The CoRa Tensor Compiler: Compilation for Ragged Tensors With Minimal Padding

Pratik Fegade[1],
Tianqi Chen[1,2], Phillip B. Gibbons[1], Todd C. Mowry[1]

[1]Carnegie Mellon University [2]OctoML

Carnegie Mellon University
Computer Science Department

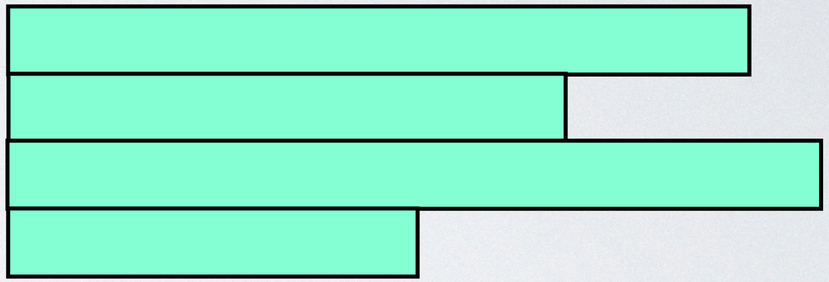
catalyst

Ragged Tensors in Deep Learning

- Natural language processing

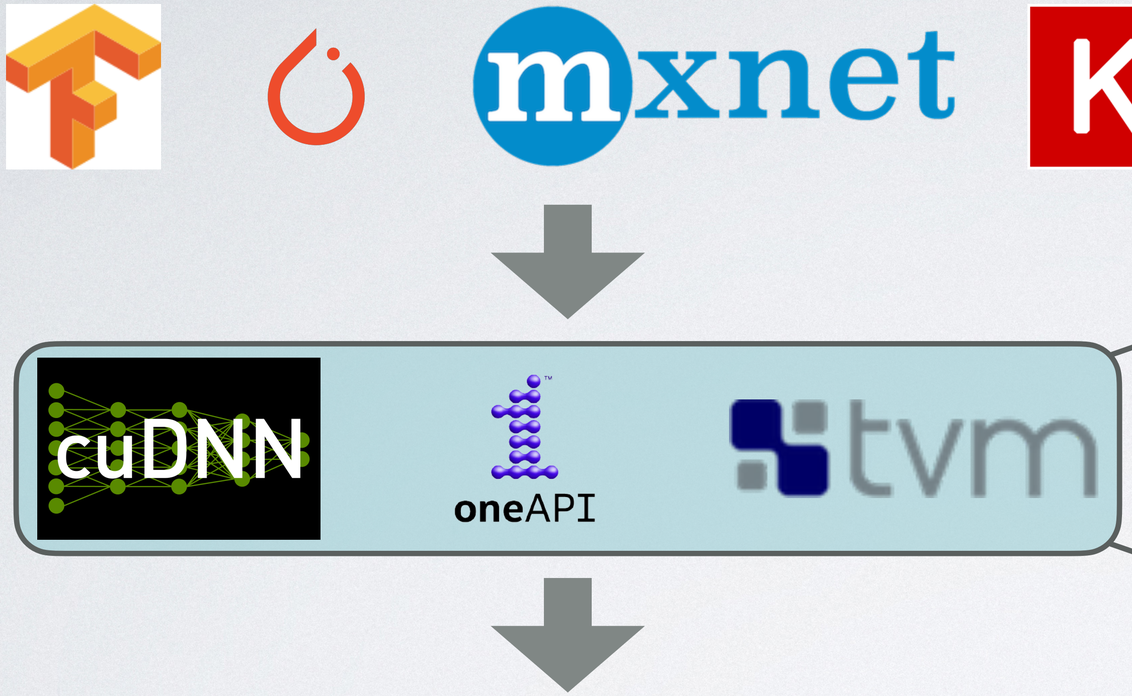
```
   input_batch = [
3    [Dogs,  bark, .],
5    [Maine, is,   a,      state, .],
4    [The,   song, rocks, !],
1    [Hello]
   ]
```

