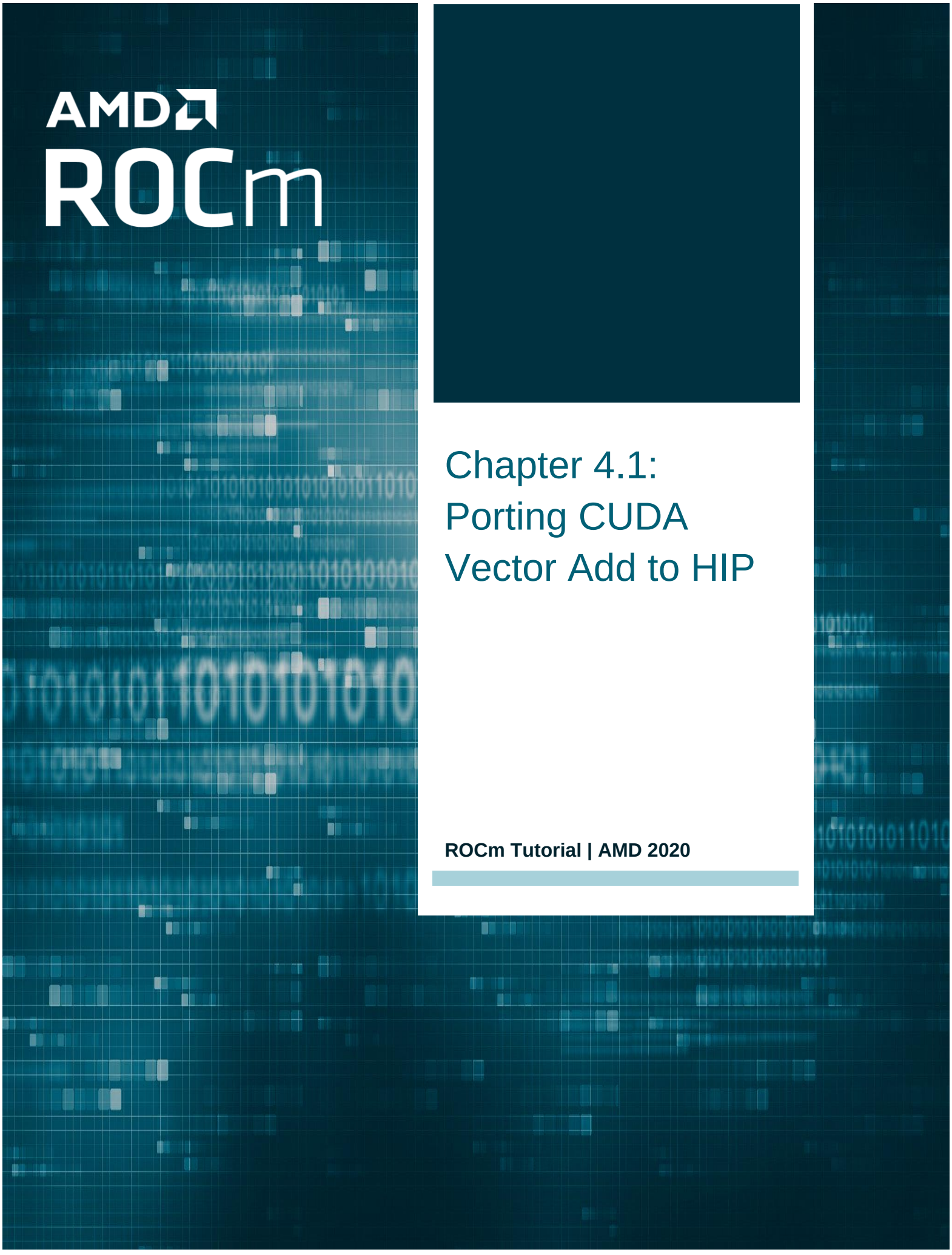

Chapter 4.1: Porting CUDA Vector Add to HIP

ROCm Tutorial | AMD 2020

AMD ROCm

Table of Contents

AMD ROCm

Chapter 4.1: Porting CUDA Vector Add to HIP

This hands-on tutorial shows how we can convert a vector add application written in CUDA to HIP using the HIP source conversion tools.

