Flash Attention 3

A Preprint

Words for context:

- SMEM(shared memory) refers to small chip, programmer-managed cache that is directly addressable by all threads within a Cooperative Thread Array.
- WGMMA (Warp-Group Matrix Multiply-Accumulate): This is a specific instruction on NVIDIA's Hopper architecture (and newer GPUs) that accelerates matrix multiplication operations, particularly for Tensor Cores
- GEMM (General Matrix Multiply) refers to the operation of multiplying two matrices and adding the result to a third.
- s-stage circular SMEM buffer is a circular buffer which is used to efficiently synchronize warp groups

Warp groups are split in two: Producer and consumer referring to loading data from HBM and calculations respectively.

FlashAttention-3 accelerates attention computation on GPUs by leveraging hardware-specific features like asynchrony and low-precision formats. The core forward pass, as illustrated in Algorithm 1, employs a producer-consumer model within a Cooperative Thread Array (CTA).

- Data Organization: Input matrices Query (Q), Key (K), and Value (V) are partitioned into blocks (Q_i, K_j, V_j) to facilitate tiled processing.
- Warp Specialization: The CTA's warps are divided into two primary roles:
  1. Producer Warpgroup: This group is responsible for asynchronously loading blocks of Q_i, K_j, and V_j from High Bandwidth Memory (HBM) into faster Shared Memory (SMEM). It uses a circular SMEM buffer to manage stages and synchronize with the consumer.
  2. Consumer Warpgroup: This group performs the main attention computations. It initializes local accumulator O_i and scaling factors ℓ_i, m_i. In a loop over key/value blocks (K_j, V_j):
     - It waits for Q_i and K_j to be loaded into SMEM.
     - It computes the attention scores $S_i^{(j)} = Q_i(K_j)^T$ using a Shared Memory-sourced General Matrix Multiply (SS-GEMM).
     - It updates the numerical stability factors m_i and computes $\exp(S_i^{(j)} - m_i)$ and ℓ_i for the softmax operation.
     - It waits for V_j to be loaded into SMEM.
     - It computes the partial output O_i by scaling and adding $\exp(S_i^{(j)} - m_i)V_j$ using a Register-sourced General Matrix Multiply (RS-GEMM).
     - It signals the producer warpgroup to release the current SMEM buffer stage.
- Finalization: After processing all blocks, the consumer warpgroup performs final scaling of O_i and computes the logsumexp L_i, then writes the completed block O_i and L_i back to HBM.

This approach minimizes global memory access by fusing operations and utilizing SMEM, and hides latency through asynchronous data movement and computation.

Also, even within one group it is possible to ovelap p some instructions in the softmax with some instructions in the GEMMs.

FlashAttention-3's FP8 optimization addresses two key challenges:

- Layout Conformance (Efficiency): Standard FP8 WGMMA requires V V tiles in SMEM to be contiguous in the sequence length dimension, conflicting with typical HBM layouts. FlashAttention-3 overcomes this by performing an in-kernel transpose of V V tiles after loading them into SMEM, leveraging LDSM/STSM instructions. This also includes permuting the FP32 accumulator layout to match FP8 WGMMA requirements, ensuring compatibility.
- Accuracy (Numerical Error): To mitigate high numerical error from FP8 (e4m3) and outliers, two techniques are used:
    1. Block Quantization: Instead of per-tensor scaling, each block of Q, K, and V is quantized with its own scalar. This is fused with pre-attention operations and helps manage outliers more effectively.
    2. Incoherent Processing: Q and K are multiplied by a random orthogonal matrix M M before FP8 quantization. This "spreads out"outliers without changing the attention output, thus reducing quantization error.