week11实验记录

zxp

May 11, 2024

1 environment

cpu:Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz (使用时申请了56个核心并且使用独占指令)

gpu:rtx3090(使用时申请了一块)

System:CentOS7

Compiler:9.5

2 code

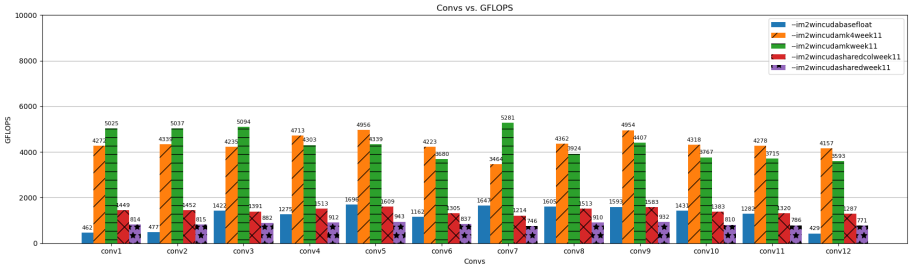

Figure 1: GPU

3 Experiment

这周按照卢帅师兄说的隐式矩阵乘法写GPU上的代码，把输入张量看成矩阵A过滤器看成矩阵B，和把这些当成矩阵加载到共享内存中去。因为要当成矩阵，所以直接使用的bhwc布局的im2win这样一个窗口就是矩阵的一行不需要跨通道去找其他元素方便一点。然后因为是当成矩阵一小块一小块加载到共享内存，所以对张量进行了填充，填充至32的倍数，这样不需要考虑分块后剩下的元素，填充方式和cpu上一样，对过滤器是填充每个窗口，对input张量只填充最后一个窗口。
这样做完之后，就可以把矩阵乘法的优化套上去，这周试了共享内存、float4（基本没效果，并且因为数据布局和矩阵乘法不一样也基本套不进去）和微内核（1x4，4x1）。共享内存减少数据加载次数，增加复用，和cpu不一样的是共享内存不是每个核心连续范围更合适，而是列优先更合适，貌似和GPU共享内存的特性有关，共享内存会被很多线程同时访问，相邻线程访问共享内存中靠近的部分会更快。微内核可以增加每个线程的工作量，卢帅师兄给的那篇教程做最多的事情也是逐步加大每个线程的工作量，最后还剩下的优化还有逐步调整微内核、unroll（这个在GPU上加个#pragma unroll就可以）和预取。

3.1 Analysis

这周重要正优化了。之前共享内存使用是完全不对的，一个是放的东西不对，一个是顺序不对，可以看到列优先比行优先快了一倍，这是共享内存的特性。增加每个线程的工作量也是关键，glop翻了好几倍，1x4或者4x1的内核应该不是极限，接下来试试4x4，8x8（有些conv没法用8x8，通道数量不够大）。

4 Experiment2

nsight compute需要sudo指令，没法在服务器上跑，nsight compute是内核级的，GPU总体情况看nsight。获取显存使用量nsight应该可以，不过还没找到怎么看。