

# Matrix Multiplication: $O(n^3)$ Complexity and GPU Parallelization

## Matrix Multiplication Mechanics

Matrix A [4×4]

2	1	3	4
1	2	1	3
3	1	2	1
2	3	1	2

Matrix B [4×4]

1	2	1	3
3	1	2	1
2	1	3	2
1	3	1	2

×

Result C [4×4]

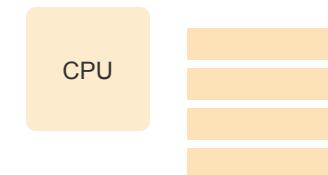
16	18	15	19
12	10	11	12
12	12	12	15
14	12	13	15

=

Each element:  $C[i,j] = \sum A[i,k] \times B[k,j]$

## GPU Parallelization

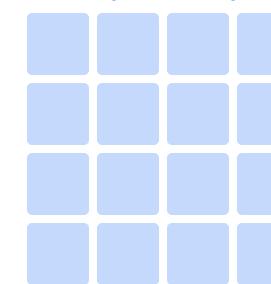
CPU (Sequential)



128 time steps

One at a time

GPU (Parallel)



8 time steps

16 operations simultaneously

**16x faster**

## Total Operations: 2mkn FLOPs

For  $n \times n$  matrices:  $O(n^3)$

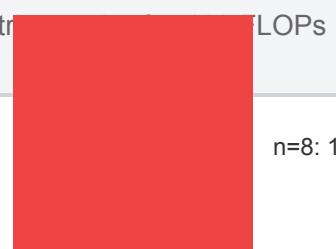
Example: 4×4 matrix multiplication: 256 FLOPs

Cubic Growth:

n=2: 16 FLOPs



n=4: 128 FLOPs



n=8: 1,024 FLOPs

Doubling size → 8x computation

## Modern GPUs: 1000s of parallel operations

A100 GPU: 312 TFLOPS

= 312 trillion operations/second

## Key Insight: $O(n^3)$ Complexity

- Each element requires  $k$  multiply-adds
- For  $n \times n$  matrices:  $2n^3$  total operations
- GPU parallelization provides 1000x speedup
- Critical for deep learning training