# **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 *sn\_body*. Here we use

```
./cudag 50 100
```

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

			GPGPU-Sim			D
I Body Simulatio	on CUDA Implementation				t,	<b>En ◀))</b> 11:04 PM
	N Body Simulation CUDA Im	nplemen				
		< > < pr	oject4_template	•	९ ≡ ःः	
				#ind #ind		
		les)   (2 samples) = 0.00161142 (2 samples)		cuda.cu	cudag	
		>les) 2\samoles)_,  e = 0.000545756 (2 sample: >les)	s)	_cuobjdump_	_cuobjdump_	
		'(2 samples) = 0.00161142 (2 samples)  les) {(2 samples) = 0.00161142 (2 samples)		1.sass 	2.elf 	
	Injected packet size	= 0.00101142 (2 samples) (les) 2 samples) e average = 2.95264 (2 samples)		_cuobjdump_	_cuobjdump_	
*	Accepted packet size Hops average = 1 (2	e average = 2.95264 (2 samples) samples) END-of-Interconnect-DETA	ILS	complete_ output_7s	complete_ output_8x	
23 ?	gpgpu_simulation_tir gpgpu_simulation_rat	me = 0 days, 0 hrs, 0 min, 1 sec ( te = 90366 (inst/sec)	(1 sec)	_cuobjdump_ complete_ output_a4j	_cuobjdump_ complete_ output_Ag	
		(Cycle/sec)	cuobidumo cuobidumo	Fatt	Fati	
			completeoutput_aGoutput_AX	complete_ output_bE	complete_ output_bS	
-		Law Contraction	Fatt _cuobjdumpcuobjdump_	_cuobidump	Fatt	
			complete complete	com cudag	Selected (27.7 KB)	

Then you will see the bodies moving.

• • • 00	Ð	$\langle S \rangle$		CPGPU-Sin	n		đ
Body Simulation C	UDA Imple	ementation				1	📭 En 🐠)) 11:04 PM -
	Body Sim	ulation CUDA Implen	er 😮 🕒 💿 cuda_emu < > < project4_	_demo template <b>cuda_emu_de</b>	emo≯		
			-VirtualBox: ~/CSC4005_2022Fall_Dem  ples)  ples)  e = 0.000666617 (12 samples)		uda eint cuda.cu	cudag	
3			ples) 3 (12 samples) 9 = 0.000666617 (12 samples) 10 [les) 3 (12 samples) = 0.00196825 (12 samples)		_cuobjdump_ 1.sass	_cuobjdump_ 2.elf	
5	Injoct	ad packat size ave	<pre>iples) i (12 samples) = 0.00196825 (12 samples) iples) ! (12 samples) . (12 samples) . (12 samples)</pre>		_cuobjdump_ 3.elf	_cuobjdump_ 3.ptx	
6	Accepte Hops av	ed packet size ave ed packet size ave verage = 1 (12 sam	rage = 2.9526 (12 samples) rage = 2.9526 (12 samples) ples) END-of-Interconnect-DETAILS		complete_ output_7s	_cubjdump_ complete_ output_8x	
<del>-</del> • ?	gpgpu_s gpgpu_s gpgpu_s	simulation_time = simulation_rate = simulation_rate =	0 days, 0 hrs, 0 min, 5 sec (5 sec 108452 (inst/sec) 15889 (cycle/sec)	)	_cuobjdump_ complete_ output_a4j	_cuobjdump_ complete_ output_Ag	
			-	cuobjdumpcuobjdur completecomplet putput_aG Fatt	mpcuobjdump_ ecomplete xoutput_bE	_cuobjdump_ complete_ output_bS	
			-	cuobjdumpcuobjdur completecomplet	mpcuobidump_ ecom <sup>#</sup> cudag	selected (27.7 kB)	

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 <code>syncthreads()</code>. 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.

<b>Q</b>					
_		< > < project4_template cuda_emu_demo	×	Q = :::	
	😣 🖱 🗉 gpgpu-sim@gpgpusim-VirtualE	Box: ~	#inc #inc		
	<pre>gpgpu-sim@gpgpusim-VirtualBox:~\$ i 1: lo: <loopback,up,lower_up> mtu t</loopback,up,lower_up></pre>	.p addr 65536 qdisc noqueue state UNKNOWN group defaul	cuda.cu	cudag	
	link/loopback 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lc valid lft forever preferrec	0 brd 00:00:00:00:00 )   lft forever	code Fune Zeor	~~	
	inet6 ::1/128 scope host		_cuobjdump_	_cuobjdump_	
	valid_lft forever preferred 2: eth0: <broadcast,multicast,up,l oup default glen 1000</broadcast,multicast,up,l 	_ltt forever .OWER_UP> mtu 1500 qdisc pfifo_fast state UP gr	1.3033	.vei	
	link/ether 00:0c:29:98:c0:0d b	ord ff:ff:ff:ff:ff	_cuobjdump_	_cuobjdump_	
	inet 192.168.201.129/24 brd 19 valid lft forever preferred	2.168.201.255 scope global eth0   lft forever	3.etr	3.ptx	
	inet6 fe80::20c:29ff:fe98:c00c	//64 scope link	Fati	Fati	
	valid_ift forever preferred apapu-sim@apapusim-VirtualBox:~\$	_lft forever	_cuobjdump_	_cuobjdump_	
	SF SF		output_7s	output_8x	
			Fatl	Fati	
			_cuobjdump_	_cuobjdump_	
			complete_	complete_	
and the second second			output_a4j	output_Ag	
and the second			Fatt	Fati	
		_cuobjdumpcuobjdump_ complete complete	_cuobjdump_ complete	_cuobjdump_ complete	
		output_aG output_Ax	output_bE	output_bS	
		Fati	Fati	Fati	
		_cuobjdumpcuobjdump_	_cuobidump	selected (27.7 kP)	
		_cuobjdumpcuobjdump_	_cudad'	" selected (27.7 kB)	

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

	•			makefile — gpgpu-sim [SSH: 192.168.201.128]			
ζh	REMOTE EXPLORER		Enter SSH Connection Command				
	$\sim$ remote	ssh gpgpu-sim@192.168.201.129					
Q	$\sim$ SSH	Press 'Enter' to	confirm your input or 'Escape' to cancel				
1	V 4 192.168.201.128 connected			J=C++11 −y			
9 o (135)	apapu-sim /home						
			4				
	✓ L VM4005		5				
			6	ead —02 —std=c++11			
₽~	✓  ☐ cluster-csc4005		7				
		I Demo Iroot		+11			
		IL_Defile /root					
	root /		10	/include -L/usr/local/lib -L/usr/lib -lglut -lGLU -lGL -lm ·	-DGUI -02 -		
	CSC4005_2022Fal	ll_Demo /root/	11				
	119010355 /nfsmnt		12	lude -L/usr/local/lib -L/usr/lib -lglut -lGLU -lGL -lm -DGU	I -std=c++1		
0	∨ □ <b>+</b> 7		13				

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 <u>119010355@link.cuhk.edu.cn</u>.

Hope it is helpful.

Bokai XU