CS 217 : GPU Architecture and Parallel Programming

Name : S . Hanisha Sree

SID : 862191473

Lab 2

1. For the naive reduction kernel, how many steps execute without divergence? How many steps execute with divergence?

   Ans : 10 steps (5 steps without divergence, 5 steps with divergence)

2. For the optimized reduction kernel, how many steps execute without divergence? How many steps execute with divergence?

   Ans : 10 steps

3. Which kernel performed better? Use profiling statistics to support your claim.

   Ans : Naïve Reduction:

```
GPGPU-Sim uArch: Shader 3 empty (release kernel 1 '_Z14naiveReductionPfS_j').
GPGPU-Sim uArch: GPU detected kernel '_Z14naiveReductionPfS_j' finished on shader 3.
kernel_name = _Z14naiveReductionPfS_j
kernel_launch_uid = 1
gpu_sim_cycle = 126806
gpu_sim_insn = 71024154
gpu_ipc =     560.1009
gpu_tot_sim_cycle = 126806
gpu_tot_sim_insn = 71024154
gpu_tot_ipc =     560.1009
gpu_tot_issued_cta = 0
gpu_stall_dramfull = 2686
gpu_stall_icnt2sh    = 11024
gpu_total_sim_rate=496672
```

Optimized Reduction :

```
GPGPU-Sim uArch: GPU detected kernel '_Z18optimizedReductionPfS_j' finished on shader 14.
kernel_name = _Z18optimizedReductionPfS_j
kernel_launch_uid = 1
gpu_sim_cycle = 89134
gpu_sim_insn = 62024030
gpu_ipc =     695.8516
gpu_tot_sim_cycle = 89134
gpu_tot_sim_insn = 62024030
gpu_tot_ipc =     695.8516
gpu_tot_issued_cta = 0
gpu_stall_dramfull = 5337
gpu_stall_icnt2sh    = 37314
gpu_total_sim_rate=614099
```

On comparison Optimized reduction has better performance.

4. How does the warp occupancy distribution compare between the two Reduction implementations?

   Ans : Naïve Reduction:

```
Warp Occupancy Distribution:
Stall:146546   W0_Idle:56748   W0_Scoreboard:315050   W1:369306      W2:187584      W3:0   W4:187584      W5:0   W6:0    W7:0   W8:187584      W9:0   W10:0

W11:0   W12:0   W13:0   W14:0   W15:0   W16:187584     W17:0   W18:0   W19:0   W20:0   W21:0   W22:0   W23:0   W24:0   W25:0   W26:0   W27:0   W28:0   W29:0   W30:0   W31:0
```

   Optimized Redcution :

```
Warp Occupancy Distribution:
Stall:92060     W0_Idle:98762   W0_Scoreboard:385670    W1:13678        W2:7816 W3:0    W4:7816 W5:0    W6:0    W7:0   W8:7816
9:0     W10:0   W11:0   W12:0   W13:0   W14:0   W15:0   W16:7816        W17:0   W18:0   W19:0   W20:0   W21:0   W22:0  W23:0
24:0    W25:0   W26:0   W27:0   W28:0   W29:0   W30:0   W31:0   W32:2024274
traffic_breakdown_coretomem[CONST_ACC_R] = 120 {8:15,}
```

   On comparison Optimized reduction has better performance.

5. Why do GPGPUs suffer from warp divergence?

   Ans : Because amount of data processed is far greater than the needs of scheduling, they are serialized and it will lower the performance. Thus, GPGPU’s suffer from warp divergence.