Events

```
cudaEvent_t   start, stop;
cudaEventCreate(&start, &stop)
cudaEventRecord(start,0) // save here
.
.
.
cudaEventRecord(stop,0) // done on the gpu
cudaEventSynchronize(stop)
cudaEventElapsedTime(float, start, stop)

cudaEventDestroy()
```

Constant Memory

__constant__ int …
- a value read from constant memory is broadcasted by a thread to the other threads in its half-warp
- it is also cached, so subbsequent reads are fast

Registers/Local Memory

- if the number of rigs / thread is exccedeed, the remaining variables are split into off chip

How to get how many registers are you using CUDA:
mvcc -Xptas -v -arch=sm-20 pcu
+ how many registers are available
=>
the number of threads you can use

Private OpenCL / Shared memory CUDA

- as a holding place for frequently used data to avoid global memory access
- used as a mean of communication between threads in the same block

```
__shared__ int a[10];
extern __shared__ int a[ ];
<<<B[locks],T[hreads],SharedMemorySize>>> // if we want to allocate memory dinamically when we don't know the size of a
```

Shared memory is divided in 16 or 32 banks

1 2 3 4 5 6 7 8 9 10 11 …16