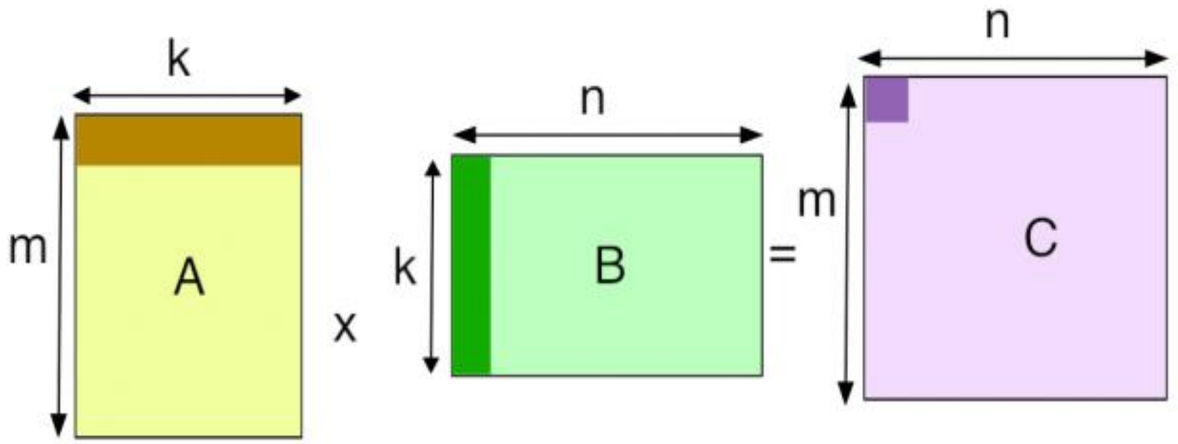
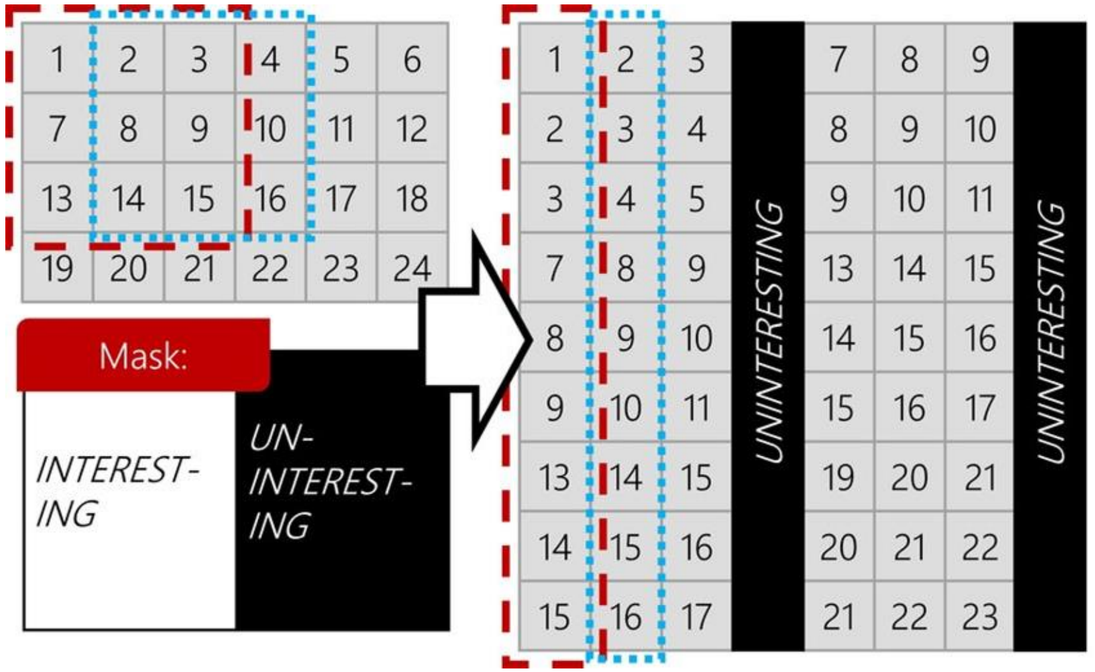
RICE UNIVERSITY

Week-2: GEMM Operation Optimization

Motivation: Why GEMM Matters

- GEMM dominates compute workloads in ML and scientific computing.
- Performance gains in GEMM directly impact model training and HPC efficiency.
- Optimized GEMM enables high GPU utilization and energy efficiency.





