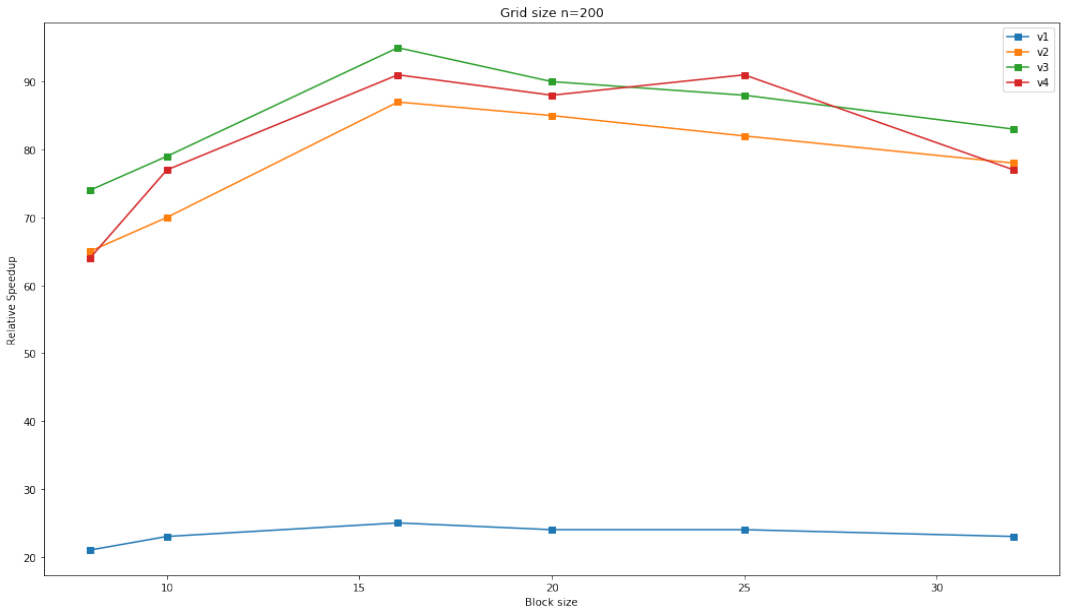
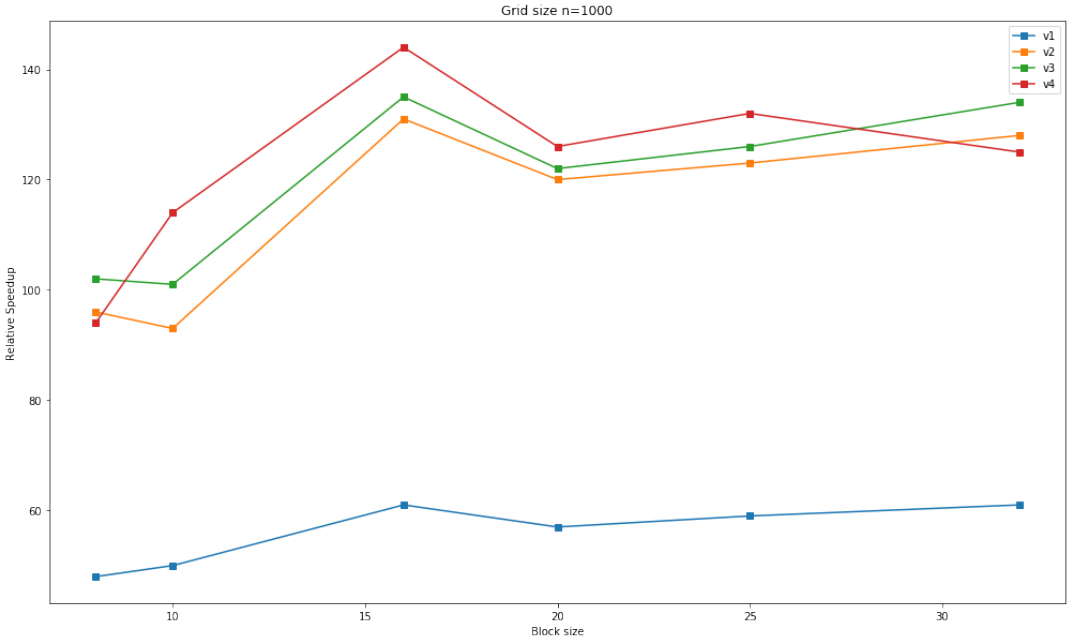
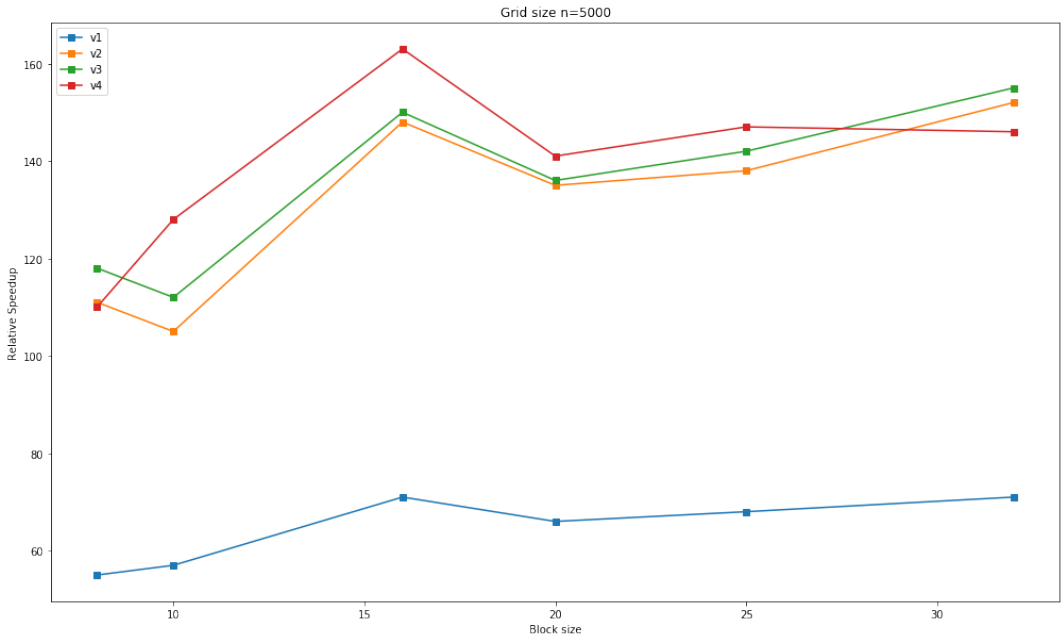
Boyao Zhu & Jiaxin Yang
CMSE 822
HW 7
Due Dec 13