- Image processing



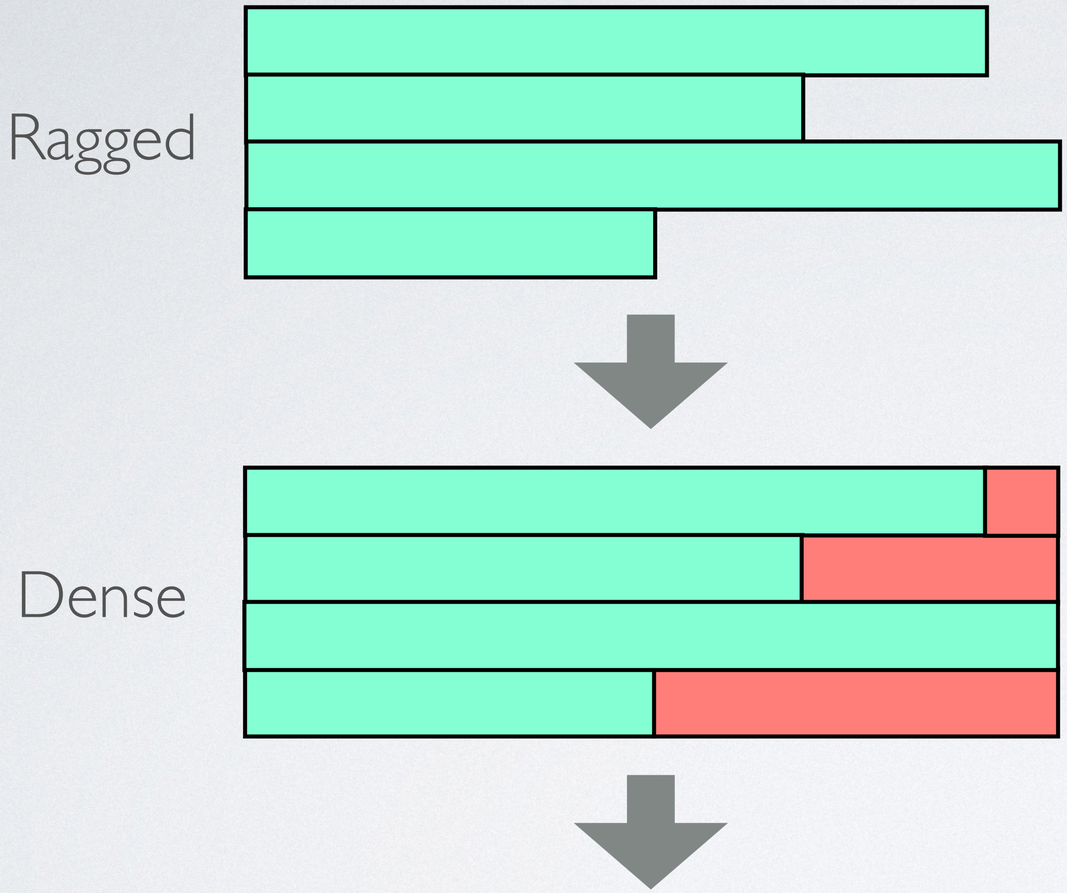
Ragged Tensor

Limited Support for Ragged Tensor Operators



- Limited support for ragged tensors
- Extensive support for dense tensors

Padding Leads to Wasted Computation



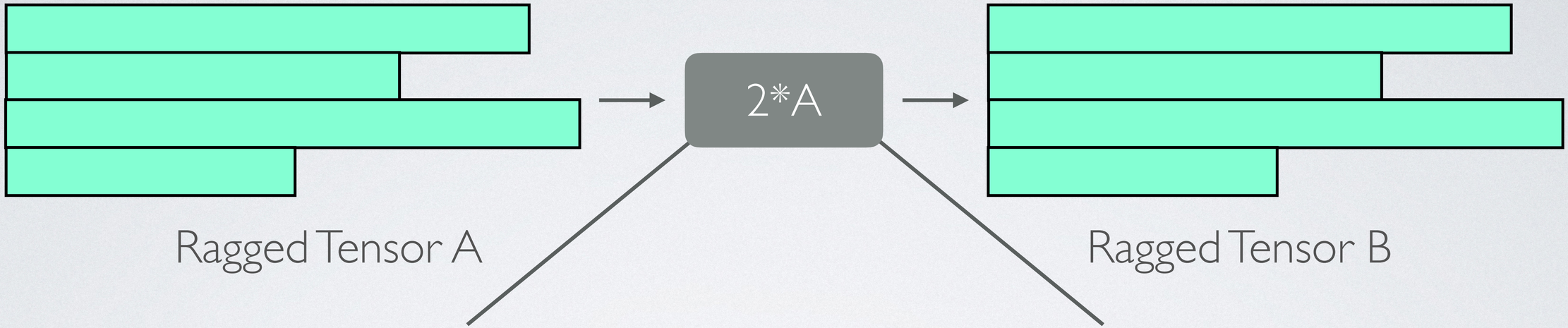
