

HW4 Write Up

scan.cu

The scan algorithm required the creation of 3 separate kernels. I began with writing the exclusive scan that scans an entire thread block of (1024) and saves the chunks to the array maxValues. It uses two buffers to eliminate conflicts and a shared memory array called partial that holds each partial result. The next kernel scans each chunk saved in the maxVals array and uses the same algorithm to find each max value in each chunk and save it to the output. Then the last kernel is responsible for performing the final addition operation of the inputs and the maxValues array. Last, I launched each individual kernel with its appropriate block and grid size while passing the outputs along through each kernel. Then I copied the output from the device back to the host.

spmv.cu

For the sparse matrix vector multiplication, I started with creating a thread block size of 32. I then created a stride variable that started at half of the thread block size (in the middle). I then created two bounds a minimum and a maximum bound. The minBound was initialized to the pointer vector at blockIdx.x and the maxBound at blockIdx.x + 1. Then I used two for loops, one to compute the multiplication between the matrix and the dense vector and the other to accumulate the partial sum that I then used thread 0 to save to the output vector. Lastly, I performed the necessary memory allocations for inputs, outputs, indices, dense vector, and the pointer vector. I launched the kernel with the block size being the thread block size previously created and the grid size being the number of rows.