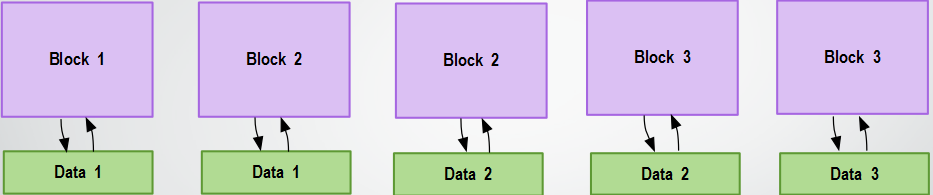
**Static Shared Memory: Exploiting Data Persistence in GPU Shared Memory**

**Motivation and Idea:**

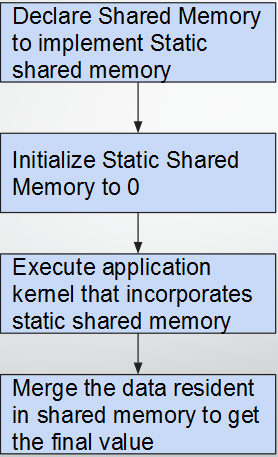
* Traditional GPU programming model dictates that data in shared memory only persist through lifetime of a thread block.
* When data is shared across multiple blocks, shared memory must be initialized at beginning after a thread block terminates and contents must be read from global to shared.
* To avoid the overhead, we have made a novel discovery that that data is not lost when thread block terminates.



T1 T2 T3 T4 T5

* Above is the proposed model of Static shared memory, At first block 1 is initialized and at T1 valid data is generated, At T2 Block 1 exits and block 2 is scheduled but data is still available from block 1 and can be used by block2.Further, it can be seen that persistent data is used.

**Algorithm:**



**Result:**

* **On Histogram and Reduction programs speedups of approx. 10.6x was observed when compared to** [Persistent threads](http://on-demand.gputechconf.com/gtc/2012/presentations/S0157-Persistent-Threads-Style-Programming-Model-for-GPU-Computing.pdf)**.**
* **On N-body and Linear regression we observed speedup of approximately 2x. We had to hand tune the GPU configurations. Further optimizations can be done to have greater speedups.**