1.07 - 2.41X wasted computation for a transformer encoder layer!

cuDNN

oneAPI

tvm

Ragged Computations Are Similar to Dense Computations



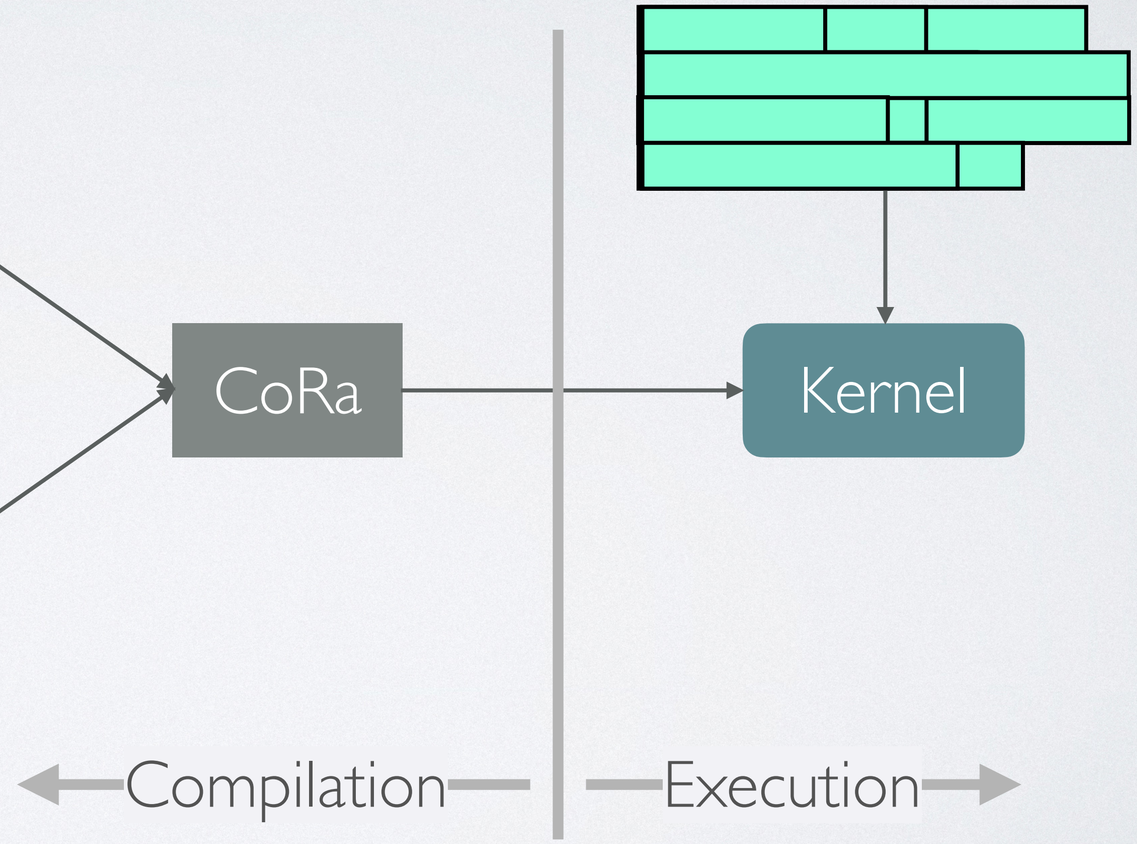

```
for i in 0:32:
  for j in 0:s(i):
    B[i,j] = 2*A[i,j]
```

