

Chandler Bottomley
Github : Chabo17

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

The first four steps will execute without divergence. Then the final five steps will diverge.

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

The optimized kernel will execute four steps before divergence because the size will be too small for a warp. Then the last 5 steps will execute with divergence.

3. Why do GPGPUs suffer from warp divergence?

GPUS suffer from warp divergence because they are made run in parallel for maximum efficiency. We run into the problem where not all code is parallel so we end up getting warp divergence in the threads. Also we can run into situations where we don't execute all the threads.

4. Which kernel performed better? Implement both naive and optimized code to compare the timing of each kernel.

The optimized kernel ran faster because there is less divergence so we end up getting a better efficiency with the optimized kernel.