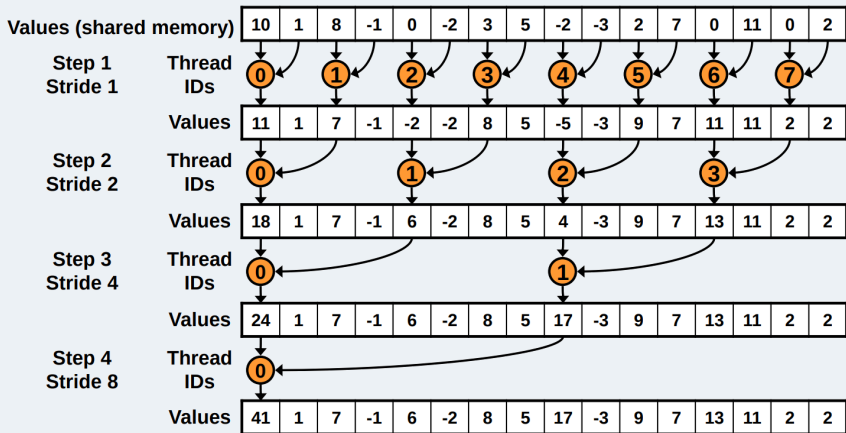


- 1 шаг (0, 2, 4, ...) - 8
- 2 шаг (0, 4, 8, ...) - 8
- 3 шаг (0, 8, 16, ...) - 8
- 4 шаг (0, 16, 32, ...) - 8
- 5 шаг (0, 32, 64, ...) - 8
- 6 шаг (0, 64, 128, ...) - 4
- 7 шаг (0, 128) - 2
- 8 шаг (0) - 1
- **Итого:** 47 warp-ов!

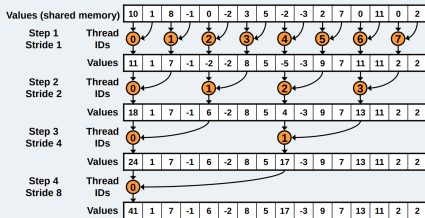
Решение задачи!

Решение задачи!

Нумерация потоков изменена!

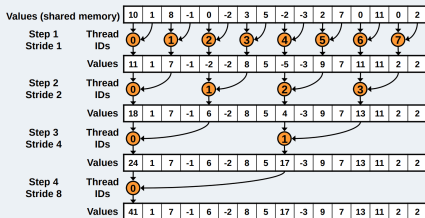


А что с warp-ами? $BS = 256$



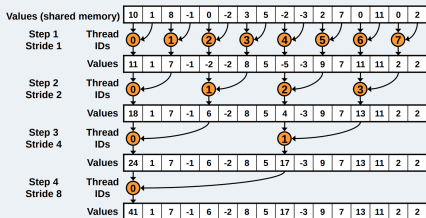
А что с warp-ами? $BS = 256$

- 1 шаг (0, 1, ..., 127) - 4



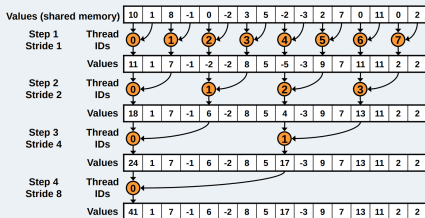
А что с warp-ами? $BS = 256$

- 1 шаг (0, 1, ..., 127) - 4
- 2 шаг (0, 1, ..., 63) - 2



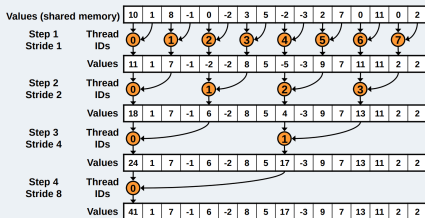
А что с warp-ами? $BS = 256$

- 1 шаг (0, 1, ..., 127) - 4
- 2 шаг (0, 1, ..., 63) - 2
- 3 шаг (0, 1, ..., 31) - 1

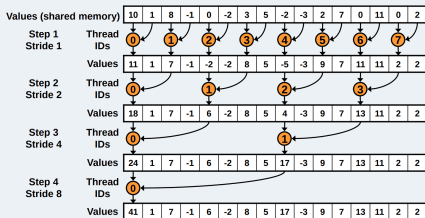


А что с warp-ами? $BS = 256$

- 1 шаг (0, 1, ..., 127) - 4
- 2 шаг (0, 1, ..., 63) - 2
- 3 шаг (0, 1, ..., 31) - 1
- 4 шаг (0, 1, ..., 15) - 1

