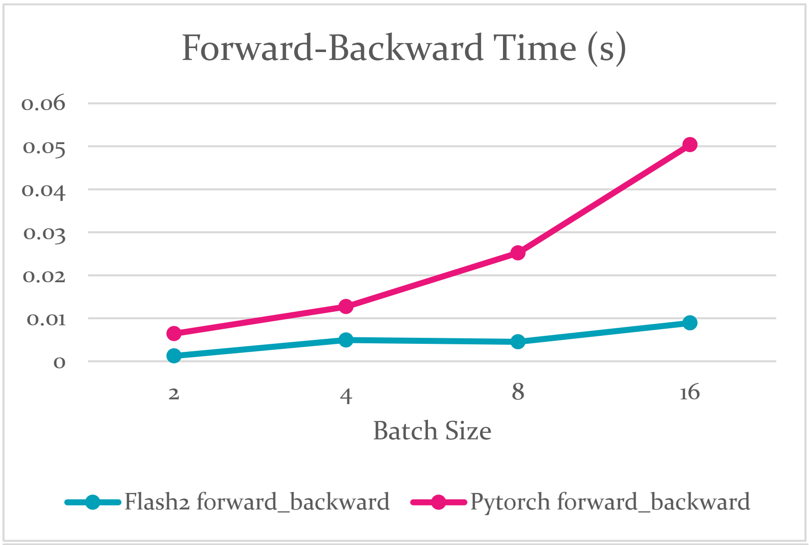
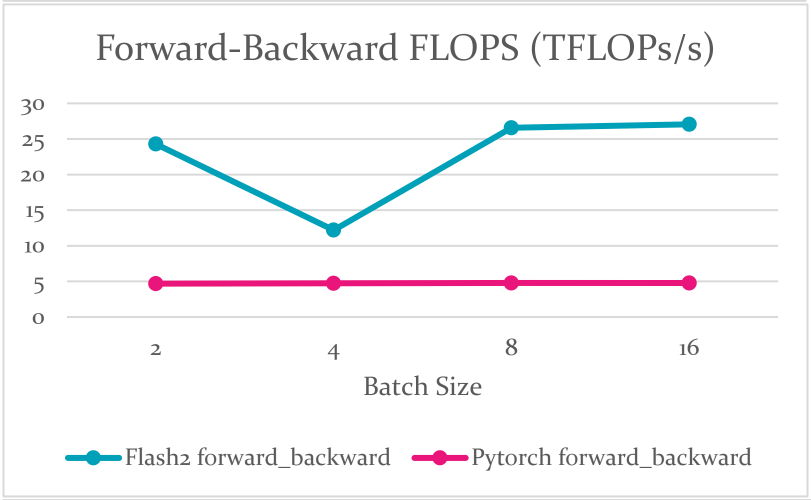
Flash Attention Benchmark

