MAESTRO: An Open-source Infrastructure for the Cost-Benefit Analysis of Dataflows within Deep Learning Accelerators

Hyoukjun Kwon
(Graduate Student)
Georgia Institute of Technology
Atlanta, Georgia
hyoukjun@gatech.edu

Tushar Krishna
(Academic Advisor)
Georgia Institute of Technology
Atlanta, Georgia
tushar@ece.gatech.edu

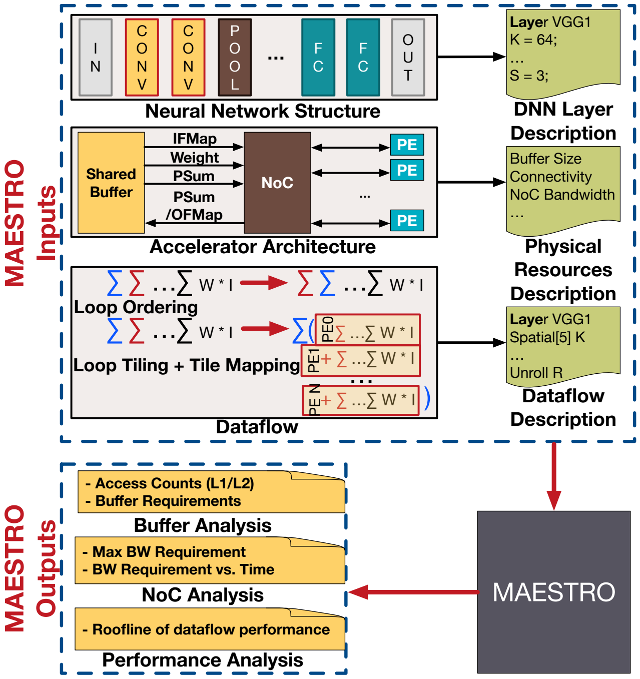

Figure 1: An overview of MAESTRO tool chain.

ACM Reference Format:
Hyoukjun Kwon and Tushar Krishna. 2018. MAESTRO: An Open-source Infrastructure for the Cost-Benefit Analysis of Dataflows within Deep Learning Accelerators. In *Proceedings of ACM Student Research Competition (SRC-MICRO2018).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 MOTIVATION

Efficiently tiling and mapping high-dimensional convolutions onto limited execution and buffering resources is a challenge faced by all deep learning accelerators today. We term each unique tiling and mapping approach as dataflow. The dataflow determines overall throughput (utilization of compute units) and energy-efficiency (number of reads/writes and the reuse of model parameters and partial sums across the acceleratorâĂŹs memory hierarchy). However, the research community today lacks an infrastructure to evaluate deep neural network (DNN) dataflows and architectures systematically and reason about performance, power, and area implications of various design choices. To provide the missing infrastructure for the research community, we develop a framework called MAESTRO that enables to formally describe and analyze DNN dataflows, predict roofline performance and energy-efficiency when running neural network layers, and report the amount of hardware resources (size of buffers across the memory hierarchy, and network-on-chip (NoC) bandwidth) to support a dataflow, as illustrated in Fig. 1.

Constants Definition
LayerName ∈ (String) sz, ofs, bw ∈ (Integer)
LoopVar := **K** | **C** | **R** | **S** | **Y** | **X**
Layer := CONV | POOL | FC

Physical Resource Description
PEInfo:= NumPEs (*sz*)
BufferInfo := L1Size (*sz*) | L2Size (*sz*)
NoCInfo := Bandwidth (*bw*)

DNN Layer Description
LoopSz := (*LoopVar*) (*sz*)
LayerInfo := Layer (*LayerName*) (*LoopSz*)* endLayer

Dataflow Description
PragmaBody := Tile (*sz*) | Spatial_Map (*sz,ofs*)
| Temporal_Map (*sz,ofs*) | Unroll
Pragma := (*PragmaBody*) (*LoopVar*)
Dataflow := (*Pragma*)*
HWInfo := (*BufferInfo*)* (*NoCInfo*)* (PEInfo)
MAESTO := (*HWInfo*) (*LayerInfo*) (*Dataflow*)

