CSC4005 CUDA Emulator Manual

Dear all,

We have prepared a CUDA Emulator Virtual Machine for you. You can **emulate CUDA or CUDA-GUI program** on your personal computer (**without Nvidia GPU!**).

Acknowledgement:

Setting Up Virtual Machine

1. Download CUDA Emulator VM.
2. Import to VMware.

If it says `importing failed`, please click `Retry`.

If your host machine has 8 CPU cores, please allocate 2 cores to this VM.

If your host machine has 4 CPU cores, please try to allocate 2 cores to this VM.

3. Power on.

If you find it black, just wait for a few seconds (it should not be black for several minutes). If so, please report.
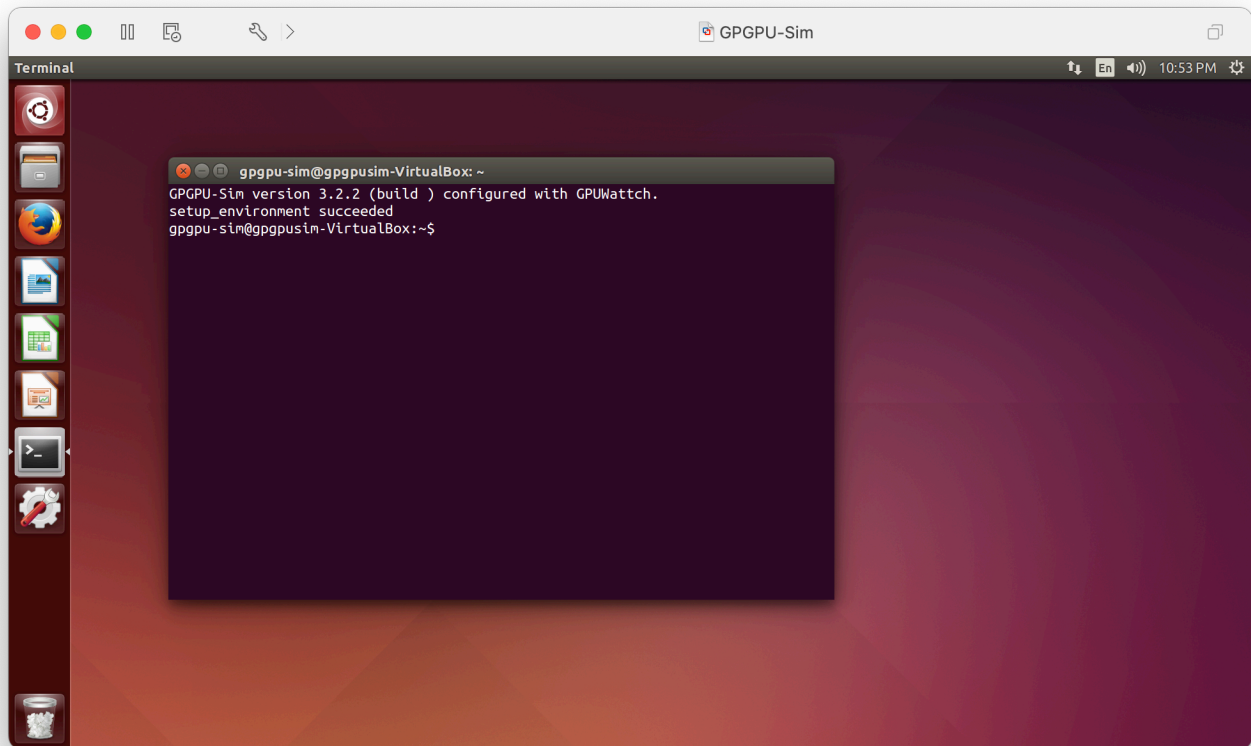
Usage

