GPGPUsim Report

曹金坤 515260910022

Lab0: environment setup

Have done all tasks as required, following tasks are done in the virtual environment.

Lab1: Benchmarking

1. Download Rodinia and build BFS: **done**
2. Simulate BFS with GPGPU-sim: **done**
3. What's your L1 data cache miss rate? **0.3919**
4. What's your average memory fetch latency? **196**
5. Name several characteristic of BFS based on your results, such as cache, memory and instructions:

- The L2 miss rate is much higher than the L1 miss rate, but the program does not provide as detailed log for L2 cache as that for L1 cache, which is divided into icache, dcache and tcache.
- The instruction cache miss rate is much lower than the data cache miss rate. Besides, for L1 cache there are also a *L1C_cache* and *L1T_cache* (constant cache and texture cache), both of which report much lower miss rate (there may be even no operation occurs on L1T_cache).
- BFS shows stable performance comparing multiple tests on the benchmark
- BFS asks for very imbalanced use for different cores of L1 data cache, some cache core may be accessed more than 10 times more than some others. And their miss rate also varies a lot. Generally speaking, the core used more would have a lower miss rate

Lab2: Schedule Policy

- GTO: runtime is **62sec**, simulation rate is **1058 c/s;**
- LRR: runtime is **61sec**, simulation rate is **1075 c/s;**
- TL: runtime is **73sec**, simulation rate is **899 c/s;**
- GTO: L1 data cache miss rate is **0.568**, average memory fetch latency is **251;**
- LRR: L1 data cache miss rate is **0.568**, average memory fetch latency is **251;**
- TL: L1 data cache miss rate is **0.568**, average memory fetch latency is **252;**

Lab3: Exploration

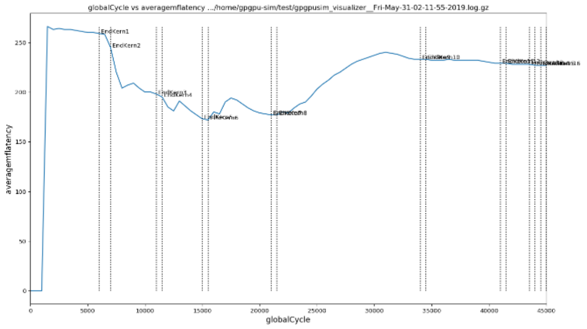
Some problems encountered in my experience:

1. When trying to run **aerialvision.py**, some dependencies should be installed first, while due to some version mismatch issue, we need to install *matplotlib* with the tag "--ignore-installed six" to make it work.
2. We need to modify the code in guiclass.py "from matplotlib import mpl" to "import matplotlib as mpl" to fix an important error.
3. When trying to draw multiple subplots at the same time, an issue rises from time to time that all subplots except for the most above one can not be displayed. I don't know how to reproduce this problem stably and fix it.

In this section, I visualize some running log results with given AerialVision.py script. The results are shown and explained as follows, for which the items on x-axis are always the global cycle count:

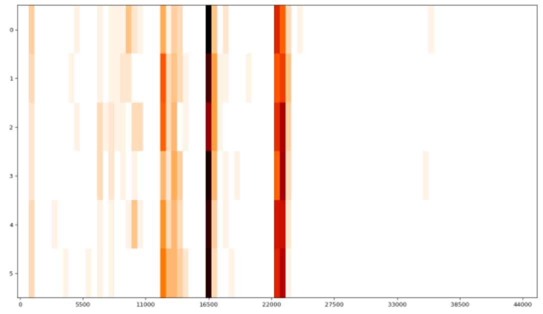
AverageMFLatency

This figure shows the average amount of global cycles that threads waited for off-chip memory requests have been stalled for. It's one of the most important index to measure the effectiveness.