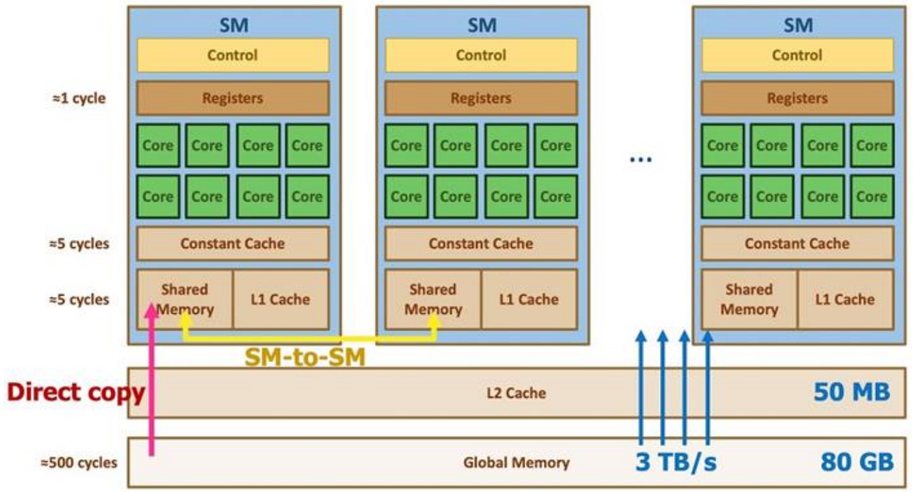
What is GEMM?

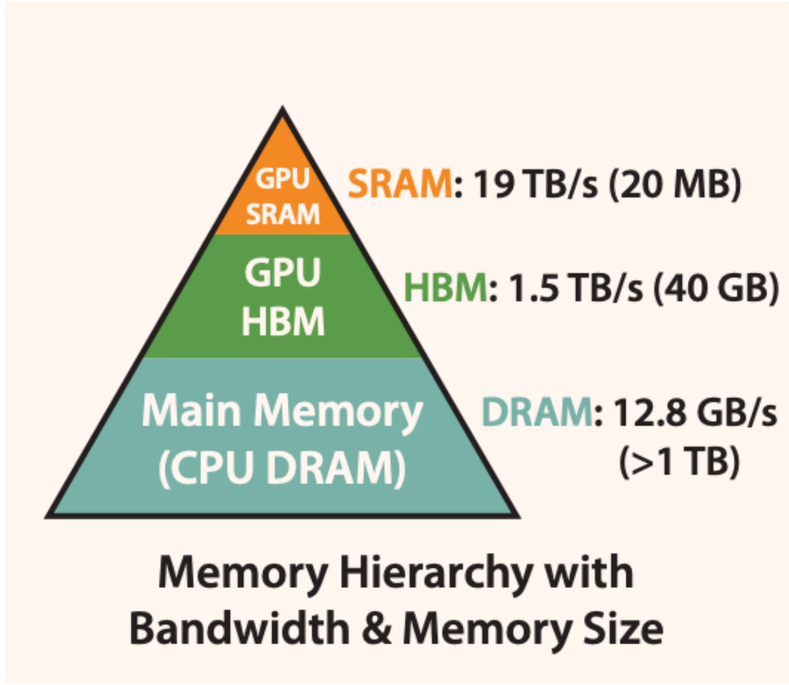
- Definition: C = αAB + βC (General Matrix-Matrix Multiplication).
- BLAS Level-3 routine, foundation for many numerical computations.
- Key operation in deep learning frameworks and HPC systems.

BLAS subprograms		Semantic	FP ops.	Mem. ops.	Ratio
Level 1	Vector Addition	$y_i = x_i + y_i$	n	$3n$	1:3
	Vector Scaling	$x_i = sx_i$	n	$2n$	1:2
	Dot Product	$s = \sum_{i=0}^{n-1} x_i y_i$	$2n$	$2n$	2:2
Level 2	Matrix-vector multiplication	$y_i = y_i + \sum_{j=0} a_{ij} x_j$	$2n^2+n$	n^2+3n	2:1
	Rank-one update	$a_{ij} = a_{ij} + x_i y_j$	$2n^2$	$2n^2+2n$	2:2
Level 3	Matrix-Matrix multiplication	$c_{ij} = c_{ij} + \sum_{k=0}^{n-1} a_{ik} b_{kj}$	$2n^3+n^2$	$4n^2$	n:2

