Vienna University of Technology

360.252 Computational Science on Many Core Architectures

Institute for Microelectronics

Exercise 6

Authors:
Camilo Tello Fachin
12127084

Supervisor:
Dipl.-Ing. Dr.techn. Karl Rupp

November 28, 2022

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Abstract

Here documented the results of exercise 6.

Contents

1 Task1 : Inclusive and Exclusie Scan (1/4 Points)

1.1 Describe the Workings of the Kernels (1 Point)

workings of `scan_kernel_1()`

Algorithm `scan_kernel_1()` - Scan within each block

1: Call `scan_kernel_1` ▷ with `<<<256,256>>>`
2: Kernel initiates `shared_buffer[256]` and `my_value`
3: Calculate `work_per_thread` from `N` ▷ Yields indices of vector chunk interfaces
4: Inc. scan for single vector chunk within `shared_buffer[256]` ▷ concurrent for chunks
5: All threads write `my_value` except last one ▷ `(conditon) ? exp_1 : exp_2`
6: Write result to vector `Y`

workings of `scan_kernel_2()`

Algorithm `scan_kernel_2()` - Add results from Kernels

1: Call `scan_kernel_2` ▷ with `<<<1,256>>>`
2: Reduction over all Kernels
3: Write to output array ▷ Exclusive Scan manner:
4: `carries[threadIdx.x] = (threadIdx.x > 0) ? shared_buffer[threadIdx.x - 1] : 0;`

workings of `scan_kernel_3()`

Algorithm `scan_kernel_3()`

1: call `scan_kernel_3` ▷ with `<<<256,256>>>`
2: Kernel initiates `shared_offset`
3: Calculate `work_per_thread` from `N` ▷ Yields indices of vector chunk interfaces
4: 0^{th} thread of every block set `offset` as `carries` from `blockIdx.x`
5: Add `offset` to all values between `blockstart` and `blockstop`:
6: `Y[i] += shared_offset;`

1.2 Provide an Implementation of Inclusive Scan (1 Point)

1.3 Modify Exclusive Scan Code to Convert to Inclusive scan (1 Point)

1.4 Compare Performances (1 Point)

2 Task 2: Finite Differences on the GPU (0/5 Points)

2.1 Kernel which counts/stores Number of Nonzero Entries for Rows of A (1 Point)

2.2 Row Offset Array of CSR Format (1 Point)

2.3 Kernel to Write Column Indices and Nonzero Matrix Values to CSR Arrays (1 Point)

2.4 Code to Allocate Necessary Arrays and call CG-Solver (1 Point)

2.5 Benchmark Code for Different System Sizes (1 Point)