SE 346 LAB: GPU Architecture

GPU acts as universal accelerators for many applications. GPU architecture is a hot area in architecture research. We will use GPGPU-sim to simulate the architecture of GPU in this lab.

You are encouraged to record down problems your encounter and how you solve them.

Write down your answers and paste important result figures in your report. Please sent your PDF format report to wpybtw@sjtu.edu.cn

Lab0: environment setup

GPGPU-sim leverages CUDA virtual ISA 'PTX' to simulate. Let's setup experiment first. You can setup the environment using virtual machine (this for VirtualBox) or on your physical machine .

TODO: Install CUDA and GPGPU-sim

Refer to this for virtual machine setup.

If you want install on your physical machine, note that master branch only support CUDA version up to 4.0 while dev branch of GPGPU-sim has been tested with CUDA version 4.2, 5.0, 5.5, 6.0 and 7.5, 8.0, 9.0, 9.1. Thus, you should make sure your chosen version of CUDA and GPGPU-sim work with your OS.

There are several tutorials about GPGPU-sim available on internet such as this. Refer to them for more architecture model details.

Lab1: Benchmarking (45%)

TODO:

1. Download Rodinia and build BFS
2. Simulate BFS with GPGPU-sim.
3. What's your L1 data cache miss rate? (15%)
4. What's your average memory fetch latency? (15%)

5. Name several characteristic of BFS based on your results, such as cache, memory and instructions. (15%)

How:

```
wget
http://www.cs.virginia.edu/~kw5na/lava/Rodinia/Packages/Current/rodinia_3.1.tar.bz2
tar jxvf rodinia_3.1.tar.bz2
```

Add 'export CUDA_DIR=/home/gpgpu-sim/cuda/toolkit/4.2/cuda' to 2rd line of ~/rodinia_3.1/cuda/bfs/Makefile

```
cd ~/rodinia_3.1/cuda/bfs/
make
```

Simulate BFS:

```
source ~/gpgpu-sim_distribution/setup_environment
mkdir test; cd test
cp -a ~/gpgpu-sim_distribution/configs/GTX480/* .
~/rodinia_3.1/cuda/bfs/bfs ~/rodinia_3.1/data/bfs/graph4096.txt
```

Lab2: Schedule policy (35%)

GPU warp scheduler is crucial for performance. Schedule policy is one of the hottest topics for GPU. GPGPU-sim comes with three policies, such as **Loose round robin (LRR)**, **Two-level (TL)**, and **Greedy then oldest scheduler (GTO)**.

TODO:

1. Modify '-gpgpu_scheduler' in **gpgpusim.config** to switch schedule policies.
2. Build and simulate 'pathfinder' with parameters ‘ 1000 100 20’ with all three schedule policies.
3. What is the runtime, simulation rate (cycle/sec) for each configuration? (15%)
4. List L1 data cache miss rate, average memory fetch latency of three schedule policies. (20%)

Lab3: Exploration (20%)

Choose one of listed tasks or explore other functions about GPGPU-sim of your interest.

Optional tasks:

- Visualize your results and draw some meaningful figures with AerialVision.py .
- Conduct one of applications in this and characterize it.
- ……