GEMM and GPU Memory Hierarchy

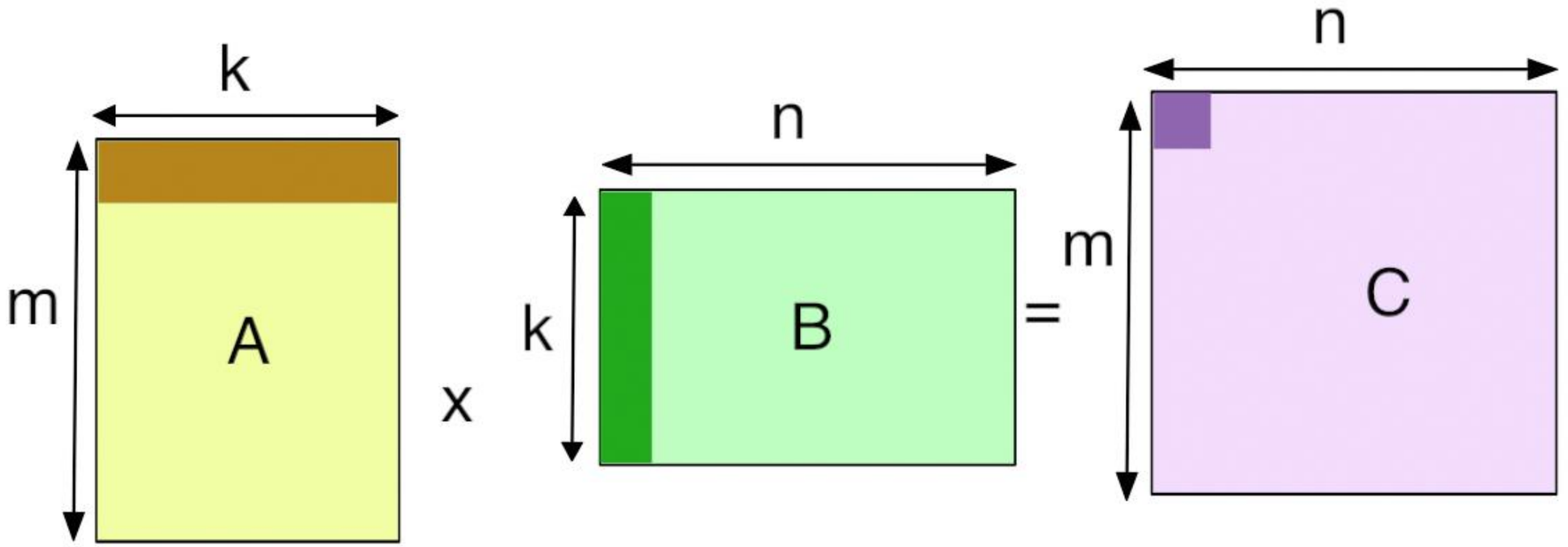
- GPU architecture includes multiple memory levels: global, shared, registers.
- Bottlenecks often stem from memory access rather than arithmetic.
- Optimization focuses on reuse and minimizing bandwidth waste.





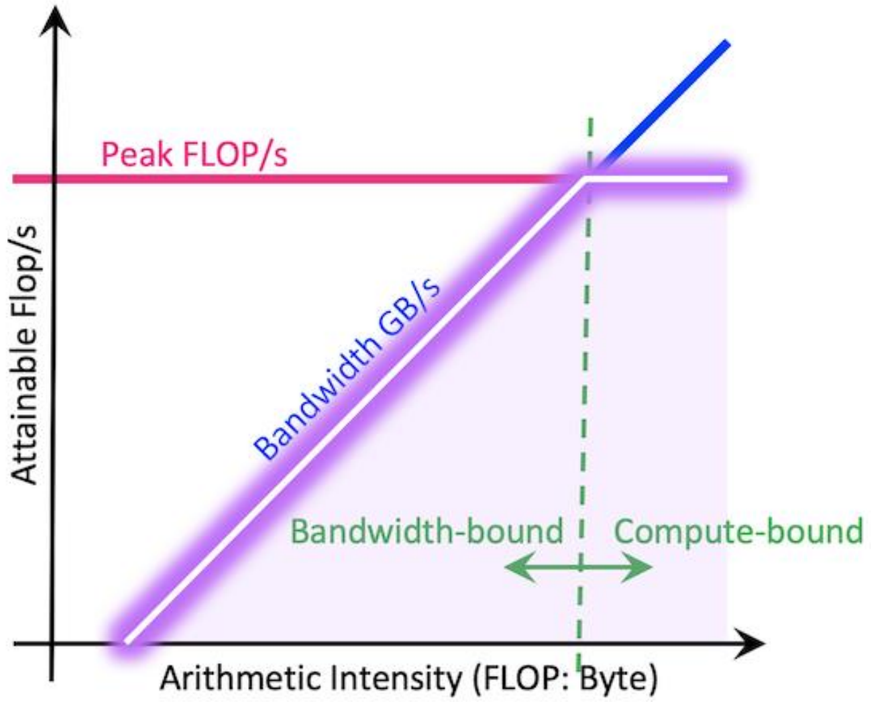
Naïve GEMM Kernel

- Each thread computes one output element.
- Triple nested loop, redundant memory access from global memory.
- Performs only ~1–2% of cuBLAS throughput.