(a) Syntax

```
//Physical Resource Description
 L1Size 256;
 L2Size 1024;
 Bandwidth 32;
 NumPEs 256;

//DNN Layer Description
 Layer CONV VGG_C1;
   K 64; C 3; R 3; S 3; Y 224; X 224;
 endLayer

//Dataflow Description
Temporal_Map (1,1) K
-> Temporal_Map (1,1) C
-> Temporal_Map (1,1) Y
-> Spatial_Map (1,1) X
-> Unroll R
-> Unroll S
```

(b) Example

Figure 2: he syntax of MAESTRO DSL and an example of it. (a) The unit of size is the number of elements (e.g., L1size 256 indicates the size of 256 elements, 512 Byte with 16-bit fixed point data). We use $*$ as a shorthand for semicolon-terminated repetition. (b) The example is for a weight-stationary dataflow

2 APPRAOCH

Dataflow Description DSL. MAESTRO provides a concise domain specific language (DSL) to describe dataflow describved in Fig. 2. The DSL consists of hardware resource, DNN layer, and dataflow description sytax. Hardware resource description syntax allows to specify number of PEs, memory size, and NoC bandwidth. DNN layer description sytax allows to specify the size of each dimension in 6D (or 7D if include batch iteration) loops. Dataflow description syntax provides four mapping, tiling, and loop unrolling pragmas inspired by parallel programming libraries in software domain, which is a familiar style to many programmers that minimizes the learning curve for MAESTRO DSL. We describe dataflows of recent deep learning accelerators using MAESTRO DSL as presented in Table 1 and evaluate those dataflows using MAESTRO analysis engine.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SRC-MICRO2018, October 2018, Fukuoka, Japan
© 2018 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

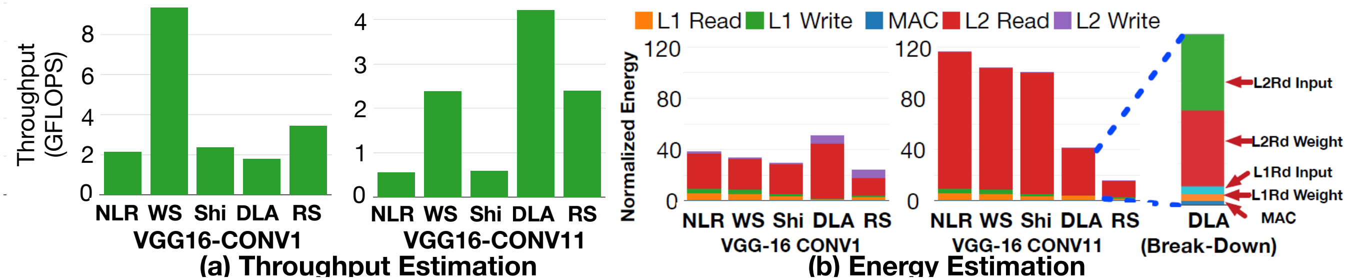

Figure 3: A cost-benefit analysis based on MAESTRO with 64 PEs and a bus NoC with bandwidth of 8X (deliver 8 data points at the same cycle). NLR is no-local reuse, WS is weight-stationary, Shi is Shi-diannao [1], DLA is NVDLA [2], and RS is row-stationary [3]. Note that this is a modeling of a dataflow, not the architecture. Energy consumption is normalized to that of MAC operation.