CoRa: a Tensor Compiler for Ragged Tensors

```
for i in 0:32:
  for j in 0:s(i):
    B[i,j] = 2*A[i,j]
```

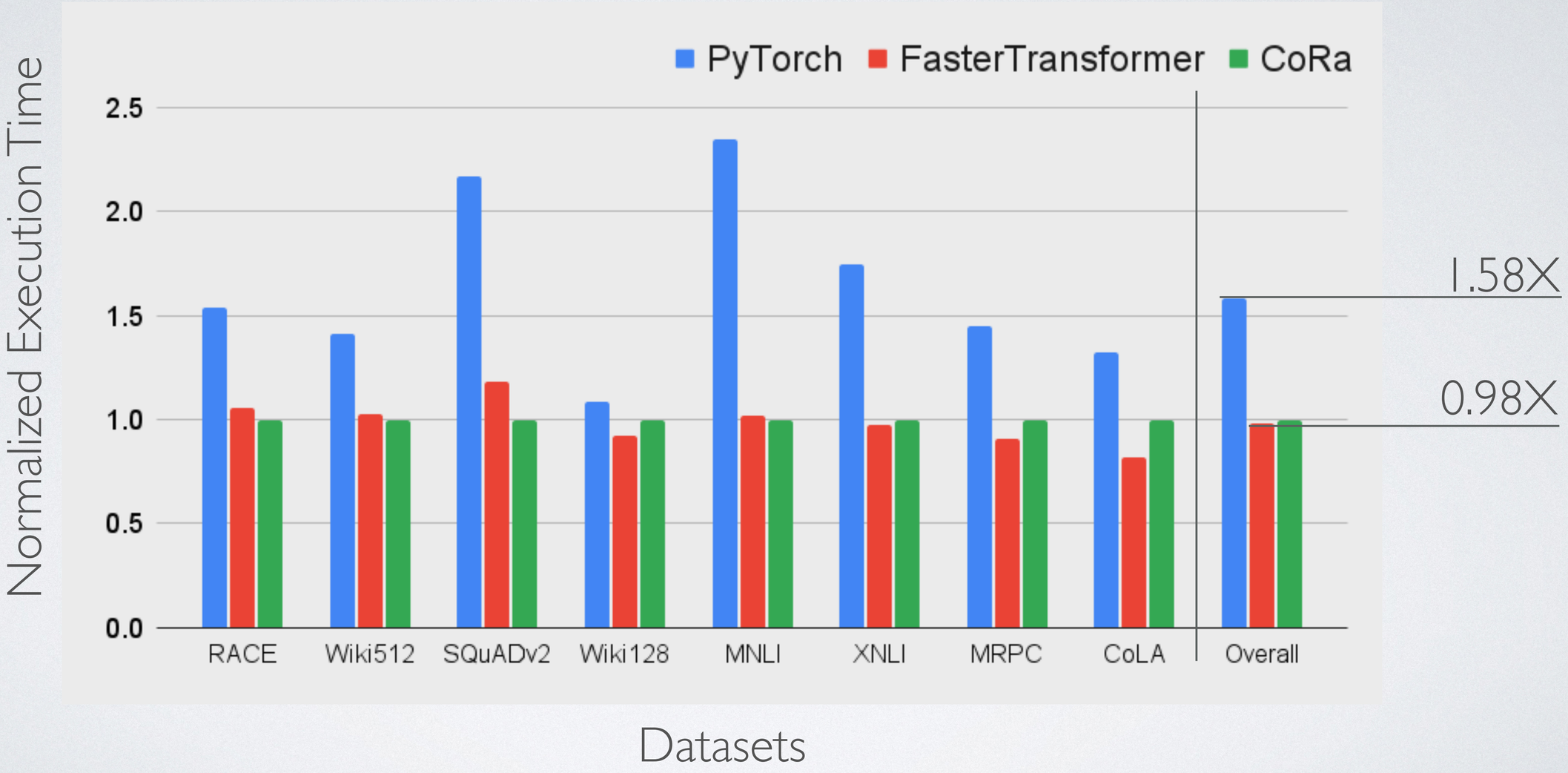
```
loop_exts = [32, lambda b: s(b)]
A = input_tensor(loop_exts)
O = compute(loop_exts,
            lambda i,j: 2*A[i,j])

i, j = O.axis
s.padding(j, 32)
jo, ji = s.split(j, nparts=32)
s.bind(i, 'blockIdx.x')
s.bind(jo, 'threadIdx.x')
```

