Questions

1. What is control flow divergence?

In CUDA, threads work in warps of 32 and all of the threads in a warp execute the same task (instruction) concurrently. If threads in the same warp have to do different tasks, it is called control flow (warp) divergence. In NVIDIA graphical processing units, there are instructions that are executed if a conditional flag is true. All threads in CUDA execute all conditional branches so the cost doubles, leading to a loss of performance. Because of this, the NVIDIA compiler checks if all threads in a warp take the same conditional branch, and does warp voting. Every warp chooses the most efficient branch and in some cases, at compilation, all of the threads in a warp may go to the same branch. Different threads executing different warps and all threads going to the same branch causes a huge performance decrease.

1. How can we create a dynamic sized shared memory?

Dynamically allocated shared memory is used when the amount of shared memory I not known a priori. To allocate dynamic sized shared memory, an optional third configuration parameter is specified when calling a kernel function that uses dynamically sized shared memory:

someFunction<<<1, n, n\*sizeof(float)>>>(arr, n);

Dynamic sized shared memory is allocated as in the below example:

extern \_\_shared\_\_ int s[];

Here, extern specifies that the array is shared and the size of the array comes from the third configuration parameter, which is a size input that is calculated during compile time.

1. How can we use shared memory to accelerate our code?

Using shared memory is much faster than the global memory that is not cached. This is due to the fact that shared memory is allocated per thread block, so all of the threads in one block utilize the same memory, causing increased locality and controlled data caches, high performance cooperative parallel algorithms such as parallel reductions.

1. Which CUDA operations give us device properties? To answer this question you should write a simple program and query the device properties of the machine you are working with.
2. What are the necessary compiler options in order to use atomic operations?