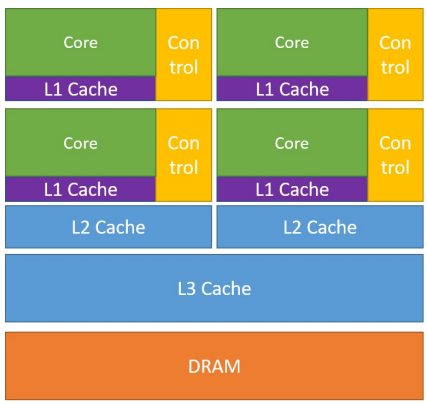
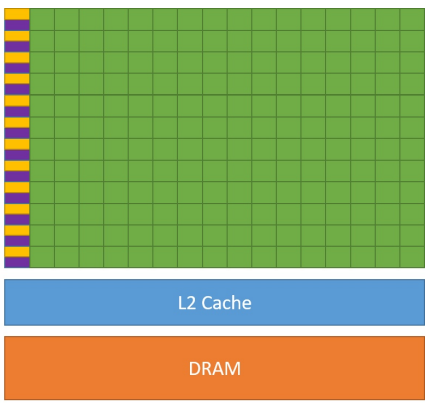
How does a GPU work?



CPU

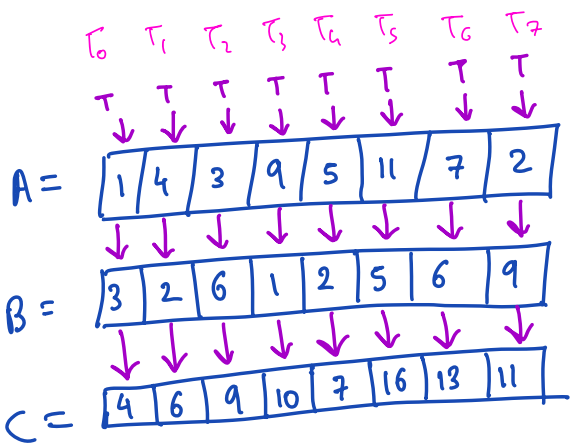
OPTIMIZED FOR REDUCING LATENCY



GPU

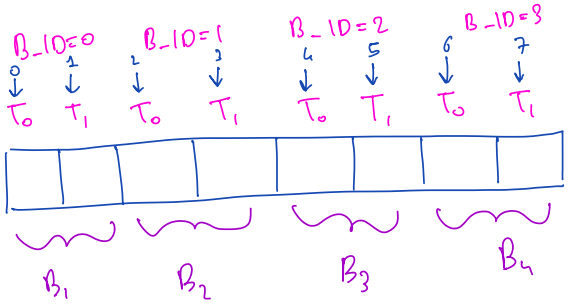
OPTIMIZED FOR MAXIMIZING THROUGHPUT

Vector addition




```cuda
__global__ void cuda_vector_add_single_block(int *out, int *a, int *b, int n)
{
    int i = threadIdx.x;
    if (i < n)
    {
        out[i] = a[i] + b[i];
    }
}
```

```cuda
// run the kernel without blocks
cuda_vector_add_single_block<<<1, N>>>(d_out, d_a, d_b, N);
```