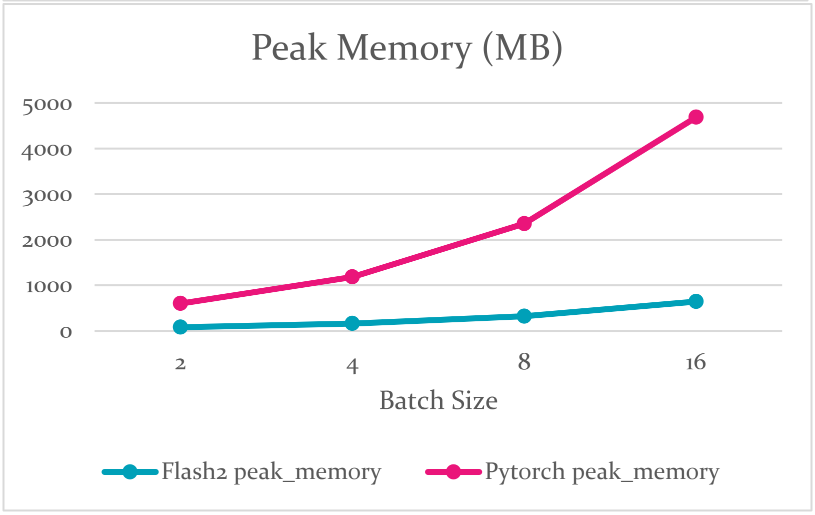
PARALLEL PROGRAMMING LAB5

劉得崙 | pp24 | 2024.11.21

FLASH2 VS PYTORCH







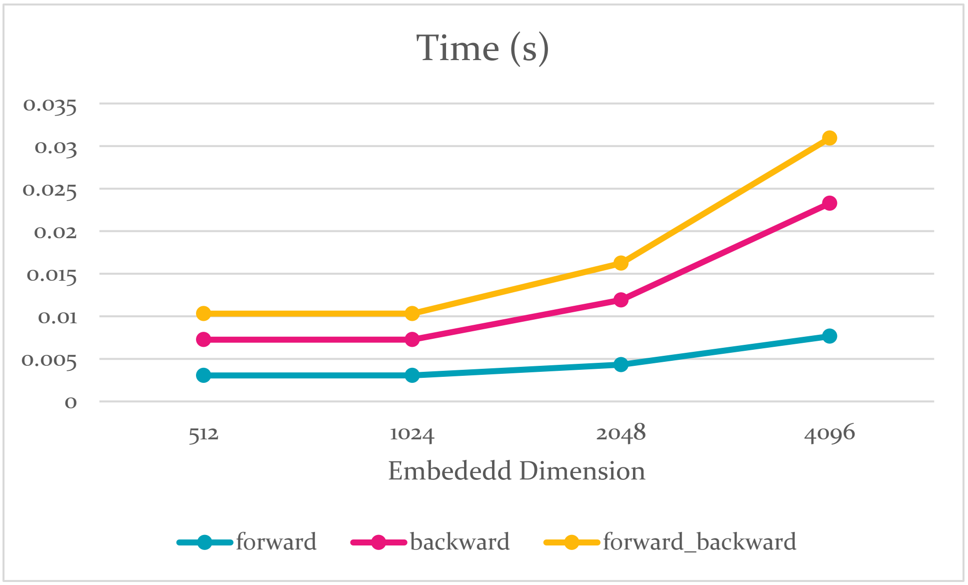
As figure shown above, using flash attention can reduce overall computing time, increase FLOPS, and reduce peak memory usage.

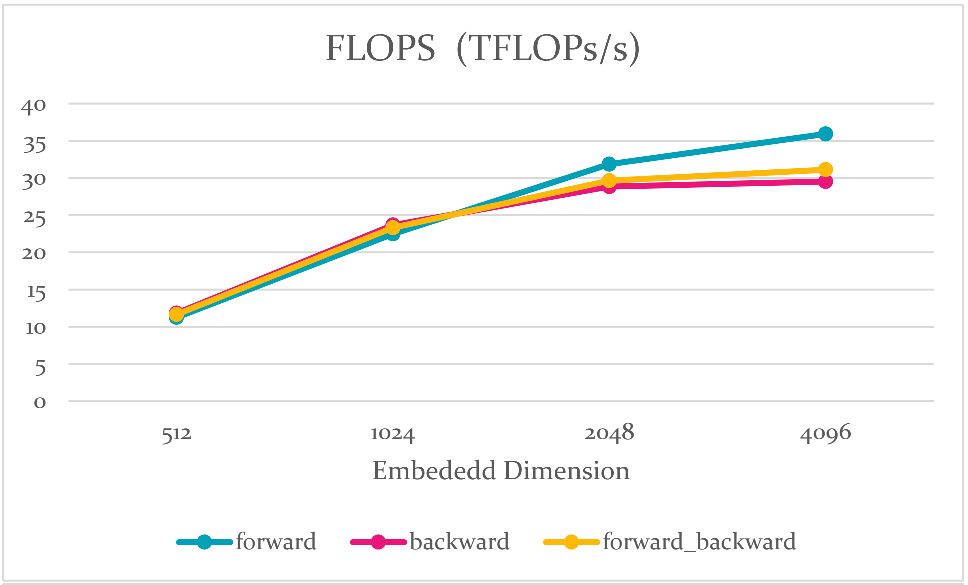
1. Reduce computing time:
   In flash attention implementation, it uses shared memory (PBSM) within each SM unit for calculation, reducing the time-consuming process of accessing global memory (HBM), which significantly improves the overall performance.

