TVM的前后端优化

TVM的前后端优化
AI编译器开发指南

Agenda

- TVM的系统结构和编译流程
- TVM的前端优化
- TVM的后端优化

TVM的前后端优化
AI编译器开发指南

TVM的系统结构和编译流程

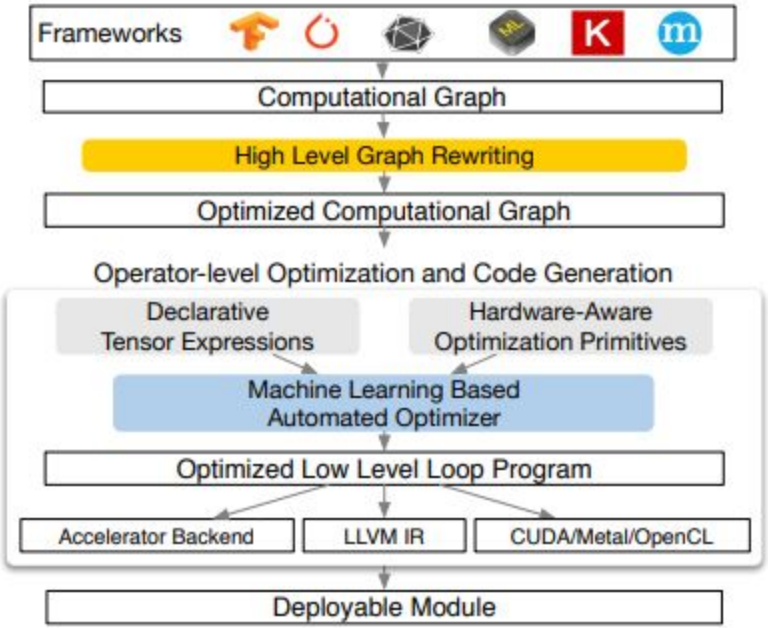
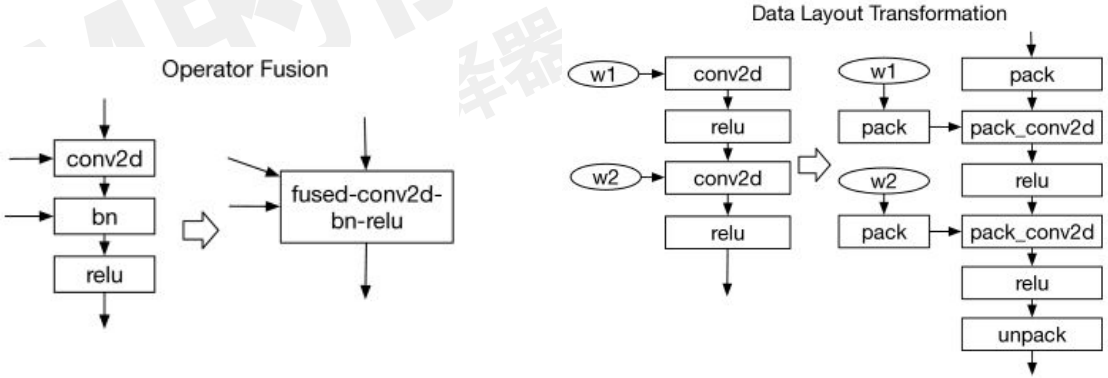


图 1-6 TVM 系统结构图[1]

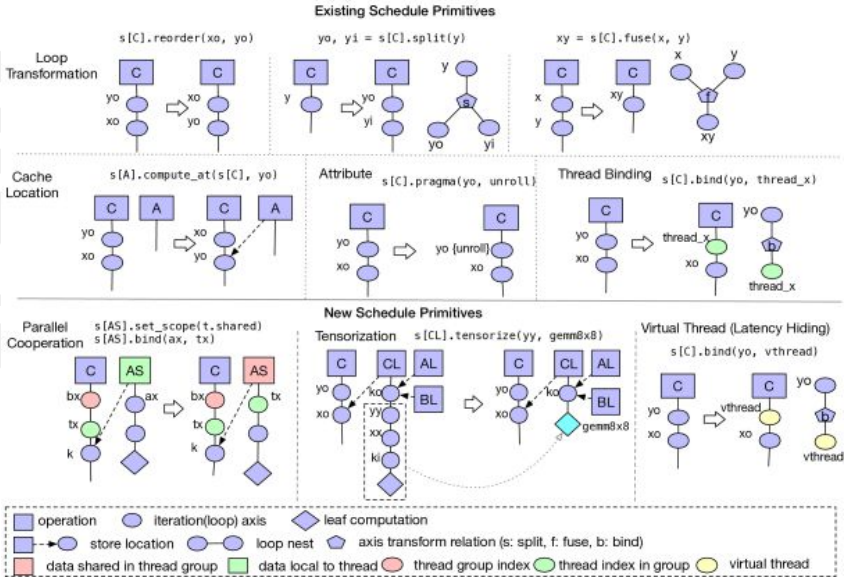
TVM的前端优化 - 计算图优化

- 计算图是一种高阶IR, 图中的节点表示张量运算或对程序输入的运算, 节点之间的边表示运算之间的数据依存关系。
- Relay IR是TVM 中的一种高阶图级 IR 和语言。
- TVM可以对计算图做不同范围和层次的优化。



TVM的前端优化 - 张量表达式

- 计算图不能描述所有优化约束
- 张量表达式在Halide调度原语的基础上，引入了新的调度原语，用于优化GPU和专用加速器性能。
  - split
  - tile
  - fuse
  - reorder
  - ...



https://tvm.apache.org/docs/how_to/work_with_schedules/schedule_primitives.html

TVM的后端优化

- TVM后端优化方法主要包括:硬件 intrinsic 函数、内存延迟隐藏、循环优化和并行化等。
- 硬件intrinsic函数是一种将低阶IR中的特定操作模式映射为优化内核的机制。
- 虚拟线程是通过创建最内层循环来模拟线程的并发执行。

```
def intrin_gemv(m, l):
    a = te.placeholder((l,), name="a")
    b = te.placeholder((m, l), name="b")
    k = te.reduce_axis((0, l), name="k")
    c = te.compute((m,), lambda i: te.sum(a[k] * b[i, k], axis=k), name="c")
    Ab = tvm.tir.decl_buffer(a.shape, a.dtype, name="A", offset_factor=1, strides=[1])
    Bb = tvm.tir.decl_buffer(b.shape, b.dtype, name="B", offset_factor=1,
                             strides=[te.var("s1"), 1])
    Cb = tvm.tir.decl_buffer(c.shape, c.dtype, name="C", offset_factor=1, strides=[1])

    def intrin_func(ins, outs):
        ib = tvm.tir.ir_builder.create()
        aa, bb = ins
        cc = outs[0]
        ib.emit(tvm.tir.call_extern("int32", "gemv_update", cc.access_ptr("w"),
                        aa.access_ptr("r"), bb.access_ptr("r"), m, l, bb.strides[0]))
        return ib.get()
    return ste.decl_tenor_intrin(c.op, intrin_func, binds={a: Ab, b: Bb, c: Cb})
```