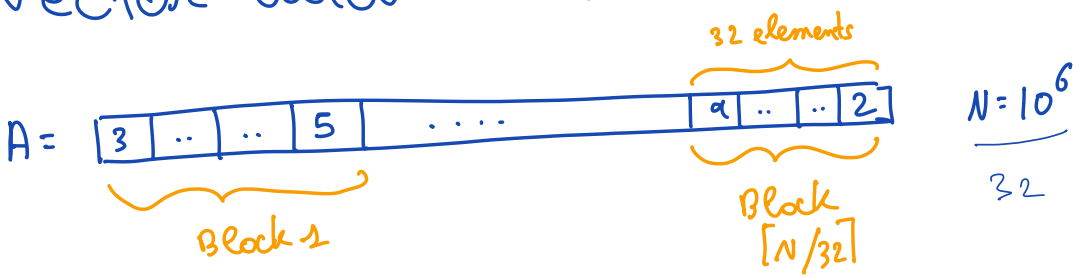
N = 8

4 cores

$$Num\_Blocks = \frac{N}{2 \rightarrow Block\_Size} = 4$$

element_id = i = B_ID × Block_Size + Tid

Vector addition with blocks



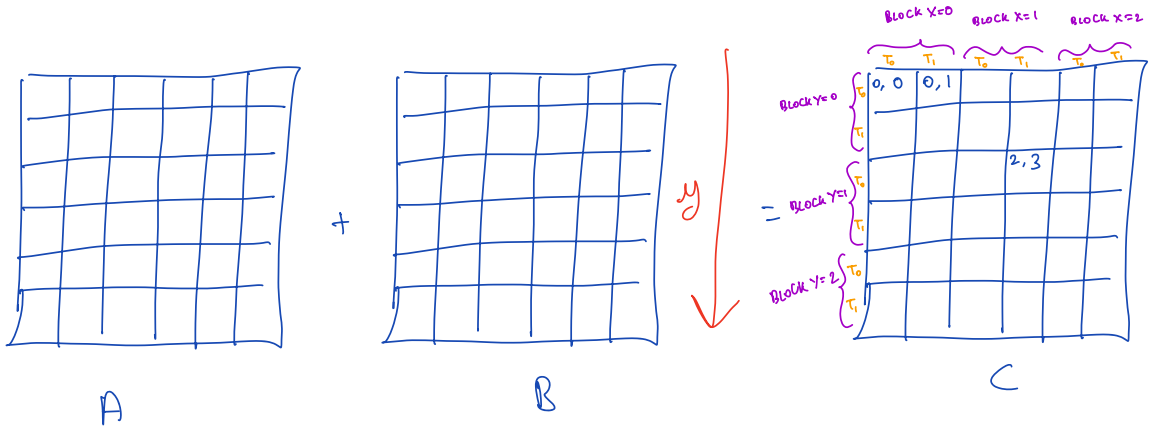
$N = 10^6$

32

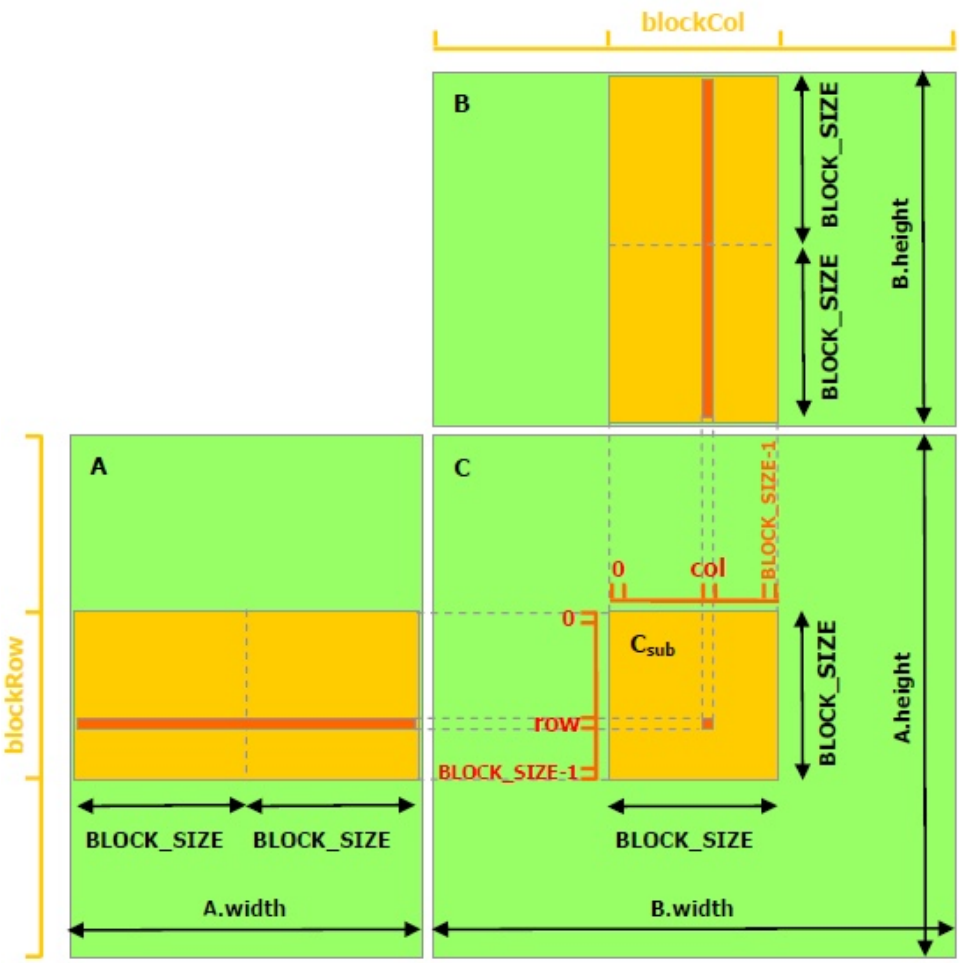
In this case we will have $\lceil \frac{N}{32} \rceil$ blocks

Block size = 32

Matrix addition with blocks


A
+
B
y
=
C
Block X=0
Block X=1
Block X=2
T0
T1
Block Y=0
Block Y=1
Block Y=2
0, 0
0, 1
2, 3

Shared memory



Many more topics

- Control divergence
- Occupancy
- Thread synchronization