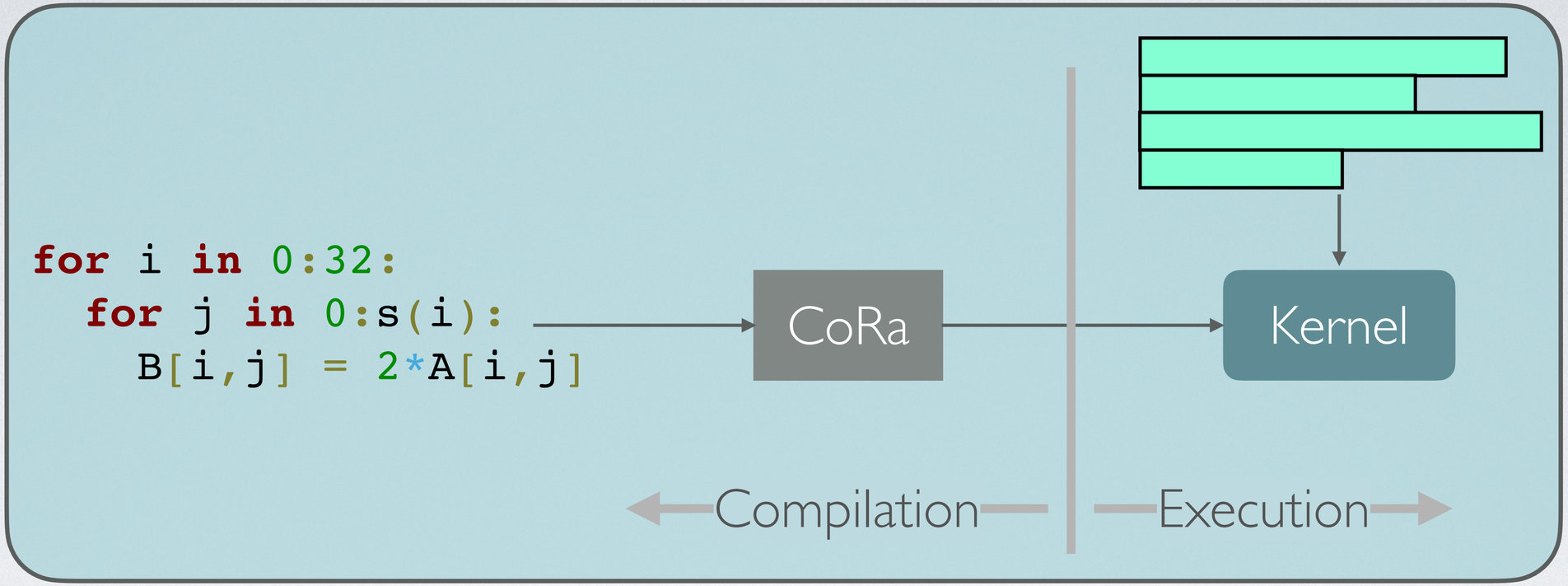


Transformer Layer Layer Forward Latencies on Nvidia V100

Lower is better



Conclusion



More details can be found in our paper