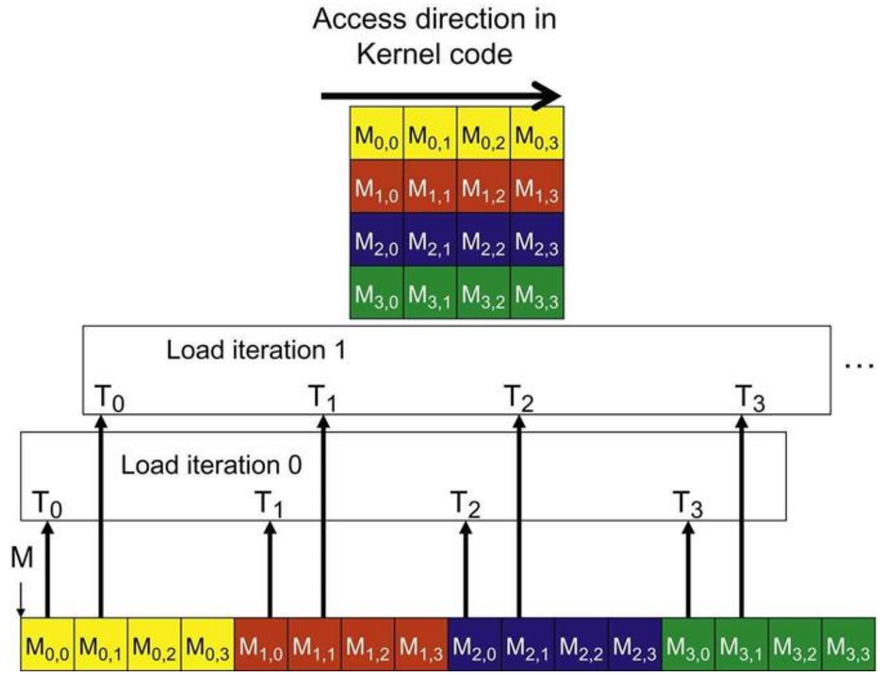
Performance Metrics

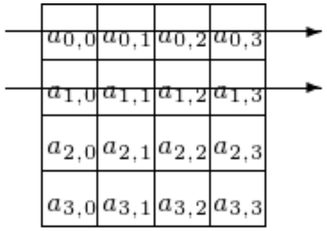
- FLOPs per second (GFLOPS).
- Arithmetic intensity: FLOPs / memory bytes loaded.
- Occupancy, memory coalescing, shared memory utilization.

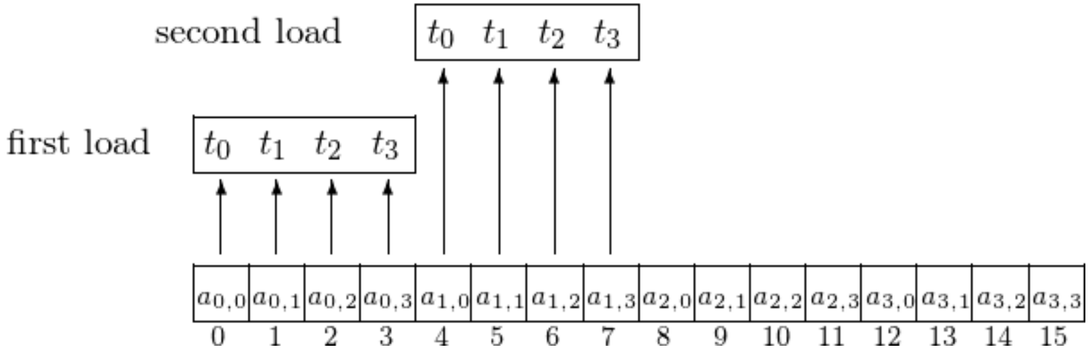


Memory Coalescing

- Threads should access contiguous memory addresses.
- Non-coalesced accesses reduce effective memory bandwidth.
- Align and batch memory loads (vectorized access).

