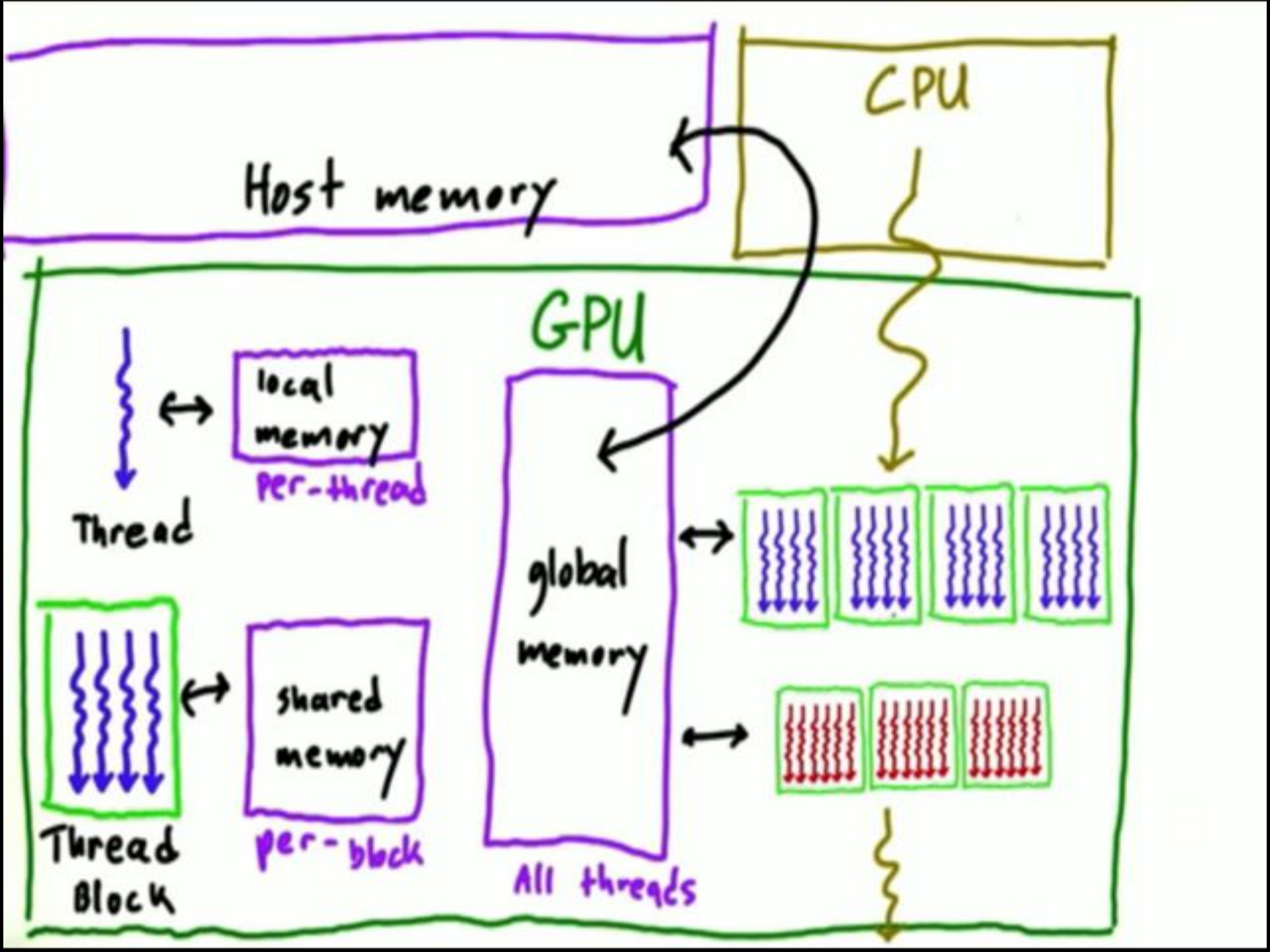
5 Memoria Compartida

Tipos de memoria:
-Local
-Compartida
-Global
-Memoria del CPU(host)


