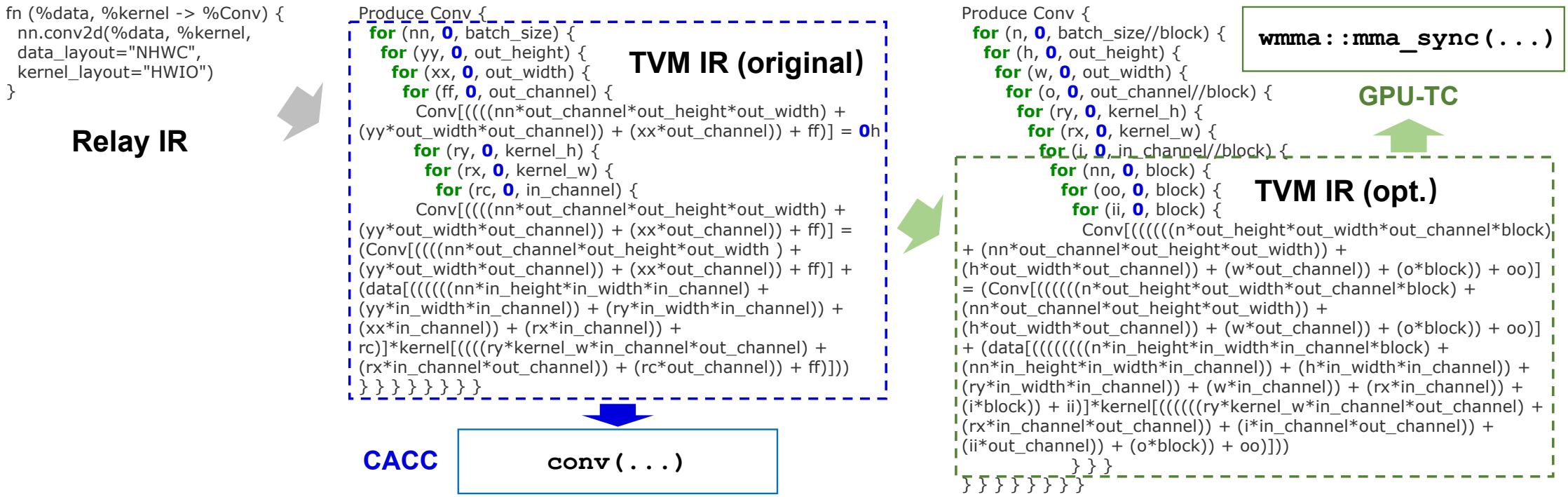
fn (%data, %kernel -> %Conv) {
nn.conv2d(%data, %kernel,
data_layout="NHWC",
kernel_layout="HWIO")
}
Relay IR
Produce Conv {
for (nn, 0, batch_size) {
for (yy, 0, out_height) {
for (xx, 0, out_width) {
for (ff, 0, out_channel) {
Conv[((((nn*out_channel*out_height*out_width) +
(yy*out_width*out_channel)) + (xx*out_channel)) + ff)] = 0h
for (ry, 0, kernel_h) {
for (rx, 0, kernel_w) {
for (rc, 0, in_channel) {
Conv[((((nn*out_channel*out_height*out_width) +
(yy*out_width*out_channel)) + (xx*out_channel)) + ff)] =
(Conv[((((nn*out_channel*out_height*out_width ) +
(yy*out_width*out_channel)) + (xx*out_channel)) + ff)] +
(data[(((((nn*in_height*in_width*in_channel) +
(yy*in_width*in_channel)) + (ry*in_width*in_channel)) +
(xx*in_channel)) + (rx*in_channel)) +
rc)]*kernel[((((ry*kernel_w*in_channel*out_channel) +
(rx*in_channel*out_channel)) + (rc*out_channel)) + ff)]))
} } } } } } } }
TVM IR (original)
CACC
conv(...)
Produce Conv {
for (n, 0, batch_size//block) {
for (h, 0, out_height) {
for (w, 0, out_width) {
for (o, 0, out_channel//block) {
for (ry, 0, kernel_h) {
for (rx, 0, kernel_w) {
for (i, 0, in_channel//block) {
for (nn, 0, block) {
for (oo, 0, block) {
for (ii, 0, block) {
Conv[((((((n*out_height*out_width*out_channel*block)
+ (nn*out_channel*out_height*out_width)) +
(h*out_width*out_channel)) + (w*out_channel)) + (o*block)) + oo)]
= (Conv[((((((n*out_height*out_width*out_channel*block) +
(nn*out_channel*out_height*out_width)) +
(h*out_width*out_channel)) + (w*out_channel)) + (o*block)) + oo)]
+ (data[(((((((n*in_height*in_width*in_channel*block) +
(nn*in_height*in_width*in_channel)) + (h*in_width*in_channel)) +
(ry*in_width*in_channel)) + (w*in_channel)) + (rx*in_channel)) +
(i*block)) + ii)]*kernel[((((((ry*kernel_w*in_channel*out_channel) +
(rx*in_channel*out_channel)) + (i*in_channel*out_channel)) +
(ii*out_channel)) + (o*block)) + oo)]))
} } }
} } } } } } } }
TVM IR (opt.)
wmma::mma_sync(...)
GPU-TC