

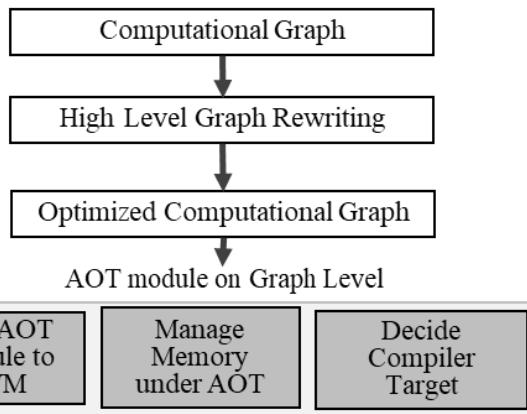
Towards Optimized Tensor Code Generation for Deep Learning on Sunway Many-Core Processor

**Mingzhen LI, Changxi LIU, Jianjin LIAO, Xuegui ZHENG,
Hailong YANG , Rujun SUN, Jun XU, Lin GAN,
Guangwen YANG, Zhongzhi LUAN, Depei QIAN**

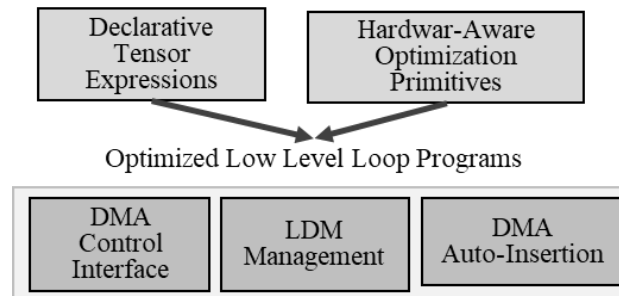
Frontiers of Computer Science, DOI: [10.1007/s11704-022-2440-7](https://doi.org/10.1007/s11704-022-2440-7)

Problems & Ideas

- Problems of optimized tensor code generation on Sunway:
 - The unique compilation environment and architecture features prevent a naive adoption of DL compiler (e.g., TVM) to Sunway.
 - Needs the support of ahead-of-time code generation for MPE/CPE.
 - Needs automatic DMA control, LDM management, and parallelization.
- Ideas: Propose swTVM, a deep learning compiler extending TVM to support optimized tensor code generation for deep learning on Sunway processor automatically.



