GPGPU-SIM REPORT

516030910005 陈星伊

Lab 1:

(3)

L1D_total_cache_miss_rate = 0.3234
L1D_total_cache_miss_rate = 0.3388
L1D_total_cache_miss_rate = 0.3806
L1D_total_cache_miss_rate = 0.4407
L1D_total_cache_miss_rate = 0.4034
L1D_total_cache_miss_rate = 0.4529
L1D_total_cache_miss_rate = 0.4064
L1D_total_cache_miss_rate = 0.4202
L1D_total_cache_miss_rate = 0.4535
L1D_total_cache_miss_rate = 0.4578
L1D_total_cache_miss_rate = 0.3944
L1D_total_cache_miss_rate = 0.3973
L1D_total_cache_miss_rate = 0.3927
L1D_total_cache_miss_rate = 0.3924
L1D_total_cache_miss_rate = 0.3922
L1D_total_cache_miss_rate = 0.3919

(4)

averagemflatency = 259
averagemflatency = 220
averagemflatency = 196
averagemflatency = 184
averagemflatency = 172
averagemflatency = 170
averagemflatency = 177
averagemflatency = 177
averagemflatency = 233
averagemflatency = 232
averagemflatency = 229
averagemflatency = 229
averagemflatency = 227
averagemflatency = 227
averagemflatency = 227
averagemflatency = 227

(5)

L1I_total_cache_miss_rate goes down to a steady value as the number of times increases.

L1D_total_cache_miss_rate goes up then goes down to a steady value as the number of times increases.

L2_total_cache_miss_rate goes down to a steady value as the number of times increases.

Averagemflatency goes to a steady value as the number of times increases.

Average row locality goes up as the number of times increases.

Lab 2:

(3)

gto:
gpgpu_simulation_time:8/15/22/28/35 (sec)
simulation rate:1814/1869/1890/1925/ 1875 (cycle/sec)

lrr:
gpgpu_simulation_time:8/14/21/28/34 (sec)
simulation rate:1814/2003/1980/1925/ 1930 (cycle/sec)

tl:
gpgpu_simulation_time:9/19/28/37/46 (sec)

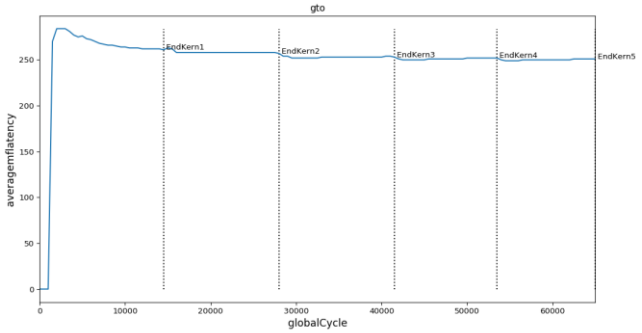
simulation rate:1652/1514/1522/1492/ 1461 (cycle/sec)

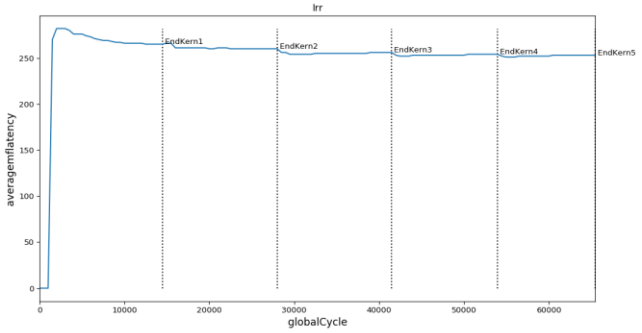
From the statistics, TL differs a lot from the other two policies

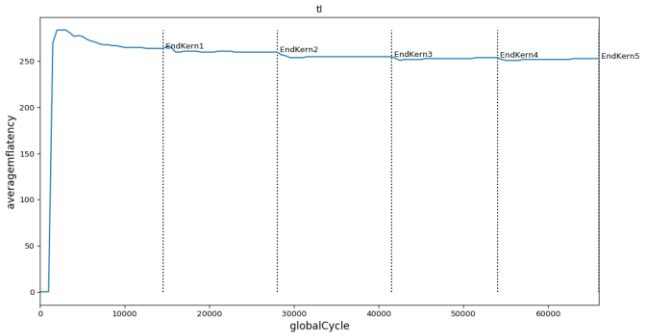
(4)
gto:
L1D_total_cache_miss_rate:0.5653/0.5653/0.5653/0.5653/ 0.5680
averagemflatency:263/254/251/252/251
lrr:
L1D_total_cache_miss_rate:0.5653/0.5653/0.5653/0.5653/ 0.5680
averagemflatency:263/254/251/252/251
tl:
L1D_total_cache_miss_rate:0.5653/0.5653/0.5653/0.5653/ 0.5680
averagemflatency: 264/260/255/253/252

From the statistics, the three policies are similar

Lab3:
This lab takes me plenty of time to check and install the dependencies of the AerialVision.py which has not been installed in the virtual machine.







From the figure, it shows that the average memory fetch latency of three policies are similar and all goes to a steady value.