Host memory
CPU
GPU
local memory
per-thread
Thread
global memory
shared memory
per-block
Thread Block
All threads

5 Memoria Compartida

Velocidades de acceso a memoria

local > compartida >> global >>> CPU

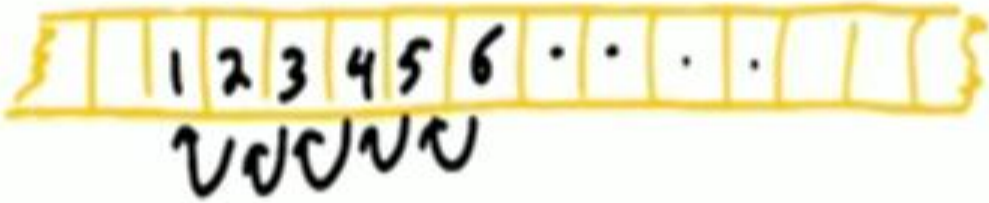
5 Memoria Compartida

Desventajas de usar memoria compartida:

*Varios hilos pueden acceder a la misma localidad y modificarla antes de que otro hilo termine, usando información “desactualizada”

The need for barriers

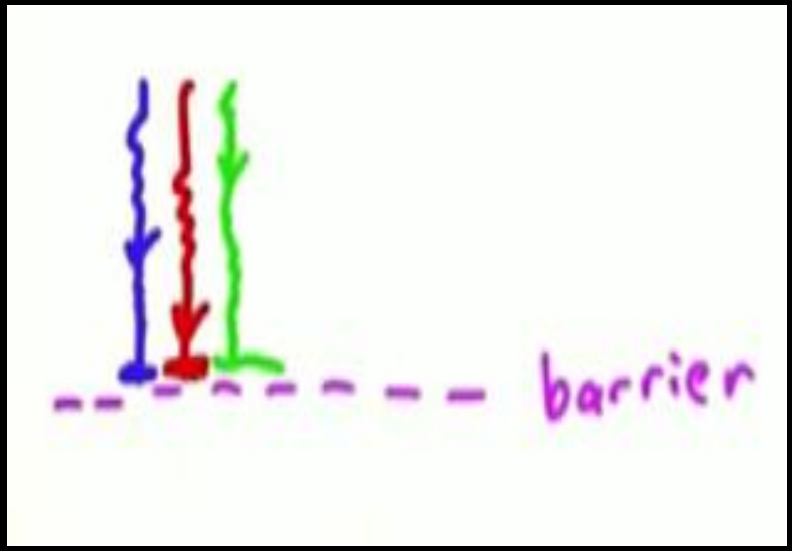
```
:
int idx = threadIdx.x

__shared__ int array[128];

array[idx] = threadIdx.x;

if (idx < 127)
    array[idx] = array[idx+1];

:
```



5 Memoria Compartida

Solución:

*Barreras (sincronizar hilos)



```
int idx = threadIdx.x;

__shared__ int array[128];

array[idx] = threadIdx.x;
__syncthreads();
if (idx < 127) {
    int temp = array[idx+1];
    __syncthreads();
    array[idx] = temp;
    __syncthreads();
```

5 Memoria Compartida

Cual(es) de los siguientes códigos es(son) correcto(s)?

a. s[i] = s[i-1] ;

a. if(i%2)
   s[i] = s[i-1];

a. s[i] = (s[i-1] + s[i] + s[i+1]) / 3.0;

5 Memoria Compartida

Cual(es) de los siguientes códigos es(son) correcto(s)?