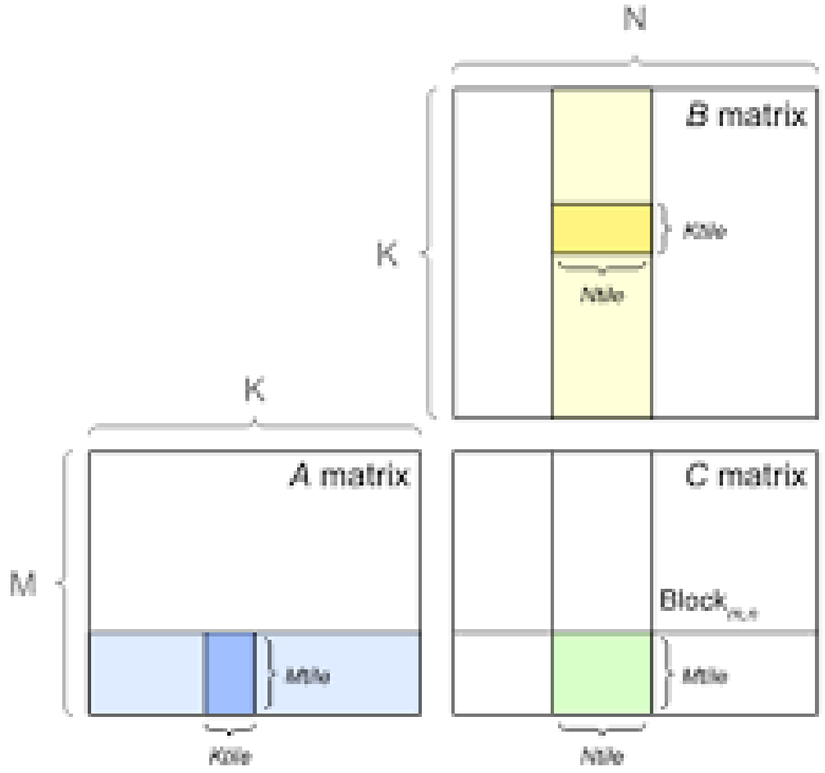


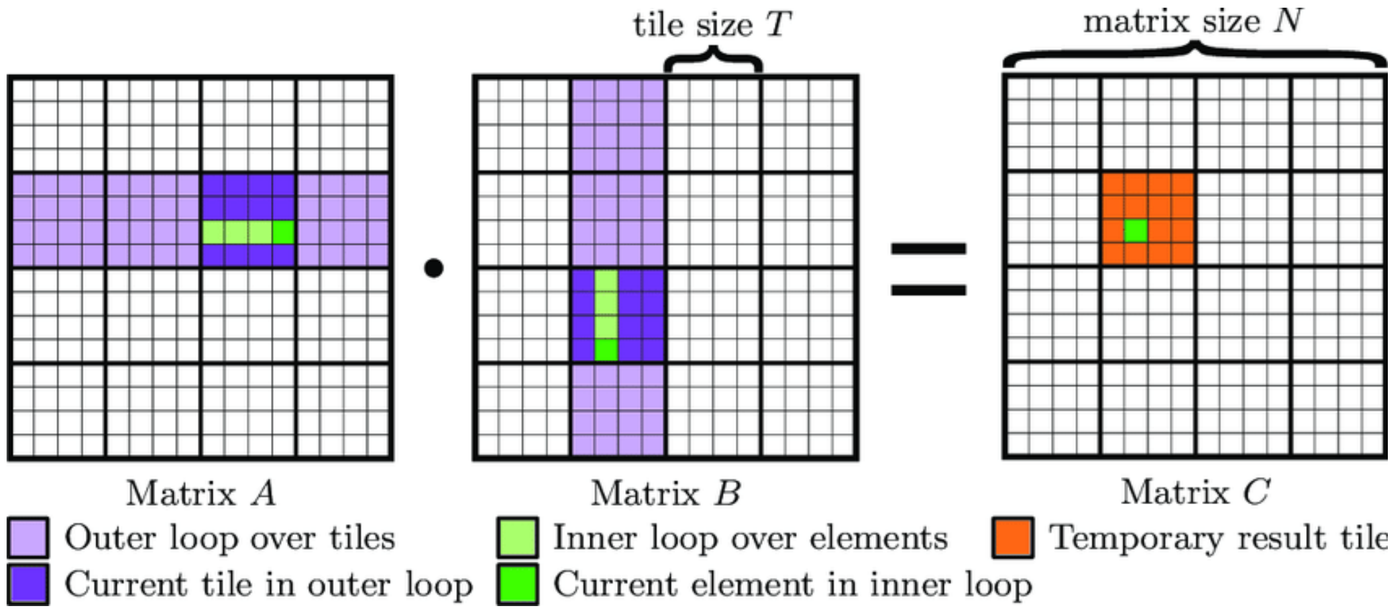




Tiling and Blocking

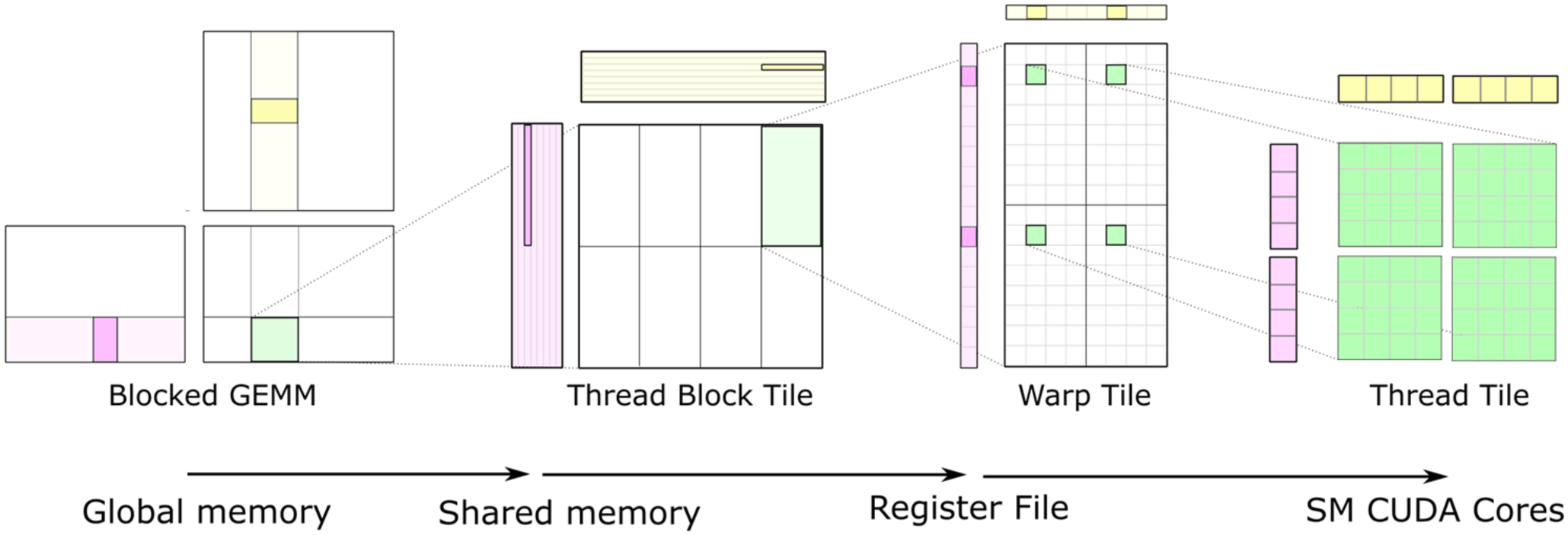
- Load sub-matrices (tiles) into shared memory for reuse.
- Each block computes a tile of the output matrix.
- Improves data locality and arithmetic intensity.