During our performance test, we vary the grid size (n) and block size while fixing the number of iterations as 4000. The results are shown below.







There are mainly three noticing points. Firstly, we see great improvement of speedup from v1 to v2. It is because we reduce kernel initiation overheads by fusing everything into single kernel. However, the differences between v2, v3 and v4 are less obvious. The optimizations of v3 and v4 are on registers and shared memories. The tread may indicate that the kernel is compute-bound.

Secondly, we see all versions benefit from increasing grid size (n), regardless of 200, 1000 or 5000. Generally, the curves shift upward in all figures above. It is due to the fact that with larger grid size (n), computation dominates and kernel initiation overheads become a small part.

Finally, we see that performance is best with block size 16 by 16. By checking in CUDA calculator, we see that 16 by 16 block has 100% device occupancy, where it exhausts several warps (multiple of 32). For other block size except 32 by 32 case, they don't meet all two features that 16 by 16 block owns. In situation of 32 by 32 block, although those two condition are met, it does not perform as well as 16 by 16, at least in grid size (n=200). So there may exist some unknown factors beyond our consideration.

As a whole, with the help of GPU and comprehensive consideration, we can achieve great accelerations of some tasks with supreme efficiencies.

PS: I combined four versions into one, with explanation on the top(how to run). Also I submitted 4 versions to git in case the combined version doesn't work.