# Hybrid Script: A Text Format for Halide IR & A Python-TVM Hybrid Frontend

Jian Weng
UCLA, PolyArch

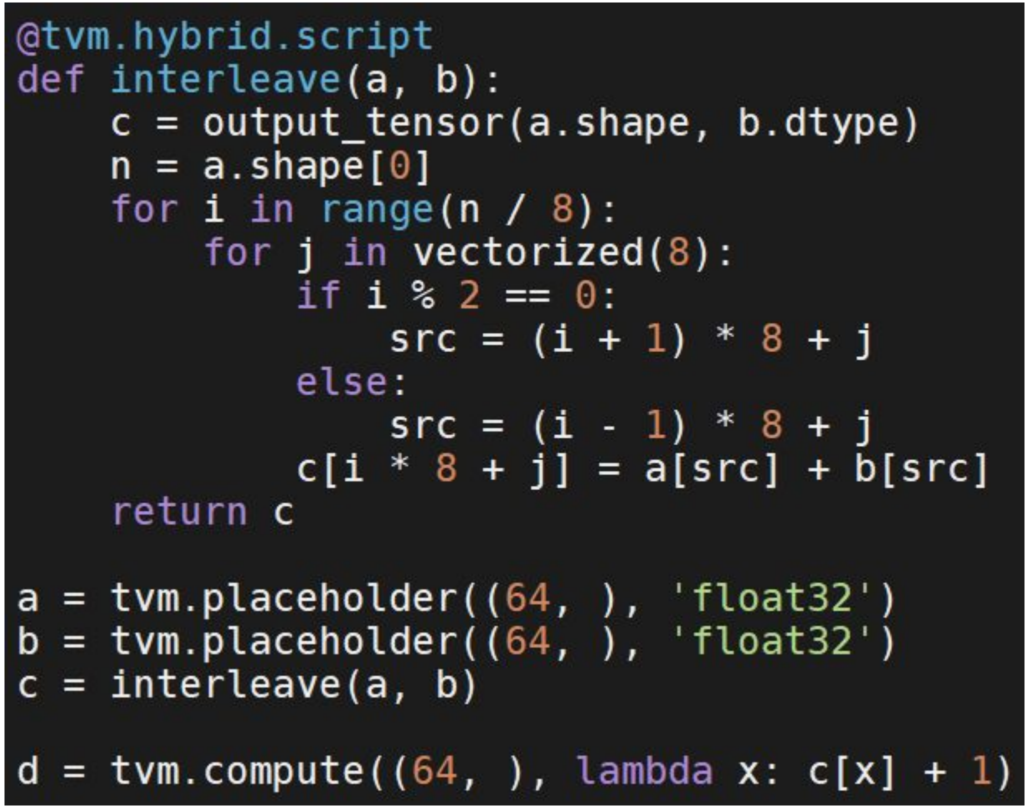
Dec. 12th, 2018

## Motivation

- *tvm.compute* works only great for dense tensor operation kernels
    - Sparse/Irregular kernels or cleanup operators
- The downside of current *IRBuilder*+*ExternOpNode*
    - *IRBuilder* is a developer-oriented module
        - Originally targets to writing test cases efficiently
        - Requires to know low-level technique details
        - No light-weighted way to do sanity check
        - Learning a large set of APIs with many arguments
- Using a subset of Python to program will be highly desirable

## HybridOpNode

- After annotating a function with *hybrid* decorator
  - A subset of Python with slight extension is supported
- Pass a *numpy* tensor to do in-Python emulation
- Pass TVM tensors to compile the script and get a *operator* node



```
@tvm.hybrid.script
def interleave(a, b):
    c = output_tensor(a.shape, b.dtype)
    n = a.shape[0]
    for i in range(n / 8):
        for j in vectorized(8):
            if i % 2 == 0:
                src = (i + 1) * 8 + j
            else:
                src = (i - 1) * 8 + j
            c[i * 8 + j] = a[src] + b[src]
    return c

a = tvm.placeholder((64, ), 'float32')
b = tvm.placeholder((64, ), 'float32')
c = interleave(a, b)

d = tvm.compute((64, ), lambda x: c[x] + 1)
```