# Assignment 4 - Andrew Chan

This assignmnet looks at different GPU parallel reduction methods to compute the maximum assignment mark among student records. 4 implementation were evaluated, a global atomic based reduction, a recursive global memory reduction, a shared memory block reduction, and a warp shuffle reduction. Performance and scalability were also analyzed using both small and large datasets.

In Task 0 records were converted from an Arroy of Structures layout to a structure of arrays layout to improve memory coalescing on the GPU. This also allows consecutive threads to access contiguous memory locations which helped improve global memory efficiency.

In Task 1, each thread loads one record and also attempts to update a global maximum using a spinlock implemented with `atomicCAS`. Only one thread may enter the critical section at a time which results in a serialization under high contention.

In Task 2, the dataset is iteratively halved on the GPU. Each kernel invocation compares adjacent pairs and writes the local maximum to a smaller output array. This then continues until the problem size is reduced to one block.

In Rask 3, each block performs an in block tree reduction using shared memory and synchronization. Only one value per block is written to global memory.

In Task 4, warp level intrinsics were used to perform reductions entirely in registers without shared memory or block wide synchronization.

The results are as followed with the small dataset: Atomics: ~5 ms
Recursive: ~0.07 ms
SM: ~0.04 ms
Shuffle: ~0.04 ms

Large dataset: Atomics: ~210,000 ms
Recursive: ~2.3 ms
SM: ~1.15 ms
Shuffle: ~1.21 ms

The atomic implementation uses a single global lock. This means when multiple threads attempt to update the maximum simultaneously, they serialize through the spinlock. The the larger dataset threads, this creates extreme contention and reduces the algorithm to near sequential execution.

These results show tht naive globl atomic synchronization does not scale for large datasets due to contention induced serialization. In contrast, hierarchial tree based reductions achieve near constant runtime growth by exploiting intra block and intra warp parallelism. Warp level reductions offer comparable performance to shared memory approaches while reducing synchronization overhead.