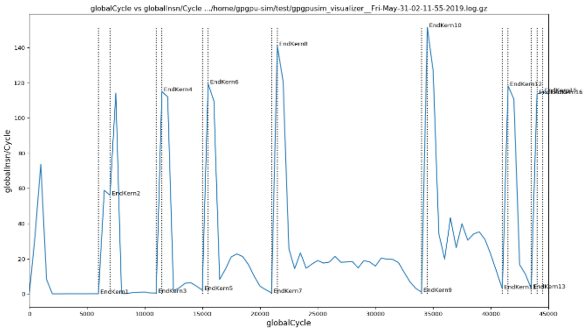
dramUtil

This figure shows the percent of full capacity that each DRAM channel is being utilized. The stronger the color is, the higher ratio of capacity is being utilized. It shows that for different channel, the utilization intensity fluctuate almost always together. We will find that dense channel utilization generally come along an active kernel activity.



globalInsn

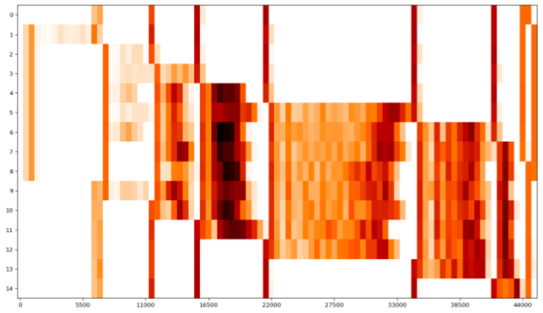
The figure shows the ratio of instruction count that has been executed. The **dy/dx** option is enabled, some the figure shows the tendency.

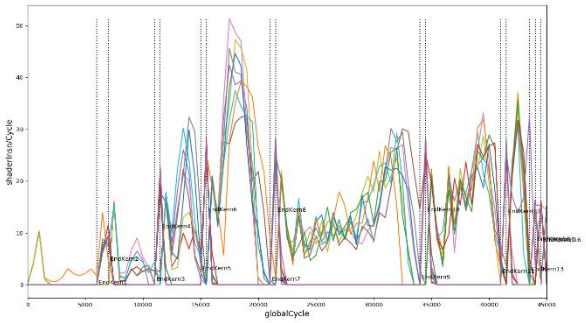


According to the figure, we know that during the booting of each emulated kernel, there is a peak of ipc index. Then, no large amount of memory use is needed after kernel starting, the ipc would come to a much lower level.

shaderInsn

This figure indicates the amount ratio of total number of instructions executed in each shader. Both the intensity plot and line plot are provided.

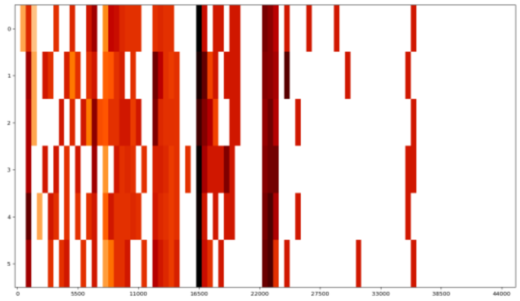




Looking into both the shadeinsn figure and aforementioned globalInsn figure, we can find they fluctuate almost synchronously.

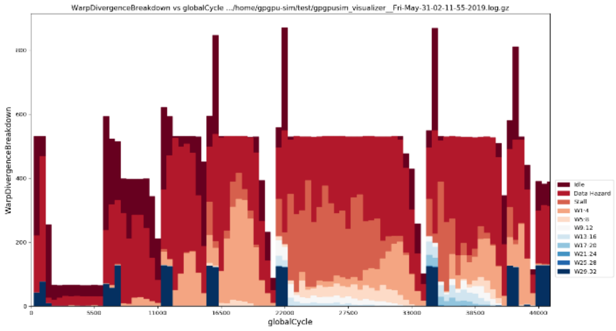
dramEff

The figure shows percent of the full capacity of each DRAM channel is used when there is a pending request at the DRAM channel. Similar as the utilization ratio of each channel, this index also output very good coherence for different DRAM channel.



warpDivergenceBreakdown

A breakdown of the number of warps warp issued for execution during the sampling period according to the number of active threads in the warp. The figure shows the warp breakdown number during monitoring. This figure clearly shows the periodicity of warp use in bfs test.


WarpDivergenceBreakdown vs globalCycle .../home/gpgpu-sim/test/gpgpusim_visualizer__Fri-May-31-02-11-55-2019.log.gz
WarpDivergenceBreakdown
globalCycle
Idle
Data Hazard
Stall
W1:4
W5:8
W9:12
W13:16
W17:20
W21:24
W25:28
W29:32