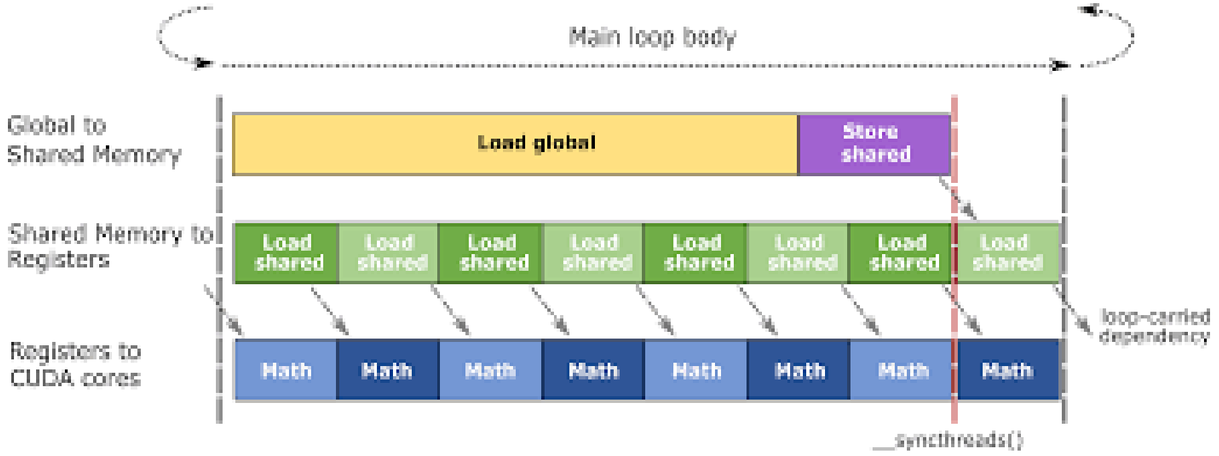
Shared Memory and Register Usage

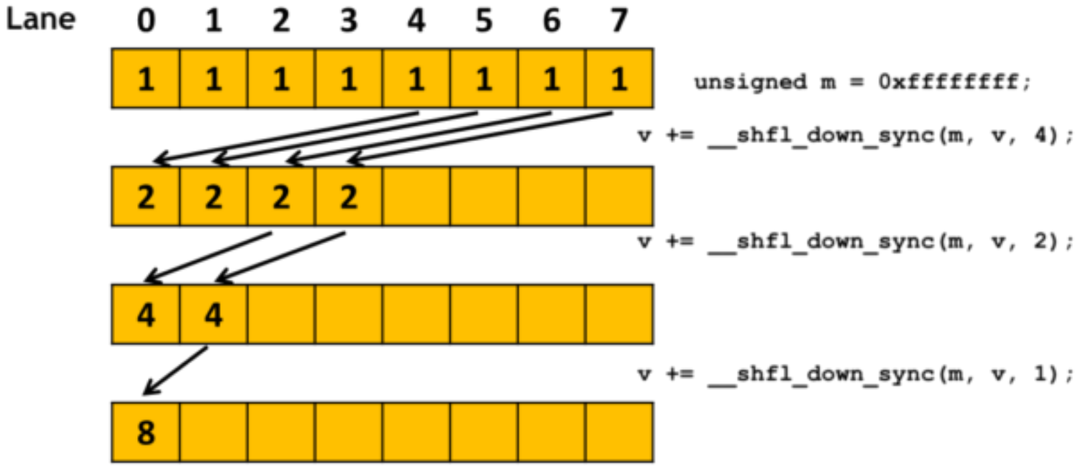
- Shared memory is faster but limited in size.
- Registers store partial results for each thread.
- Efficient usage minimizes global memory transactions.



