

Matrix Multiplication: O(n³) 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

×

=

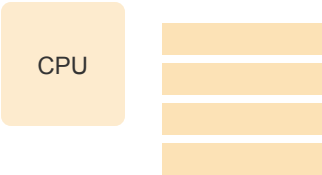
Result C [4×4]

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

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

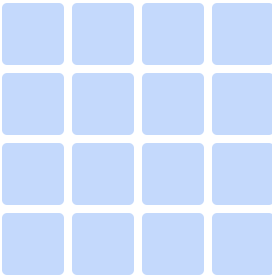
GPU Parallelization

CPU (Sequential)



128 time steps
One at a time

GPU (Parallel)



8 time steps
16 operations simultaneously

16× faster

Complexity Analysis

Total Operations: 2mkn FLOPs

For n×n matrices: O(n³)

Example: 4×4 matrices: 128 FLOPs

Cubic Growth:



n=2: 16 FLOPs



n=4: 128 FLOPs



n=8: 1,024 FLOPs

Doubling size → 8× computation

Modern GPUs: 1000s of parallel operations

A100 GPU: 312 TFLOPS

= 312 trillion operations/second

Key Insight: O(n³) Complexity

- Each element requires k multiply-adds
- For n×n matrices: 2n³ total operations
- GPU parallelization provides 1000× speedup
- Critical for deep learning training