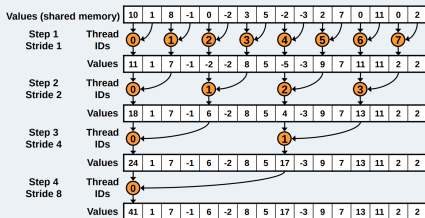


А что с warp-ами? $BS = 256$



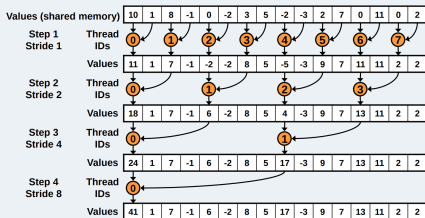
- 1 шаг (0, 1, ..., 127) - 4
- 2 шаг (0, 1, ..., 63) - 2
- 3 шаг (0, 1, ..., 31) - 1
- 4 шаг (0, 1, ..., 15) - 1
- 5 шаг (0, 1, ..., 7) - 1

А что с warp-ами? $BS = 256$



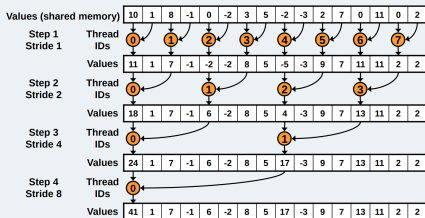
- 1 шаг (0, 1, ..., 127) - 4
- 2 шаг (0, 1, ..., 63) - 2
- 3 шаг (0, 1, ..., 31) - 1
- 4 шаг (0, 1, ..., 15) - 1
- 5 шаг (0, 1, ..., 7) - 1
- 6 шаг (0, 1, ..., 3) - 1

А что с warp-ами? $BS = 256$



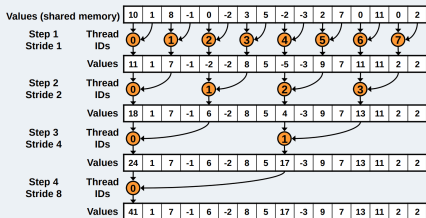
- 1 шаг (0, 1, ..., 127) - 4
- 2 шаг (0, 1, ..., 63) - 2
- 3 шаг (0, 1, ..., 31) - 1
- 4 шаг (0, 1, ..., 15) - 1
- 5 шаг (0, 1, ..., 7) - 1
- 6 шаг (0, 1, ..., 3) - 1
- 7 шаг (0, 1) - 1

А что с warp-ами? $BS = 256$



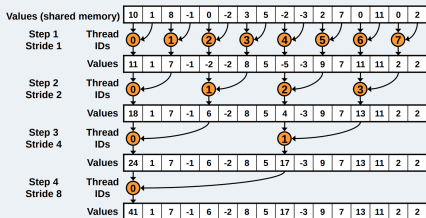
- 1 шаг (0, 1, ..., 127) - 4
- 2 шаг (0, 1, ..., 63) - 2
- 3 шаг (0, 1, ..., 31) - 1
- 4 шаг (0, 1, ..., 15) - 1
- 5 шаг (0, 1, ..., 7) - 1
- 6 шаг (0, 1, ..., 3) - 1
- 7 шаг (0, 1) - 1
- 8 шаг (0) - 1
- **Итого:** 12 warp-ов!

Но и это не очень реализация



Но и это не очень реализация

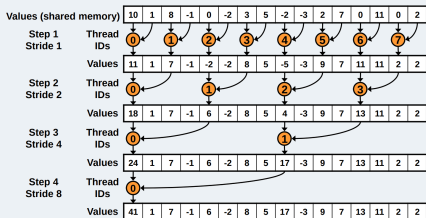
Рассмотрим пример:



Но и это не очень реализация

Рассмотрим пример:

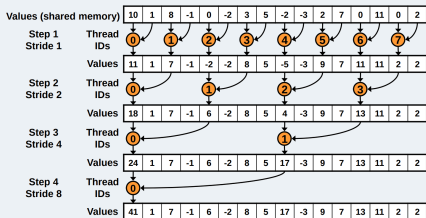
- 0-й поток берет сумму 0 и 16, пишет в 0



Но и это не очень реализация

Рассмотрим пример:

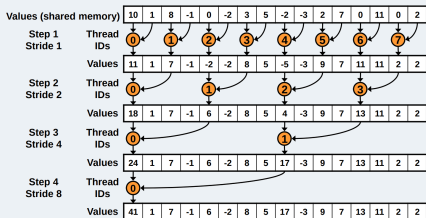
- 0-й поток берет сумму 0 и 16, пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32



Но и это не очень реализация

Рассмотрим пример:

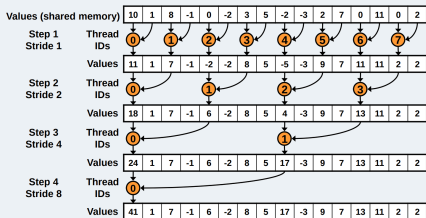
- 0-й поток берет сумму 0 и 16, пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32
- 2-й поток берет сумму 64 и 80, пишет в 64



Но и это не очень реализация

Рассмотрим пример:

