NVIDIA GPU Memory Structures :

- Each SIMD Lane in a multithreaded SIMD Processor is given a private section of off-chip DRAM, which we call the Private Memory. It is used for the stack frame, for spilling registers, and for private variables that don't fit in the registers. SIMD Lanes do not share Private Memories. Recent GPUs cache this Private Memory in the L1 and L2 caches to aid register spilling and to speed up function calls.
- We call the on-chip memory that is local to each multithreaded SIMD Processor Local Memory. It is shared by the SIMD Lanes within a multithreaded SIMD Processor, but this memory is not shared between multithreaded SIMD Processors.
- The multithreaded SIMD Processor dynamically allocates portions of the Local Memory to a thread block when it creates the thread block, and frees the memory when all the threads of the thread block exit. That portion of Local Memory is private to that thread block.
- Finally, we call the off-chip DRAM shared by the whole GPU and all thread blocks GPU Memory. Our vector multiply example only used GPU Memory.

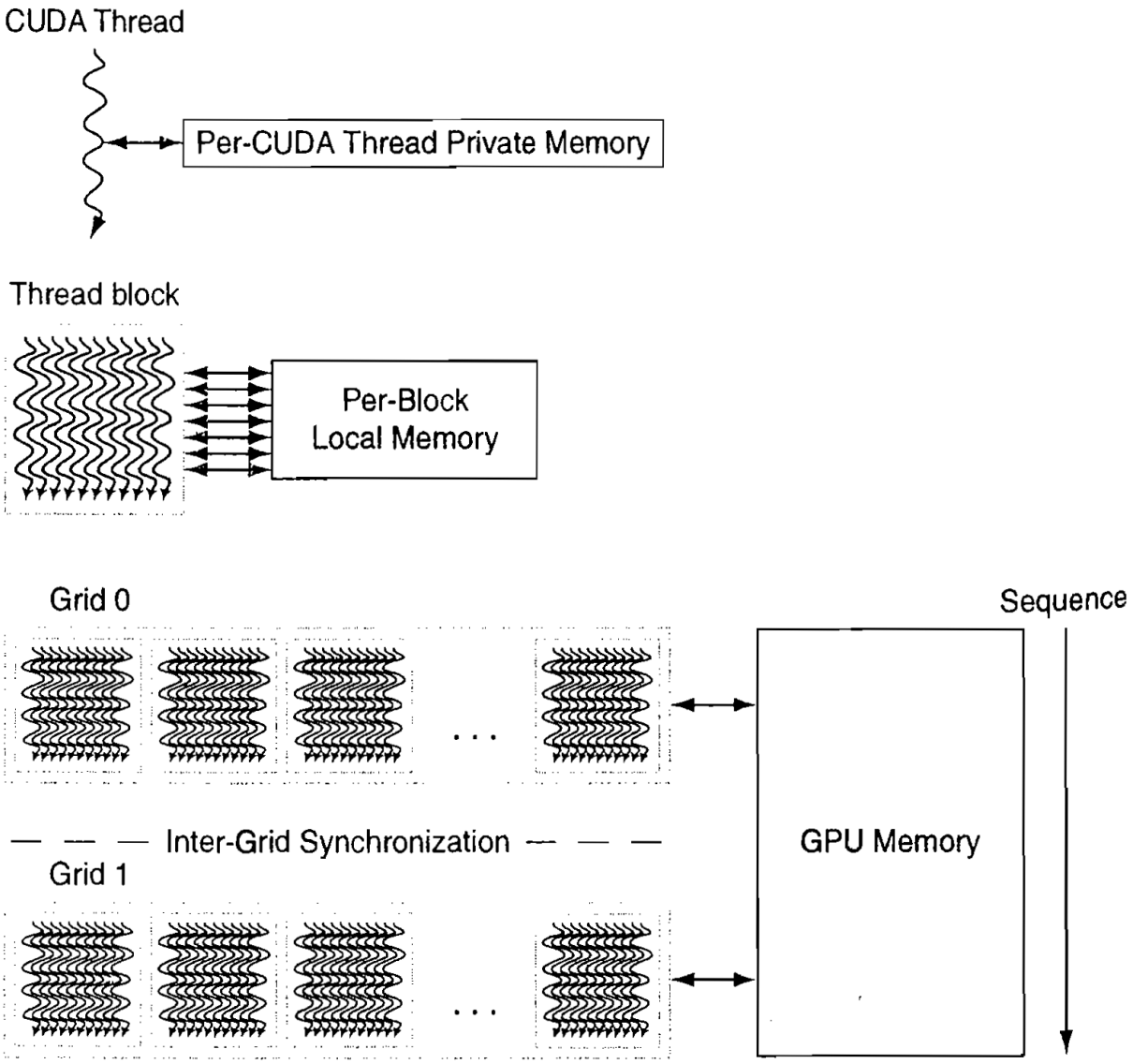

Fig: **GPU Memory Structure**

- The. system processor, called the host, can read or write GPU Memory. Local Memory is unavailable to the host, as it is private to each multithreaded SIMD processor. Private Memories are unavailable to the host as well.
- Rather than rely on large caches to contain the whole working sets of an application, GPUs traditionally use. smaller streaming caches and rely on extensive multithreading of threads of SIMD instructions to hide the long latency to DRAM since their working sets can be hundreds of megabytes.

- Given the use of multithreading to hide DRAM latency, the chip area used for caches in system processors is spent instead on computing resources and on the large number of registers to hold the state of many threads of SIMD instructions.
- The recent Fermi architecture has added caches, but they are thought of as either bandwidth filters to reduce demands on GPU Memory or as accelerators for the few variables whose latency cannot be hidden by multithreading.