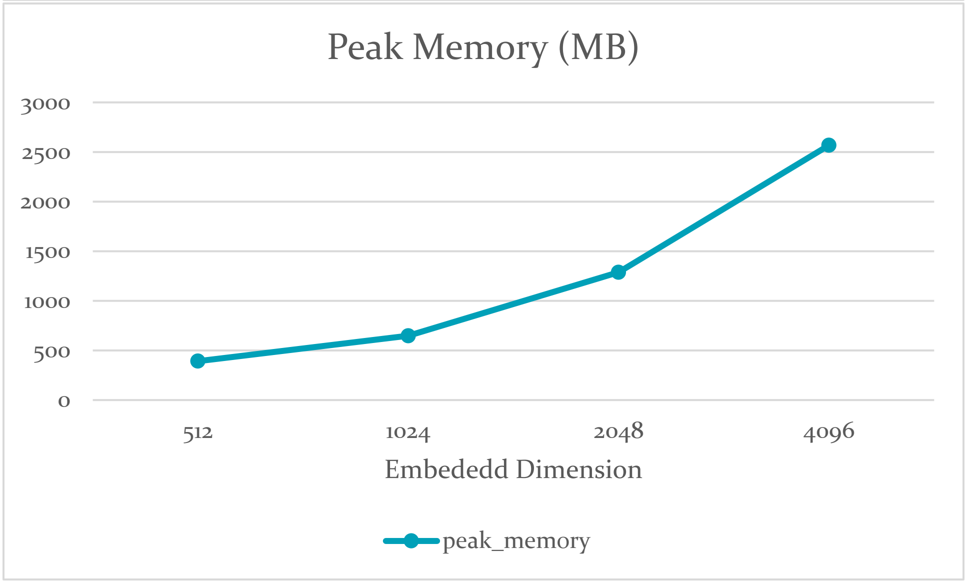
2. Increase FLOPS:
   While global memory access is not a bottleneck anymore, the throughput increases accordingly, making FLOPS also increase.

3. Reduce peak memory usage:
   While traditional attention needs to store entire softmax output, which is O(N*M) storage, flash attention only needs to the final result, the intermediate value can be stored in PBSM, making the peak memory usage less.

MODIFY EMB_DIMS (FLASH2)

