Dataflow 问题

February 5, 2018

1 问题描述

以一个简单的 1D 卷积为例，卷积核为 2x1，输入为 3x1，输出为 2x1，其计算伪代码如下所示：

```
Loopi: for(i=0; i<2; i++){
         Loopj: for(j=0; j<2; j++){
                  O[i] += K[j] * I[j+i];
                }
       }
```

2 Graph 变换

将上述计算完全展开可显示其可并行性和数据依赖。如 Figure 1 所示，K1、K2 和 I2 存在数据复用的机会，计算完全并行时需要 4 个乘法器和 2 个加法器。实际情况中，由于计算资源和访存带宽的限制并不能将计算完全并行化，同时考虑到能耗的限制，需要充分利用计算中数据复用的机会，减少从外部存储读取数据。假设 PE 单元数目为 2，下面给出 NO Local Reuse 和 Weight Stationary 时的分析。

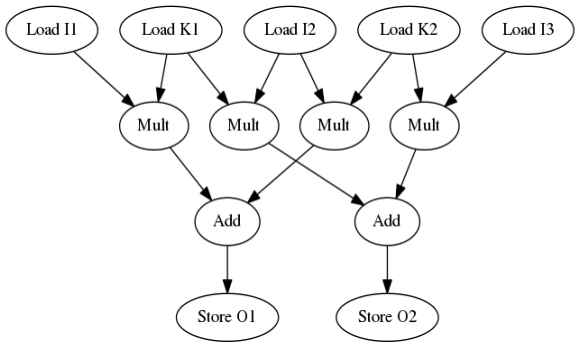
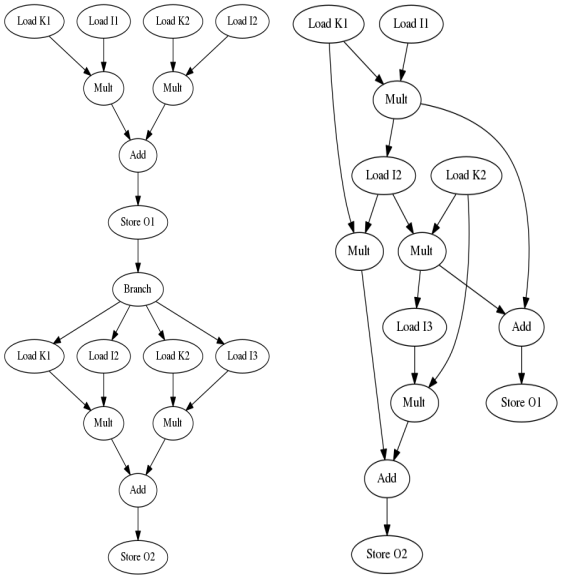


Figure 1: Origin Graph

2.1 No Local Reuse

Figure 2(a) 是 No Local Reuse 的 graph 表示图，从图中可以看出，由于硬件资源限制，内层循环被展开并行执行，外层循环顺序执行。由于没有 local register(Local Reuse)，外层循环的计算每次都需要重新读入权重和输入数据。



(a) No Local Reuse (b) Weight Stationary

Figure 2: Transformed Graph

2.2 Weight Stationary

Figure 2(b) 是 Weight Stationary 的 graph 表示图。从图中可以看出，权重被 load 一次后即存在 PE 中被复用，I1、I2、I3 依次读入并广播至所有 PE 分别与权重数据进行计算。按照 Aladdin 的策略 Schedule 该 graph，依次执行的计算是 (I1×K1)(I2xK1, I2xK2, o1=I1xk1+I2xK2)(I3xK2, 02=I2xK1+I3xK2)。即 Oi 的部分和 I1xK1 在第一个 cycle 的 PE1 中计算，I2xK2 在下一个 cycle 的 PE2 中计算，同时在第二个 cycle 将 PE1 中的输出传入 PE2 做累加，如此流水进行。

相比较 NLR，WS 的 latency 增大，即每个 output 需要两个 cycle 计算，但由于存在 local 的数据复用，减少了从 buffer 读取数据的次数。