Username and password

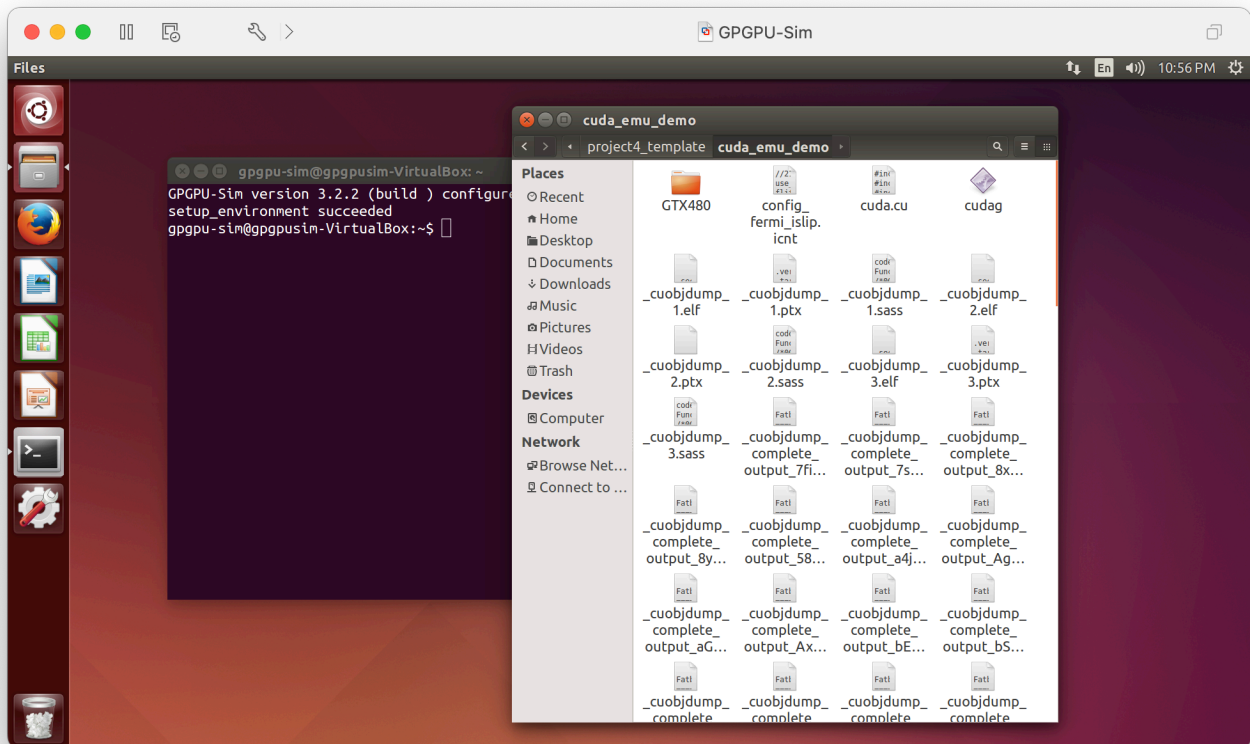**Username: gpgpu-sim**

Password: gpgpu-sim

1. Open terminal.

Once you see the prompt saying `setup_environment succeed`, you can directly run your CUDA-GUI program with only CPU.

4. Compile your **CUDA-GUI** program

Here we provide a cuda implementation for CSC4005 Project 3 for demonstration.

Git clone the latest version of CSC4005_Demo. Go to `project3_template/cuda_emu_demo`. (**Notice that you must compile and run your program inside this directory, because it contains configuration of GTX480 GPU.**). You will find `cuda.cu`.

First go to this directory,

```
cd $Somewhere/CSC4005_2022Fall_Demo/project4_template/cuda_emu_demo
```

Then compile your CUDA-GUI program:

```
nvcc cuda.cu -o cudag -lglut -lGLU -lGL -lm -O2 -DGUI
```

Notice that `nvcc` and `gcc` on this VM does not support c++11, so you should avoid using newest features when you debug your CUDA code (like `chrono::high_resolution_clock`).

Then you will see `cudag` in this directory.

Run it as you run a regular CUDA application:

```
./cudag $n_body $n_iterations
```

Notice that the performance of CPU is relatively poor, so you should consider using small `$n_body`. Here we use

```
./cudag 50 100
```

Then you would wait a few seconds when the program is initializing.

Then you will see the bodies moving.

