week12实验记录

zxp

May 18, 2024

1 environment

cpu:Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz (使用时申请了56个核心，不知道为什么这周使用独占指令的时候没法运行作业)

gpu:rtx3090(使用时申请了一块)

System:CentOS7

Compiler:9.5

2 code

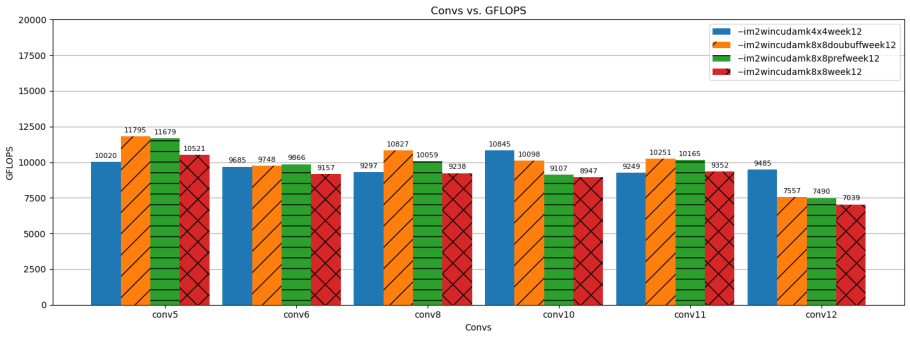

Figure 1: GPU

3 Experiment

这周继续按矩阵乘法的优化优化GPU上的代码，上周做到了1x4微内核，这周试了4x4微内核（蓝色柱子），8x8微内核（红色柱子），在8x8微内核基础上做预取（绿色柱子）和双重缓冲区（黄色柱子）。但只在6层上实现了，其他层output的通道数不能被128整除（96和64）所以需要额外处理，这周没写。
4x4和8x8内核主要是增加每个线程的工作量和增大运算/搬运的比例，增加共享内存的复用，但增加工作量同时会增加每个线程计算下标的数量和计算对应的数据搬运带来的额外开销。不同微内核每次计算的长度不同所以填充的倍数可以不同（4x4每次搬运完计算16长度，这时input和filter的窗口大小填充至16的倍数，8x8每次搬运完计算8长度，这时input和filter的窗口大小填充至16的倍数）。预取通过提前将数据放入寄存器让计算和数据搬运同时进行。双重缓冲区给input和filter都申请两块共享内存，将读写分开，减少进程间同步的次数。

3.1 Analysis

4x4微内核效果很好，比4x1和1x4微内核快一倍多，但8x8微内核就没这样的效果了甚至某些层负优化，预取也让性能增加了部分虽然不如微内核，双重缓冲区也是让性能增加了部分（除了conv6）。总体效果还是没有卢帅师兄论文中的效果，差1tflops左右。