Accelerator	Dataflow Strategy	Dataflow
Example for this work	No Local Reuse (NLR)	Temporal_Map (1,1) **K** → Temporal_Map (1,1) **C** → Temporal_Map (1,1) **Y** → Temporal_Map (1,1) **R** → Temporal_Map (1,1) **S** → Spatial_Map (1,1) **X**
Example for this work	Weight Stationary (WS)	Temporal_Map (1,1) **K** → Temporal_Map (1,1) **C** → Temporal_Map (3,3) **Y** → Spatial_Map (3,1) **X** → Unroll **R** → Unroll **S**
ShiDiannao [1]	Output Stationary (OS)	Temporal_Map (1,1) **K** → Temporal_Map (1,1) **C** → Temporal_Map (3,1) **Y** → Spatial_Map (3,1) **X** → Unroll **R** → Unroll **S**
Eyeriss [3]	Row-stationary	Temporal_Map (1,1) **K** → Temporal_Map (1,1) **C** → Spatial_Map (3,1) **Y** → Tile (Sz(R)) **X** → Temporal_Map (3,1) **X** → Spatial_Map (1,1) **R** → Unroll **S**
NVDLA [2]	Weight Stationary	Temporal_Map (1,1) **R** → Temporal_Map (1,1) **S** → Temporal_Map (64,64) **C** → Temporal_Map (1,1) **Y** → Temporal_Map (1,1) **X** → Spatial_Map (1,1) **K**

Table 1: A list of dataflows with description based on pragmas introduced in Fig. 2 and corresponding accelerators.

Analytic Cost-benefit Model. MAESTRO analysis engine receives a dataflow written in MAESTRO DSL, neural network dimensions, and hardware resource information as inputs. Using the information, MAESTRO performs cost-benefit analysis based on an analytic model we developed and reports activity counts (buffer access, arithmetic ops, NoC traversals, and so on), hardware resource requirement, and roof-line throughput for input dataflows. The analytic model is based on loop-nest analysis with mapping information embedded in dataflow description. The model considers all types of data reuse - temporal (stationary data), spatial (multicast), and spatio-temporal (inter-PE forwarding) - to identify precise number of buffer accesses and buffer size requirements. Also, the performance model includes the communication delay analysis based on the NoC description in DSL. Unlike previous works in compiler domain, we target spatial accelerators with machine learning workload and utilizes detailed architecture information that encompasses not only computation and buffer but also NoC resources.

3 RESULTS

We present trade-off study results that include energy consumption based on activity counts integrated with Cacti [4], roof-line performance, and buffer requirements Fig. 3 presents a part of the study we perform, which shows expected throughput and energy consumption analysis of two synthetic and three realistic dataflows under hardware resource constraints (64 PEs, 8X bus NoC) for two convolutional layers in VGG-16 [5].

WS (weight-stationary) dataflow provides large throughput for an early layer (VGG16-CONV1) because early layers have less number of weights, so keeping weight values stationary and braodcasting inputs is beneficial for throughput. DLA (NVDLA) dataflow does not perform very well in early layers because it spatially maps output channels but early layers have smaller number of output channels than the number of PEs in NVDLA, which leads to compute unit underutilization. However, a late layer (VGG16-CONV11), DLA dataflow performs best because layer layers have large number of output channels so DLA dataflow can fully utilize the output channel parallelism. RS (row stationary) dataflow [3] consumes the least energy in this case study. The save is based on the maximized data reuse in a PE array; converting expensive L2 buffer access to L1 read and local forwarding between PE pairs. The results presented in Fig. 3 imply that no single dataflow is the best for all the layers and optimization goals.

4 CONTRIBUTION

Because the throughput-energy optimization space is high-dimensional(six layer dimensions, number of PEs, NoC bandwidth, NoC multicast capability, loop order, mapping size, tiling size, etc.), changes in some parameters lead to dramatically different results. Therefore, MAESTRO is a timely tool that provides cost-benefit analysis for any combination of the layer dimensions/shapres, hardware configuration, and dataflow. In summary, this work makes the following three contributions: (1) concise pragma-based-DSL that enables formal dataflow description, (2) providing a dataflow cost-benefit analysis framework, and (3) showing trade-offs among dataflows from recent accelerators (Eyeriss [3], NVDLA [2], etc.), available hardware resources, and target neural network shape(dimensions).

REFERENCES

[1] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "Shidiannao: Shifting vision processing closer to the sensor," in *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 2015.

[2] "Nvdla deep learning accelerator," 2017. http://nvdla.org.

[3] Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 2016.

[4] N. Muralimanohar *et al.*, "Cacti 6.0: A tool to model large caches," *HP laboratories*, pp. 22–31, 2009.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.