Why this version of the code using shared memory will have better performance compared to earlier versions. ?

When accessing multidimensional arrays it is often necessary for threads to index the higher dimensions of the array, so strided global memory access is simply unavoidable. We can handle such cases using shared memory.

Shared memory is much faster than local and global memory as it is on-chip . Shared memory is allocated per thread block, so all threads in the block have access to shared memory. Because shared memory is shared by threads in a thread block, it provides a mechanism for threads to cooperate (thread synchronization). In reversing the array using shared memory we were able to have all global memory reads and writes performed with single stride. So we can say that the shared memory version is faster than their previous versions.

Reference

1. Using Shared memory in cuda, <https://devblogs.nvidia.com/using-shared-memory-cuda-cc/>