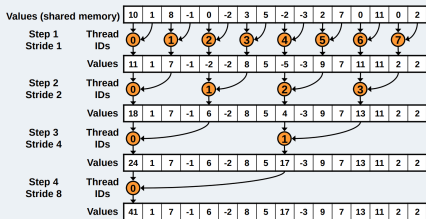
- 0-й поток берет сумму 0 и 16, пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32
- 2-й поток берет сумму 64 и 80, пишет в 64
- 3-й поток берет сумму 96 и 112, пишет в 96



Но и это не очень реализация

Рассмотрим пример:

- 0-й поток берет сумму 0 и 16, пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32
- 2-й поток берет сумму 64 и 80, пишет в 64
- 3-й поток берет сумму 96 и 112, пишет в 96

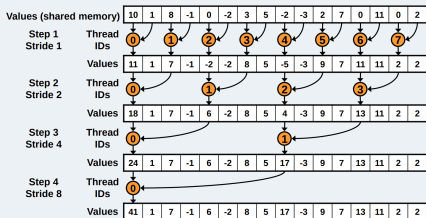


Размер кеш-линии - 32!

Но и это не очень реализация

Рассмотрим пример:

- 0-й поток берет сумму 0 и 16, пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32
- 2-й поток берет сумму 64 и 80, пишет в 64
- 3-й поток берет сумму 96 и 112, пишет в 96



Размер кеш-линии - 32!

У нас банк-конфликт!

Банк-конфликт

Банк-конфликт

Bank conflict

Поведение в shared-memory, когда два потока внутри одного warp-а записывают данные внутри разных кеш-линий по одному индексу (по модулю WS) - банк!

Банк-конфликт

Bank conflict

Поведение в shared-memory, когда два потока внутри одного warp-а записывают данные внутри разных кеш-линий по одному индексу (по модулю WS) - банк!

- 0-й поток берет сумму 0 и 16 , пишет в 0

Банк-конфликт

Bank conflict

Поведение в shared-memory, когда два потока внутри одного warp-а записывают данные внутри разных кеш-линий по одному индексу (по модулю WS) - банк!

- 0-й поток берет сумму 0 и 16 , пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32

Банк-конфликт

Bank conflict

Поведение в shared-memory, когда два потока внутри одного warp-а записывают данные внутри разных кеш-линий по одному индексу (по модулю WS) - банк!

- 0-й поток берет сумму 0 и 16 , пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32

0-й поток и 1-й конфликтуют по 0-му банку!!

Банк-конфликт

Bank conflict

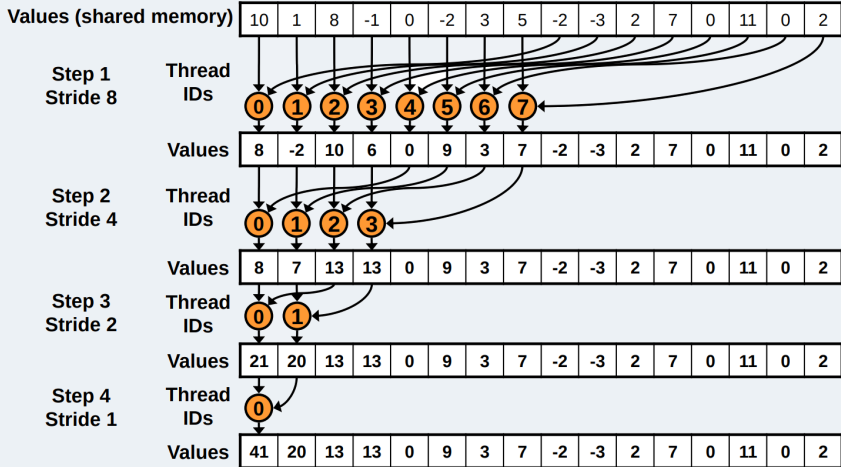
Поведение в shared-memory, когда два потока внутри одного warp-а записывают данные внутри разных кеш-линий по одному индексу (по модулю WS) - банк!

- 0-й поток берет сумму 0 и 16 , пишет в 0
- 1-й поток берет сумму 32 и 48, пишет в 32

0-й поток и 1-й конфликтуют по 0-му банку!!

Запись - последовательно!

Как решить?



Как решить итоговую задачу?

Как решить итоговую задачу?

- Посчитать сумму в блоке

Как решить итоговую задачу?

- Посчитать сумму в блоке
- Посчитать сумму между блоками

Как решить итоговую задачу?

- Посчитать сумму в блоке
- Посчитать сумму между блоками
- Повторить, пока не останется 1 блок

Что мы сделали

- Реализовали операцию сложения

Что мы сделали

- Реализовали операцию сложения
- Поняли, что индексация решает

Что мы сделали

- Реализовали операцию сложения
- Поняли, что индексация решает
- Избежали банк-конфликтов!