a. s[i] = s[i-1] ; Es lo mismo que el ejemplo

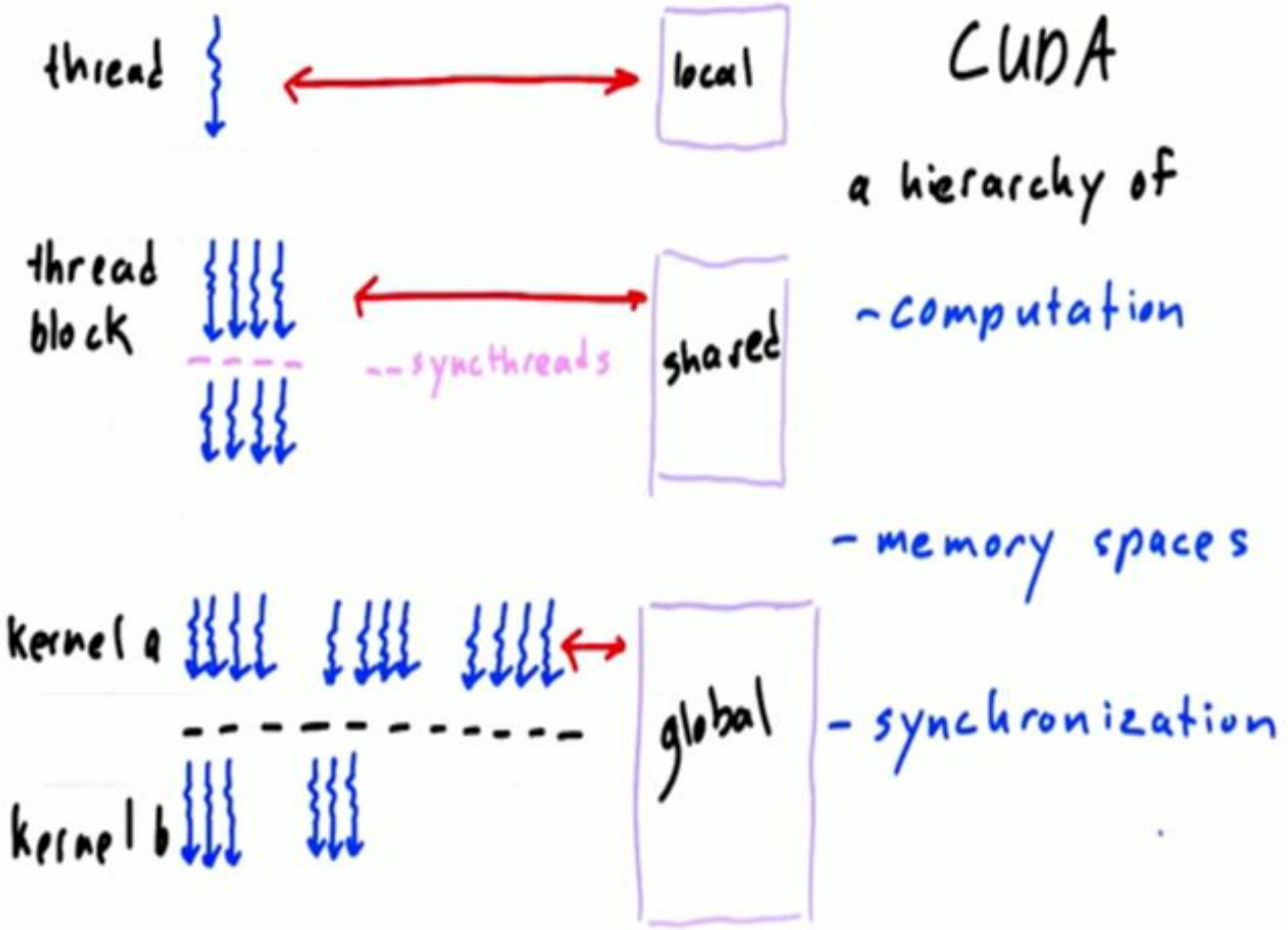
a. if(i%2)
   s[i] = s[i-1]; Lee memoria impar y guarda en memoria par

a. s[i] = (s[i-1] + s[i] + s[i+1]) / 3.0; Lee tres localidades y despues escribe

5 Memoria Compartida

1. Realizar un programa que realice el producto punto de dos vectores, usando memoria compartida(no global)
   a(ax, ay, az) . b(bx, by, bz) = ax*bx + ay*by + az*bz

1. Realizar un programa que realice



CUDA

a hierarchy of

- computation
- memory spaces
- synchronization

Alexander Ilyushin
@chronum

Follow

Multithreaded programming - theory and practice.