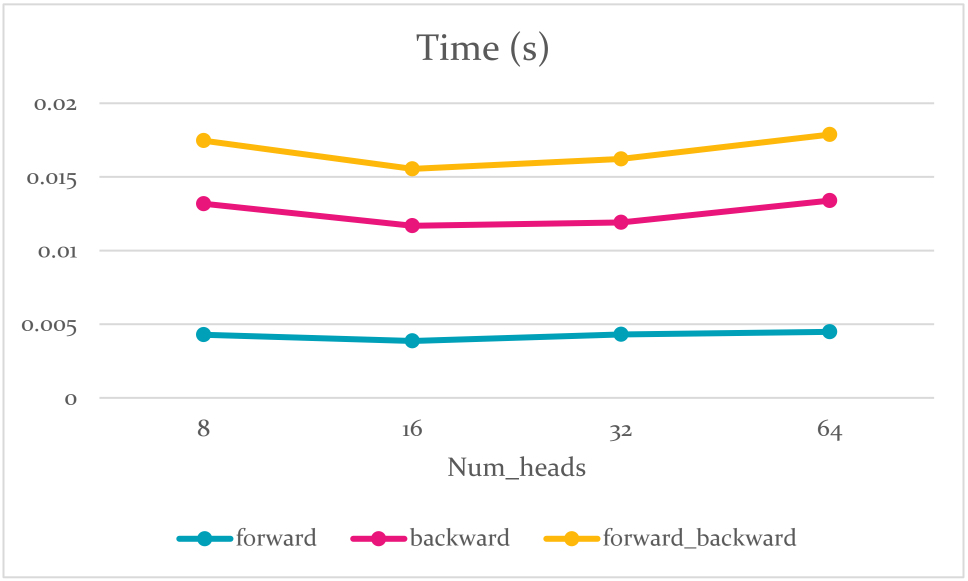


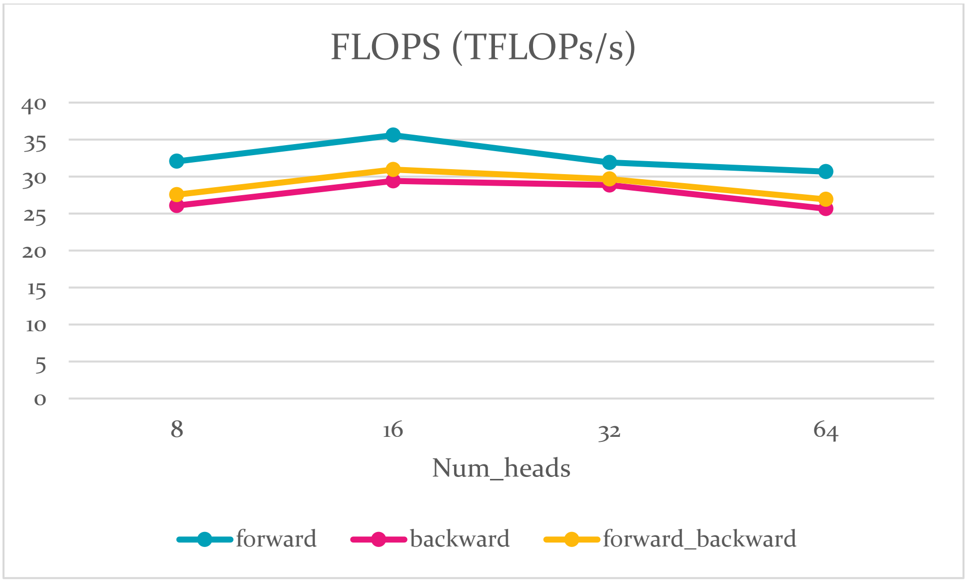


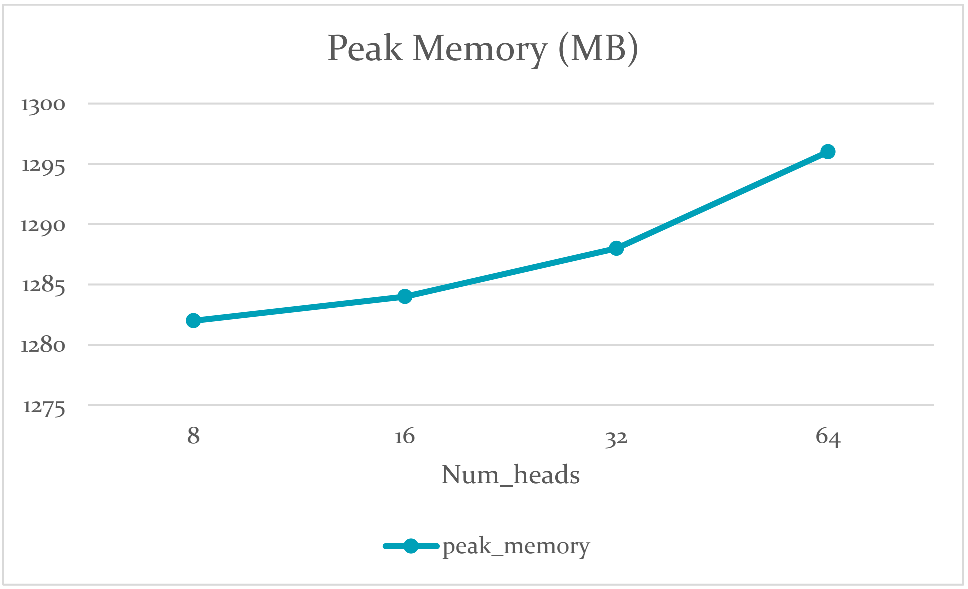


Embedded dimension directly accounts for the number of parameters/weights of the model. As embedded dimension becomes larger, both calculation time and peak memory usage goes up, as well as FLOPS.

MODIFY NUM_HEADS (FLASH2)