Another information is, there will be debugging information displaying in terminal, it is not a signal of error.
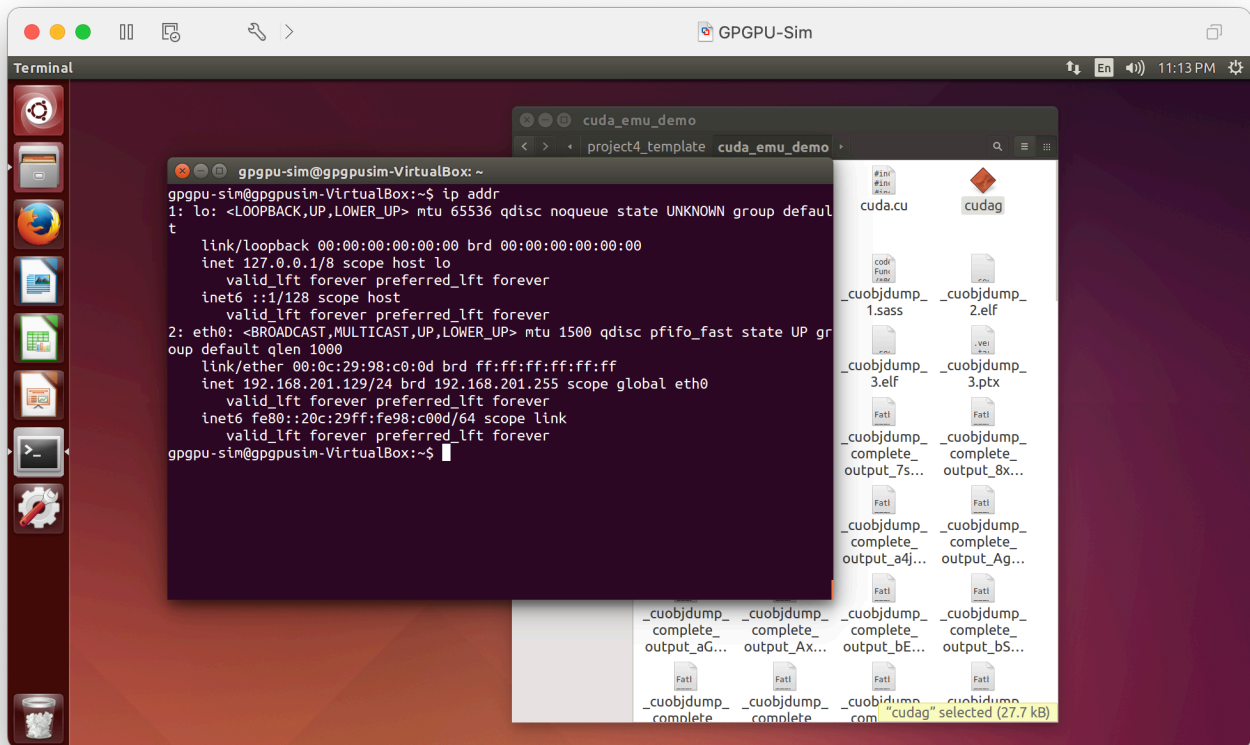
Notice

As mentioned by authors of gpgpu-sim, some cuda api are not available for gpgpu-sim like `syncthreads()`.
Please pay attention.

VSCode Remote to VM

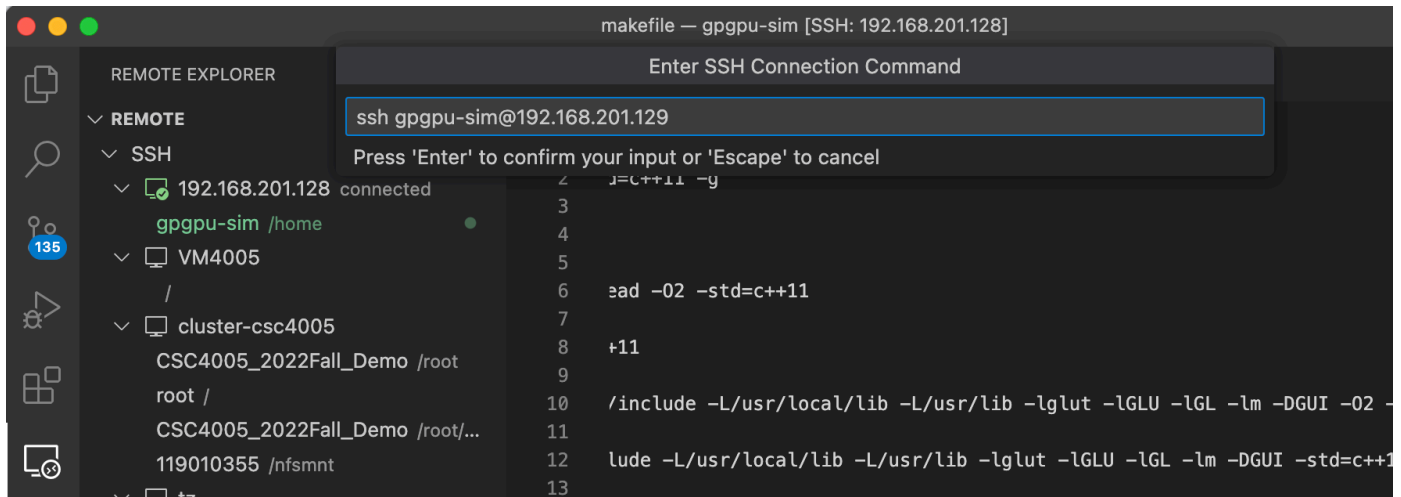This requires some fussy operations.

1. First, you need to know IP address of this VM.

Open a terminal, type `ip addr` to show ip addresses.

Please notice `2:eth0`. Remember the IP address in `2:eth0`. In my case, it is, `192.168.201.129`.

2. Open VSCode on your host machine

Choose Remote tab, and add a new server.

Type `ssh gpgpu-sim@192.168.201.129` (in my case, replace IP address with your own).

Then save and connect.

Then input your password following prompt.

Then you open folder `/home/gpgpu-sim/`.

Then you will have a good coding experience.

3.*This method is applicable for CSC4005 VM as well.*

Any problems please email 119010355@link.cuhk.edu.cn.

Hope it is helpful.

Bokai XU