week-10

JX-Ma

2024/5/4

1 本周工作

1. 结合前几周的实验，把有利于 im2win 优化的结合到一起。2. 尝试一下改变不同输出张量宽度的分块大小，看能不能继续优化。

2 实验部分

2.1 实验环境

- 系统: CentOS7
- gcc version : 13.2.0
- 优化选项: -O3 -fopenmp -avx2 -fmadd
- cpu: Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz

2.2 实验一

- 通道展开方面：将 conv1 conv2 conv3 conv4 conv7 中输入张量和卷积核张量不按照通道展开，其余 benchmark 全部按照通道展开。
- 内存对齐: 使用了 posix_memalign 对储存数据的指针动态分配对齐内存。
- 融合维度: 融合了输出张量的 batch 和 height 维度，循环顺序为 bh-c-w。

实验结果发现在 conv1,conv2,conv3,conv4 上面性能反而还不如只融合维度这一个优化，结合上周实验，不展开 conv1,conv2,conv3,conv4 是正优化，但是结合了融合维度这一优化方法后就变成了负优化，然后 conv5,6,8,9,10，11 上面是正优化。
后面尝试将 conv1conv2conv3conv4conv7 上面维度全部展开，发现性能有所上升。

2.3 实验二

这周尝试了改变对输出张量宽度分块的大小，其中 conv1 是 11 变为 5，conv2 上面是 7 变为 8，conv5-6 由 10 变为 5，conv8 由 11 变为 10，conv9-11 都变为 6. 结果表明只有 conv2 这一层有着较为明显的提升。

2.4 实验三

结合上两种实验，最后得出最优化的版本的 im2win, 也就是通道展开方面全部展开，conv2 上面分块大小变为 8，内存对齐采用 posix_memalign 对储存数据的指针动态分配对齐内存。维度融合则采用了 bh-c-w.

3 实验结果

实验一的实验结果对应下标 f1
实验二的结构对应 f3
实验三的实验结果对应 f2.
这周实验都是在上周实验只融合 bh-c-w 为基础进行的。

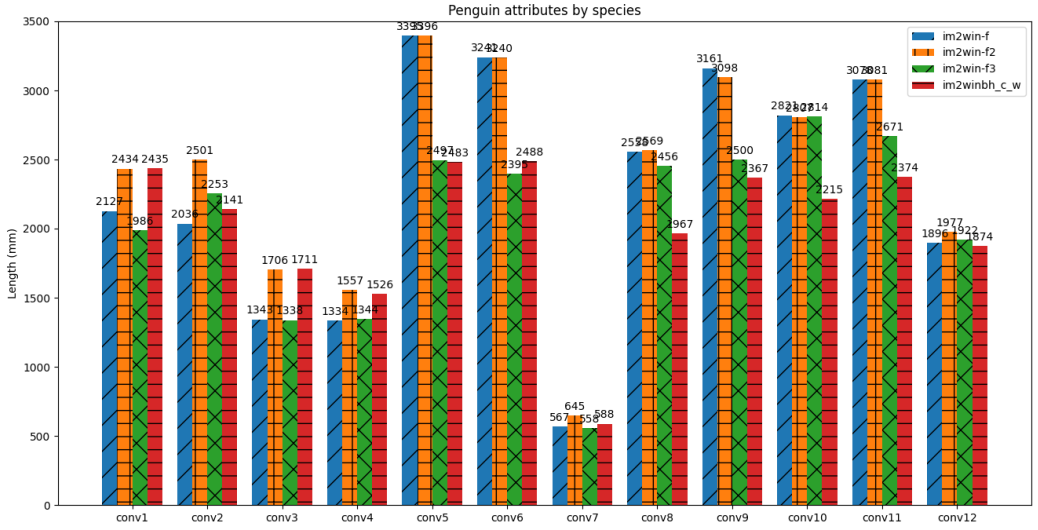

图 1: gflops