



COA ASSIGNMENT 1

GPGPU SIM



Group 9:

Ponnanna A H

Priyam Saha

Arihant Garg

Shubh Modi

Tejas Singh Rajput

JULY 31, 2023

IMPLEMENTATION:

The following commands have been tested in Ubuntu v20.04.6, gcc 7.5.0, g++ 7.5.0 and NVIDIA CUDA 11.1.0.

STEP1(Commands):

- `sudo apt install gcc-7`
- `sudo apt install g++-7`
- `sudo apt install git`
- `sudo apt install python3-pip`
- `sudo apt-get install build-essential xutils-dev bison zlib1g-dev flex libglu1-mesa-dev`
- `sudo apt-get install doxygen graphviz`
- `sudo apt-get install python-pmw python3-ply python3-numpy libpng-dev python3-matplotlib`
- `sudo apt-get install libxi-dev libxmu-dev freeglut3-dev`
- `git clone https://github.com/gpgpu-sim/gpgpu-sim_distribution`
- `wget https://developer.download.nvidia.com/compute/cuda/11.1.0/local_installers/cuda_11.1.0_455.23.05_linux.run`
- `sudo sh cuda_11.1.0_455.23.05_linux.run` (this will open a window; type accept and then select everything except the driver and proceed with the installation)

STEP2(Commands): (Building the files)

- `cd gpgpu-sim_distribution`
- `export CUDA_INSTALL_PATH=/usr/local/cuda`
- `bash`
- `source setup_environment`
- `make`

STEP3(Commands):

- `cd`
- `nano .bashrc` (opens bash file) inside the bash file, go to the end and enter the following:

```
export PATH="/usr/local/cuda-11.1/bin:$PATH"
```

```
export LD_LIBRARY_PATH="/usr/local/cuda-11.1/lib64:$LD_LIBRARY_PATH"
```

Then, `Ctrl+S` to save the file and `Ctrl+X` to close the file.
- `source .bashrc`
- `mkdir test`
- `cd test`
- `touch test.cu`

Our sample "Hello World!" program:

After this, open `test.cu` in `/home/test` folder and paste the below code:

```
C++ > cuda.cu > main(void)
1  #include <stdio.h>
2  #include <cuda.h>
3  #include <stdlib.h>
4
5  __global__ void helloFromGPU(void)
6  {
7      printf("Hello World from GPU! threadIdx.x=%d\n", threadIdx.x);
8  }
9
10 int main(void)
11 {
12     printf("Hello World from CPU!\n");
13
14     helloFromGPU <<<1, 10>>>();
15     cudaDeviceReset();
16     return(0);
17 }
```

Running our “Hello World” through terminal:

- Save test.cu and quit the file.
- `cp -r ~/gpgpu-sim_distribution/configs/tested-cfgs/SM75_RTX2060/* ./`
- `source ~/gpgpu-sim_distribution/setup_environment`
- `nvcc -test.cu -lcudart`
- `./a.out`

OUTPUT:

```
GPGPU-Sim PTX: Kernel '_Z12helloFromGPUv' : regs=8, lmem=0,
Hello World from CPU!

GPGPU-Sim PTX: cudaLaunch for 0x0x55f6bb31a461 (mode=functi
GPGPU-Sim PTX: finding reconvergence points for '_Z12helloF
GPGPU-Sim PTX: Finding dominators for '_Z12helloFromGPUv'..
GPGPU-Sim PTX: Finding immediate dominators for '_Z12helloF
GPGPU-Sim PTX: Finding postdominators for '_Z12helloFromGPU
GPGPU-Sim PTX: Finding immediate postdominators for '_Z12he
GPGPU-Sim PTX: pre-decoding instructions for '_Z12helloFrom
GPGPU-Sim PTX: reconvergence points for _Z12helloFromGPUv..
GPGPU-Sim PTX: ... end of reconvergence points for _Z12hell
GPGPU-Sim PTX: ... done pre-decoding instructions for '_Z12
GPGPU-Sim PTX: pushing kernel '_Z12helloFromGPUv' to stream
GPGPU-Sim: Performing Functional Simulation, executing kern
GPGPU-Sim PTX: PDOM analysis already done for _Z12helloFrom
padded cta size is 32 and 8 and 8GPGPU-Sim uArch: CTA/core
Max CTA : 32
GPGPU-Sim PTX: finding reconvergence points for 'vprintf'..
Hello World from GPU! threadIdx.x=0
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=1
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=2
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=3
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=4
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=5
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=6
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=7
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=8
GPGPU-Sim PTX: PDOM analysis already done for vprintf
Hello World from GPU! threadIdx.x=9
Destroy streams for kernel 1: size 0
GPGPU-Sim: Done functional simulation (130 instructions sim
```

GROUP SUGGESTIONS:

- **Organize Installation Steps:** Create a clear step-by-step installation guide with headings for each software/package to be installed. This will make it easier for readers to follow along and understand the installation process.
- **Include System Requirements:** Before listing the installation steps, mention the system requirements such as the recommended operating system version, hardware specifications, and any other dependencies required to run the project successfully.
- **Mention Version Numbers:** When installing packages, specify the version numbers to ensure consistency and compatibility. For example, instead of just `"sudo apt install gcc,"` it's better to use `"sudo apt install gcc-7"` to be more specific about the version.
- **Include Troubleshooting Tips:** Anticipate common issues that users may face during the installation process and provide troubleshooting tips or links to relevant resources. This will help users overcome obstacles without having to search for solutions elsewhere.
- **Include Update Instructions:** If applicable, include instructions for updating the project in the future. This will help users keep their installation up-to-date and take advantage of any new features or bug fixes