Warp- and Thread-Level Optimization

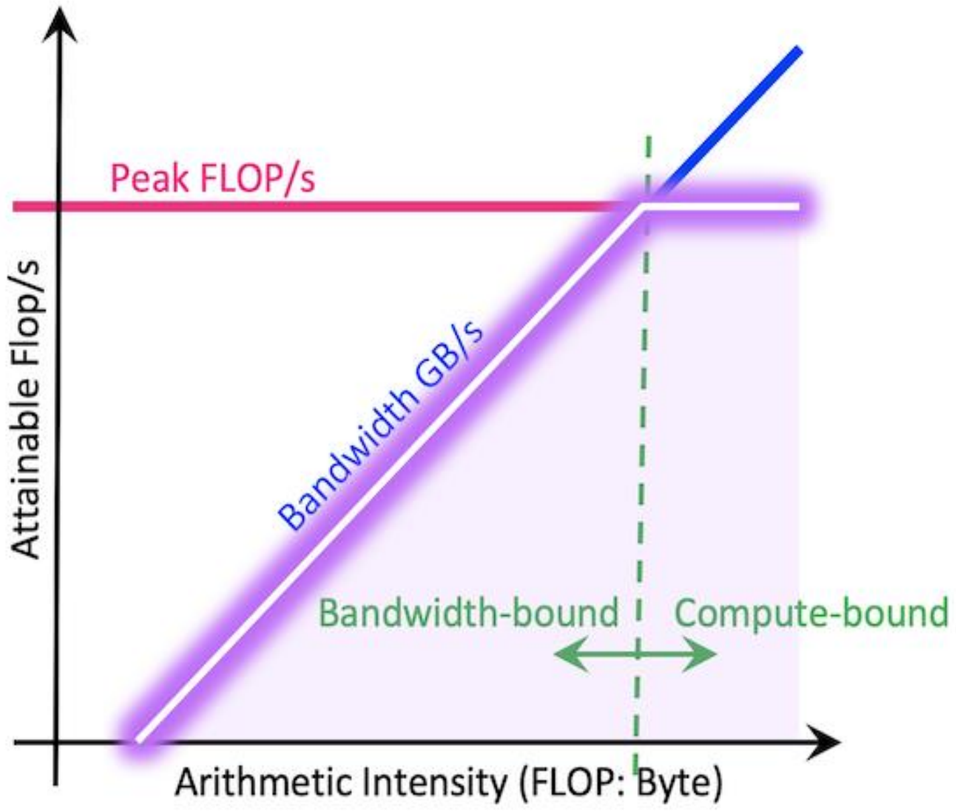
- Warp shuffle for intra-warp communication.
- Use vectorized operations (float4) for better throughput.
- Reduce instruction overhead per output element.





