Optimal Symmetric Power in CUDA

Xiaowen Zhang

1 Introduction

In this document, we explore the optimal layout for implementing symmetric power operations in the context of the chunk state kernel. We aim to design an efficient memory layout that facilitates fast computation and minimizes data movement.

2 Problem Statement

The chunk state kernel involves complex operations on matrices, including element-wise multiplications and matrix multiplications. Our goal is to find a memory layout that optimizes these operations, particularly for the symmetric power calculations.

3 Current Approach

The current implementation, as outlined in the enumerated steps, involves reading elements from a matrix K, computing a product, and storing the result in a new matrix $\phi(K)$. This approach may have room for improvement in terms of memory access patterns and computational efficiency.

4 Proposed Optimizations

We propose the following optimizations to enhance the performance of the chunk state kernel:

4.1 Tiled Memory Access

Implement a tiled approach for accessing the K matrix to improve cache utilization.

4.2 Vectorized Computations

Utilize SIMD instructions to parallelize the element-wise multiplications within each thread.

4.3 Shared Memory Usage

Leverage shared memory to store frequently accessed data, reducing global memory accesses.

5 Performance Analysis

We will analyze the performance of the proposed optimizations compared to the current implementation, focusing on metrics such as execution time, memory bandwidth utilization, and computational throughput.

6 Conclusion

By carefully considering the memory layout and computational patterns, we aim to significantly improve the efficiency of the symmetric power operations in the chunk state kernel.

The overall operation of the chunk state kernel can be decomposed into the following:

1. For each thread, do the following for all
   (a) read P elements $\{k_1, k_2, \ldots, k_P\}$ from a matrix K of size **BlockK** $\times$ **HeadDim**
   (b) read an additional coefficient c
   (c) compute $\phi(K) = c \cdot \prod_p k_p$
   (d) store the result somewhere, it should logically represent a matrix $\phi(K)$ of shape **BlockK** $\times$ **BlockD**
2. Carry out matrix multiplication $\phi(K)^T V$