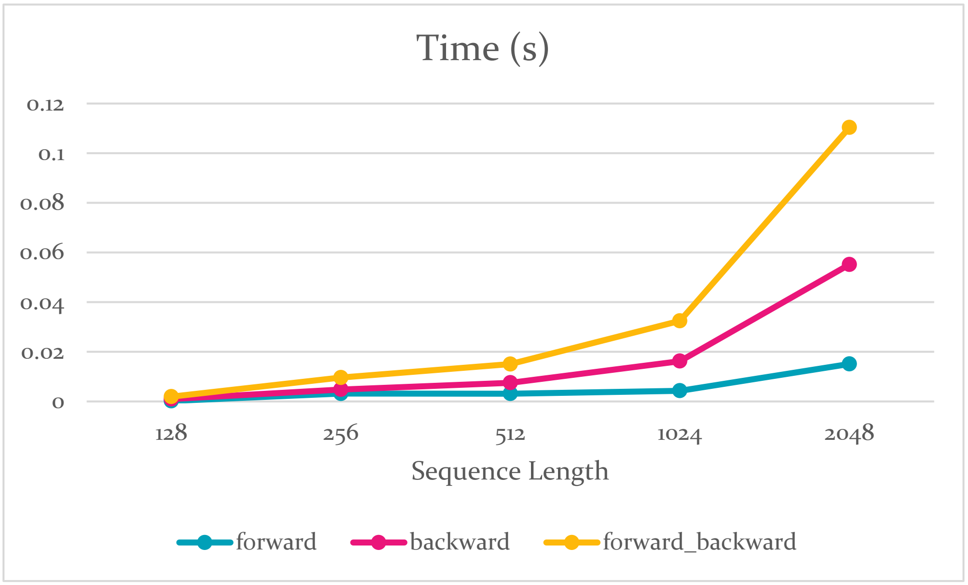


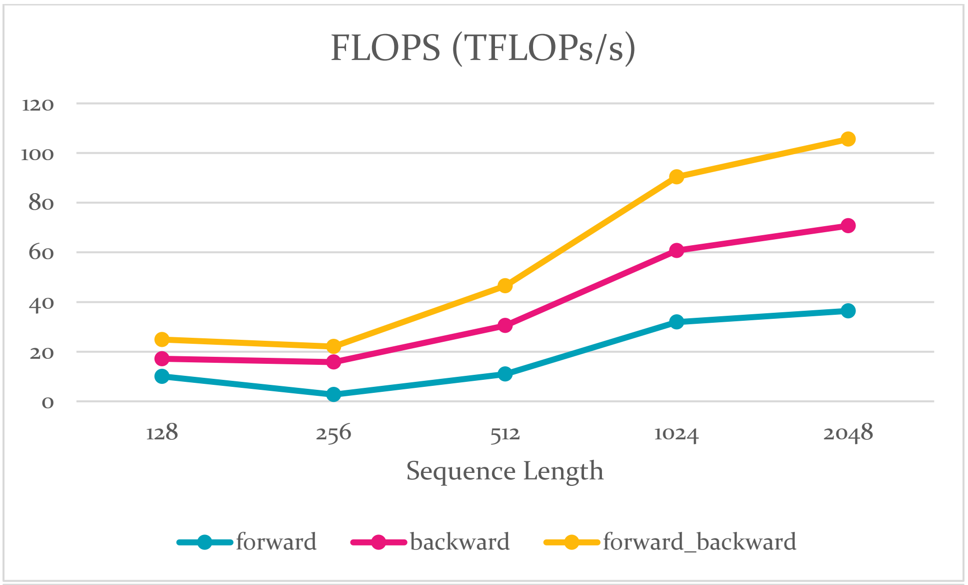


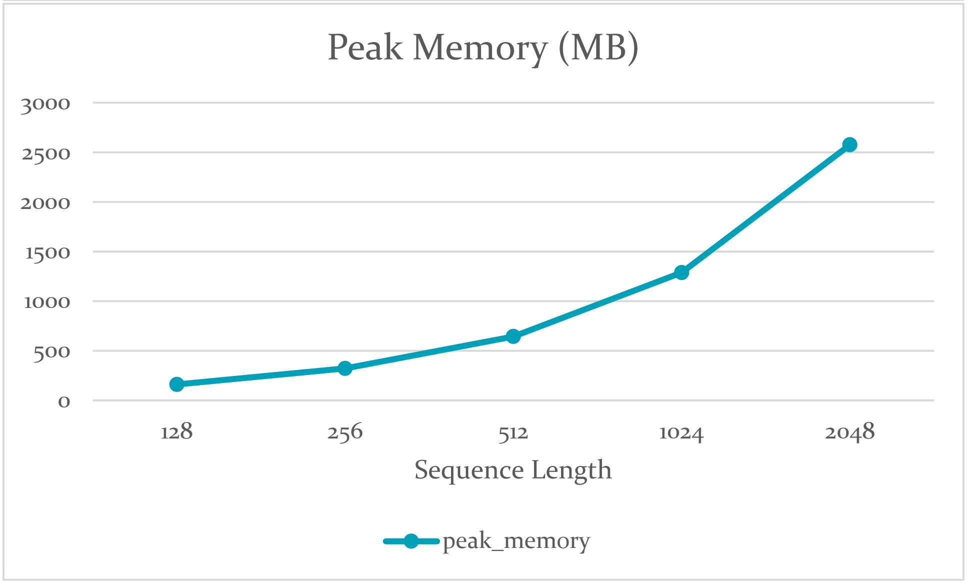


Num_heads accounts for the number of parallel attention computations in each pass. In different num_heads value, since the total matrix size stays the same, even if the memory needs to be partitioned differently for each head, the total FLOPS(calculation time) remains the same. However, we need separate (duplicate) memory blocks for each heads' computations, resulting in higher peak memory usage.

MODIFY SEQ_LEN (FLASH2)







The sequence directly accounts for dimension of matrix Q (N * d), the larger the seq_len, the more parameters/weights need to be calculated, thus results in slower time, larger peak memory, and FLOPS.