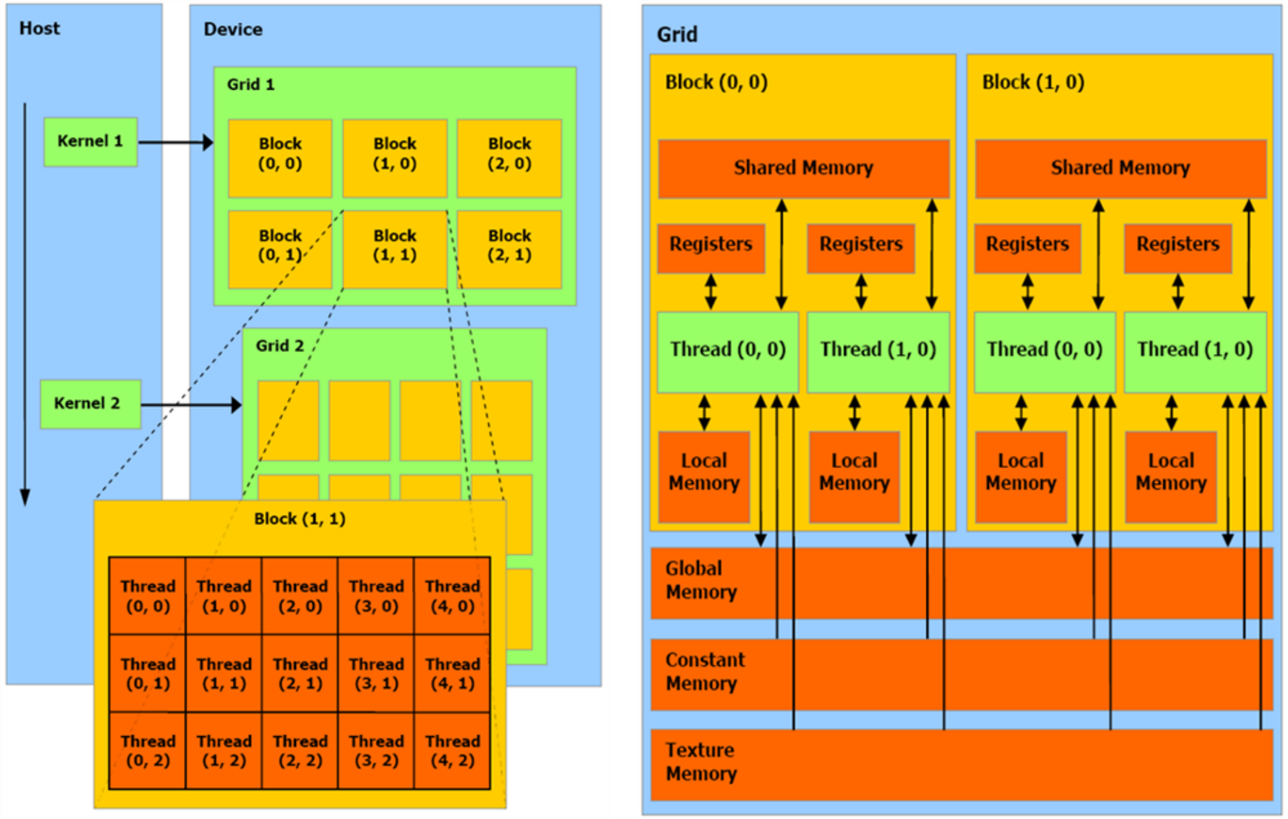
Roofline Model

- Shows performance limit by compute and memory bandwidth.
- Goal: shift from memory-bound to compute-bound region.
- Increase arithmetic intensity via reuse and blocking.



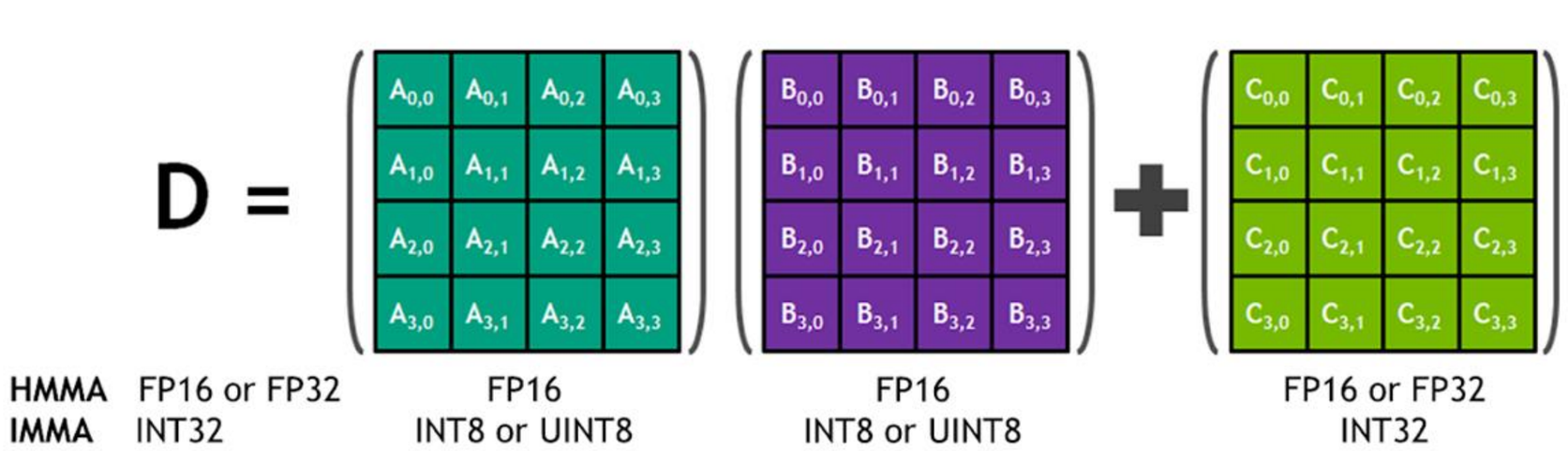
Launch Configuration

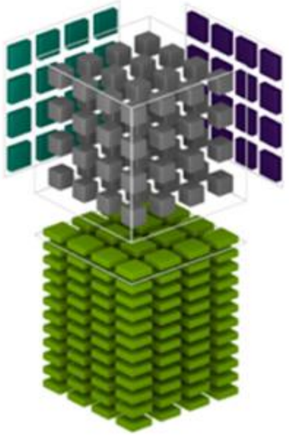
- Block and grid sizes affect occupancy and performance.
- Larger tiles: more reuse but may reduce occupancy.
- Tune parameters for target GPU architecture.



Tensor Cores and Mixed Precision

- Modern GPUs feature Tensor Cores for FP16/TF32 matrix ops.
- Provide massive speedup for GEMM in ML workloads.
- Trade-off between precision and performance.





RICE UNIVERSITY

yuke.wang@rice.edu