AOT module at graph level



Optimization and code generation

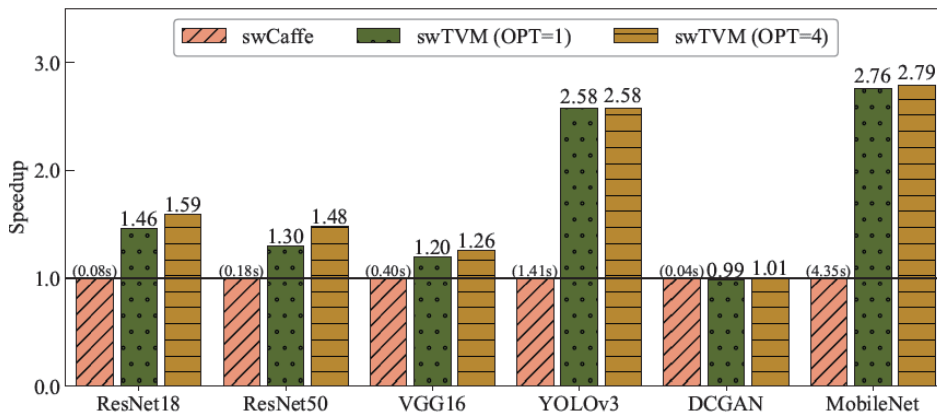
```

C = A * B
1 M=1, K=N=1024
2 A = tvm.placeholder((M,K), name='A')
3 B = tvm.placeholder((K,N), name='B')
4 C = tvm.compute((M,N), lambda x,y:
5     tvm.sum( A[x,k] * B[k,y] , axis =
6     k),
7     name = "C")
8 s = tvm.create_schedule(C.op)
9 yo,yi = s[C].split(C.op.axis[1], 128)
10 ko,ki = s[C].split(k, 64)
11 s[C].reorder(yo,ko,yi,ki)
(a)
1 For x in range(0,1)
2 For y.o in range(8)
3 For y.i in range(128)
4 For k.o in range(16)
5 For k.i in range(64)
6 BB[k.i,y.i] =
7     B[k.o*64+k.i][y.o*128+y.i]
8
9 AA[k.i] =
10     A[x,ko*64+ki]
11
12 CC[y.i] += AA[k.i] *
13     BB[k.i,y.i]
14
15 BB[k.i,y.i] =
16     C[x,y.o*128+y.i]
(b)
1 BB = s.buffer_read(B, [ki,yi])
2 AA = s.buffer_read(A, [ki])
3 CC = s.buffer_write(C, [yi])
Automatic (c)
1 For x in range(0,1)
2 For y.o in range(8)
3 For y.i in range(128)
4 For k.o in range(16)
5 For k.i in range(64)
6 BB[k.i,y.i] =
7     B[k.o*64+k.i][y.o*128+y.i]
8
9 AA[k.i] =
10     A[x,ko*64+ki]
11
12 CC[y.i] += AA[k.i] *
13     BB[k.i,y.i]
14
15 BB[k.i,y.i] =
16     C[x,y.o*128+y.i]
(d)
1 For x in range(0,1)
2 For y.o in range(8)
3 For y.i in range(128)
4 For k.o in range(16)
5 For k.i in range(64)
6 dma(BB[k.i,y.i], B[k.o*64+k.i][y.o*128+y.i], 128)
7 dma(AA[k.i], A[x][ko*64+ki], 64)
8
9 For y.i in range(128)
10 For k.i in range(64)
11 CC[y.i] += AA[k.i] * BB[k.i,y.i]
12 dma(C[x,y.o*128+y.i], CC[y.i], 128)
(e)
    
```

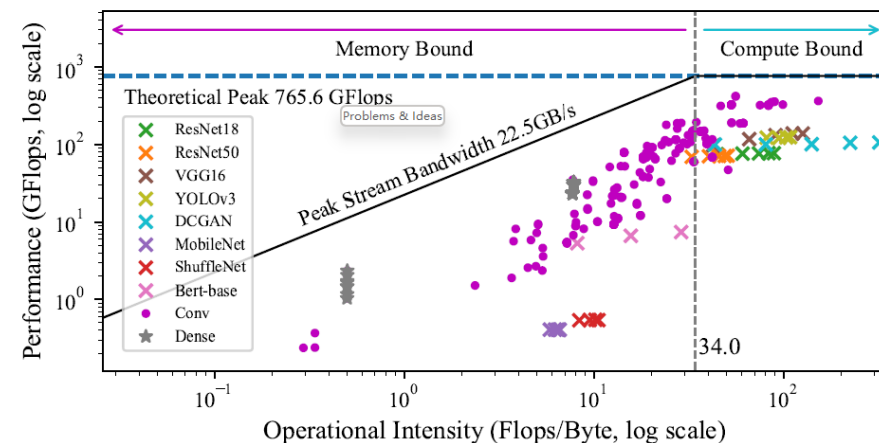
A matmul example

Main Contributions

- Contributions:
 - We implement the ahead-of-time code generation, that produces different compilation targets for MPE and CPE as well as manages the function calls between MPE and CPE efficiently; and we manage the intermediate memory space for each tensor operation globally;
 - We apply several optimizations to the tensor operations regarding the unique architecture features on Sunway, including a DMA control interface, a LDM management mechanism, and DMA instruction inserting mechanism.



The end-to-end performance of swTVM compared to the state-of-the-art DL framework swCaffe.



The roofline analysis of swTVM on eight representative DL models when bs=1.