Preparation

1. In the tutorial repo move to the following folder:

   cd Chapter4/01_Cuda_To_HIP_Vector_Add

2. Open the file vadd_cuda.cu
   - You can see this contains CUDA specific code i.e. using CUDA

3. Let us create a new folder and copy this file there:
   - mkdir HIP_Vector_Add && cp vadd_cuda.cu
   - cd HIP_Vector_Add

4. It is not compulsory to create a separate folder and move the file. But in many cases, we want to preserve the original CUDA code instead of overwriting it.

5. We are now ready to hipify this code

Scanning

1. Run the following command:

   hipexamine-perl.sh .

   where "." refers to the current directory

   - You will see the scanned output in **Figure 1**
   - Note that scanning only shows what is possible for conversion, it does not do the

conversion.

2. From Figure 1:
    - 17 CUDA calls can be converted to HIP
    - A complete breakdown of what those 17 correspond to are then provided in the brackets
    - A count of how many HIP API calls of each type were added is also shown
    - The value of warn is 0 which means all CUDA API calls can be converted successfully

```
  info: converted 17 CUDA->HIP refs ( error:0 init:0 version:0 device:1 context:0 module:0 memory:9 virtual_memory:
0 addressing:0 stream:0 event:0 external_resource_interop:0 stream_memory:0 execution:0 graph:0 occupancy:0 texture
:0 surface:0 peer:0 graphics:0 profiler:0 openGL:0 D3D9:0 D3D10:0 D3D11:0 VDPAU:0 EGL:0 thread:0 complex:0 library:
0 device_library:0 device_function:3 include:0 include_cuda_main_header:0 type:0 literal:0 numeric_literal:3 define
:0 extern_shared:0 kernel_launch:1 )
  warn:0 LOC:107 in './vadd_cuda.cu'
  hipFree 3
  hipMemcpy 3
  hipMalloc 3
  hipMemcpyHostToDevice 2
  hipLaunchKernelGGL 1
  hipMemcpyDeviceToHost 1
  hipDeviceSynchronize 1
```

Figure 1: Output of HIP scanning tool

Porting

1. Now that we have scanned and observed that porting can be done, we will use the hipconvertinplace-perl script to actually do the conversion.

2. Run “hipconvertinplace-perl.sh .” where “.” refers to the current directory
    - If there are multiple files in the directory using “.” will hipify all the files
    - If you only want to convert a single file use “hipconvertinplace-perl.sh filename” to hipify a specific file

Compiling and Executing

1. We will be using “hipcc” for compiling
    - hipcc vadd_cuda.cu -o vadd_cuda

2. Now let us run the code:
    - ./vadd_cuda

3. You will see the output as shown in Figure 2

```
Printing a subset of results till index 1024
Value at index 0 is 0.000000
Value at index 1 is 2.000000
Value at index 2 is 4.000000
Value at index 3 is 6.000000
Value at index 4 is 8.000000
Value at index 5 is 10.000000
Value at index 6 is 12.000000
Value at index 7 is 14.000000
Value at index 8 is 16.000000
Value at index 9 is 18.000000
Value at index 10 is 20.000000
Value at index 11 is 22.000000
Value at index 12 is 24.000000
Value at index 13 is 26.000000
Value at index 14 is 28.000000
Value at index 15 is 30.000000
Value at index 16 is 32.000000
Value at index 17 is 34.000000
Value at index 18 is 36.000000
Value at index 19 is 38.000000
Value at index 20 is 40.000000
Value at index 21 is 42.000000
Value at index 22 is 44.000000
```

Figure 2: Partial Output of the Vector Add Application