Extending Auto-scheduler to Support Performance Evaluation with Timing Model

Minchun Liao, NTHU
Yaohua Chen, ITRI
Chungta King, NTHU
Motivation

• Current TVM Auto-scheduler needs real devices or simulators to provide timing information for optimized code scheduling.

• If the device and/or its simulator are still in development, it is difficult to get the execution time.

• In the mean time, we still want the compiler to generate optimized code for the device in development for testing and design optimization.

• This work proposes the use of a timing model in TVM to approximate timing.
  • As a demonstration, we extend TVM Auto-scheduler to generate schedule for Computing-in-Memory (CIM) devices, like ReRAM-based accelerators.
CIM Behavior Model

- 168 PEs/chip, 12 CUs/PE, 8 Memristor arrays/CU
- Memristor array size: 128*128
• Design sketch rules, annotation rules and mutation rules for CIM
• Leverage Auto-scheduler to generate schedule and perform schedule tuning
Runner

- Add a simulation runner into TVM
- Return the time cost

- RPC Runner
  - Run generated code on remote devices

- Local Runner
  - Run generated code on local devices

- Simulation Runner
  - Evaluate performance with timing model

New
Timing model

• Approximate the convolution execution time on CIM device

\[
total_{\text{cycle}} = \text{number}_{\text{batch}} \times \text{cycle}_{\text{batch}} + 6
\]

\[
\text{cycle}_{\text{batch}} = \frac{\text{feature map resolution}}{\text{DAC resolution}}
\]

\[
\text{number}_{\text{batch}} = \left\lceil \frac{\text{kernel size}^2 \times \text{IC}}{\text{#used WL}} \times \frac{\text{OC}}{\text{#filters for MA}} \times \text{#MA} \times \text{#CU} \times W \times H \times N \right\rceil
\]

\- \text{OC} : Output channels
\- \text{IC} : Input channels
\- \text{W} : Width of input feature map over stride
\- \text{H} : Height of input feature map over stride
\- \text{N} : Number of input feature maps
\- \text{#used WL} : How many wordlines are used in a memristor array
\- \text{#filters for MA} : How many filters are mapped in a memristor array
\- \text{#MA} : Number of memristor arrays in one CU
\- \text{#CU} : Number of computation units in one PE

DAC resolution: 1 bit

Filter 1
Filter 2
Filter 3

Feature Map

Cycle 1
Cycle 2
Summary

• Use a timing model to work with TVM Auto-scheduler if the device or the simulator is not ready yet.

• Extend TVM Auto-scheduler to get the time cost through the timing model.

• During development of the device, the optimized code generated by TVM can be used to facilitate the development, e.g., for testing and design optimization.
Thank You!
Video Download Link

• https://drive.google.com/file/d/1BKlGeTSIATv1YJhxg2ND4_5Dv8wrJbTh/view?usp=sharing