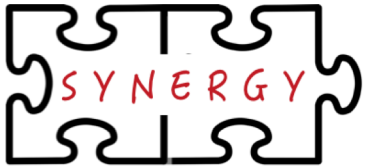
MAESTRO Tutorial – MICRO 2020
October 17th, 2020



Compiling and Running MAESTRO

Geonhwa Jeong, PhD student,
Synergy Lab,
Georgia Institute of Technology

Resources

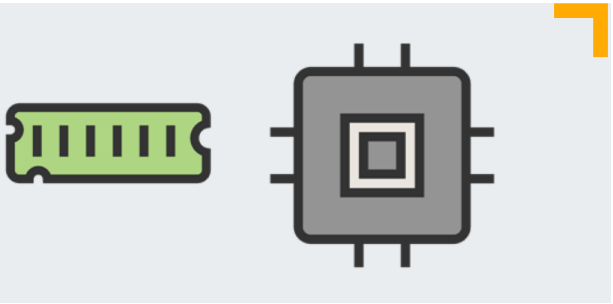
- Tutorial Website
  http://maestro.ece.gatech.edu/docs/build/html/tutorials/micro2020.html

- MAESTRO Website http://maestro.ece.gatech.edu/

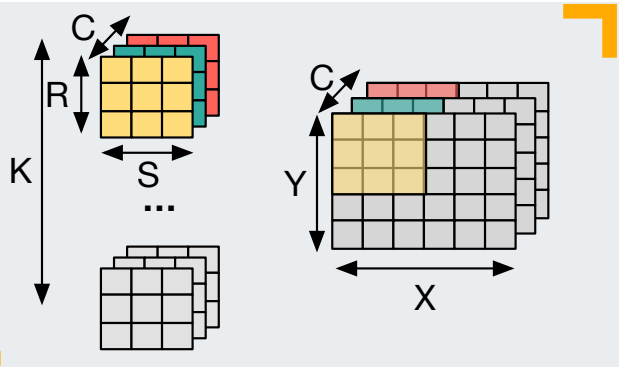
- Publications

1. Understanding Reuse, Performance, and Hardware Cost of DNN Dataflows: A Data-Centric Approach
   Hyoukjun Kwon, Prasanth Chatarasi, Micheal Pellauer, Angshuman Parashar, Vivek Sarkar, and Tushar Krishna, **MICRO 2019**
2. MAESTRO: A Data-Centric Approach to Understand Reuse, Performance, and Hardware Cost of DNN Mappings
   Hyoukjun Kwon, Prasanth Chatarasi, Micheal Pellauer, Angshuman Parashar, Vivek Sarkar, and Tushar Krishna, **IEEE Micro Top Picks 2020**
3. GAMMA: Automating the HW Mapping of DNN Models on Accelerators via Genetic Algorithm
   Sheng-Chun Kao and Tushar Krishna, **ICCAD 2020**
4. ConfuciuX: Autonomous Hardware Resource Assignment for DNN Accelerators using Reinforcement Learning
   Sheng-Chun Kao, Geonhwa Jeong, and Tushar Krishna, **MICRO 2020**

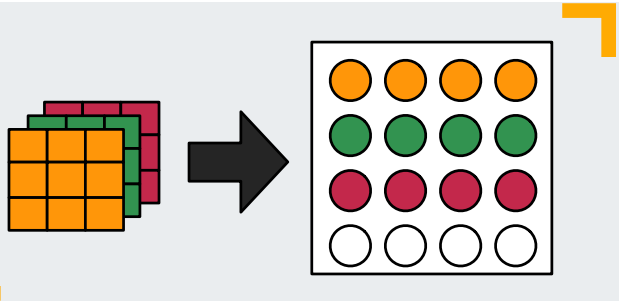
[RECAP] MAESTRO: Analytical Cost/Benefit Model



HW Resources



DNN Layer



Mapping (Dataflow)

MAESTRO

- Abstract HW Model
- Data Reuse Analysis
- Communication Analysis
- Computation Analysis

- Size Requirement
- Access Count (Energy)

Buffer Analysis

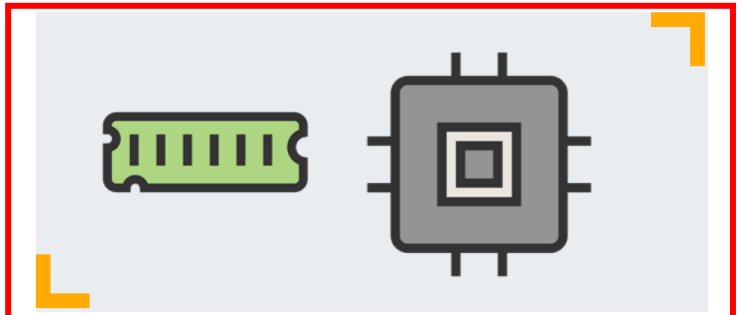
- BW Requirement
- NoC Activity Count

NoC Analysis

- Roofline Throughput
- Expected Runtime

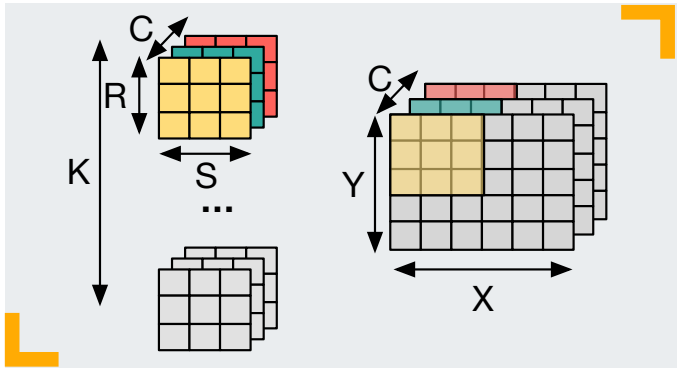
Runtime Analysis

[RECAP] MAESTRO: Analytical Cost/Benefit Model

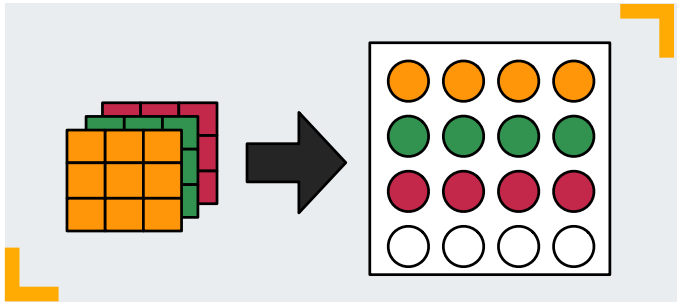


HW Resources

1. HW file



DNN Layer



Mapping (Dataflow)

2. Mapping file

MAESTRO

Abstract HW Model

Data Reuse Analysis

Communication Analysis

Computation Analysis

- Size Requirement
- Access Count (Energy)

Buffer Analysis

- BW Requirement
- NoC Activity Count

NoC Analysis

- Roofline Throughput
- Expected Runtime

Runtime Analysis

Compiling MAESTRO

- Dependencies
  - C++ compiler (which supports c++17)
  - SCONS build system (scons)
  - Boost library (libboost-all-dev)

- Get codes from GitHub and build.

```
>> git clone https://github.com/maestro-project/maestro
>> cd maestro
>> scons
```

  This will create the binary file, *maestro* in the same directory.

- We also provide MICRO 2020 tutorial version Docker Image which includes MAESTRO binary file with proper environment settings.
  - https://hub.docker.com/repository/docker/ghjeong12/maestro-micro2020-tutorial

Running MAESTRO

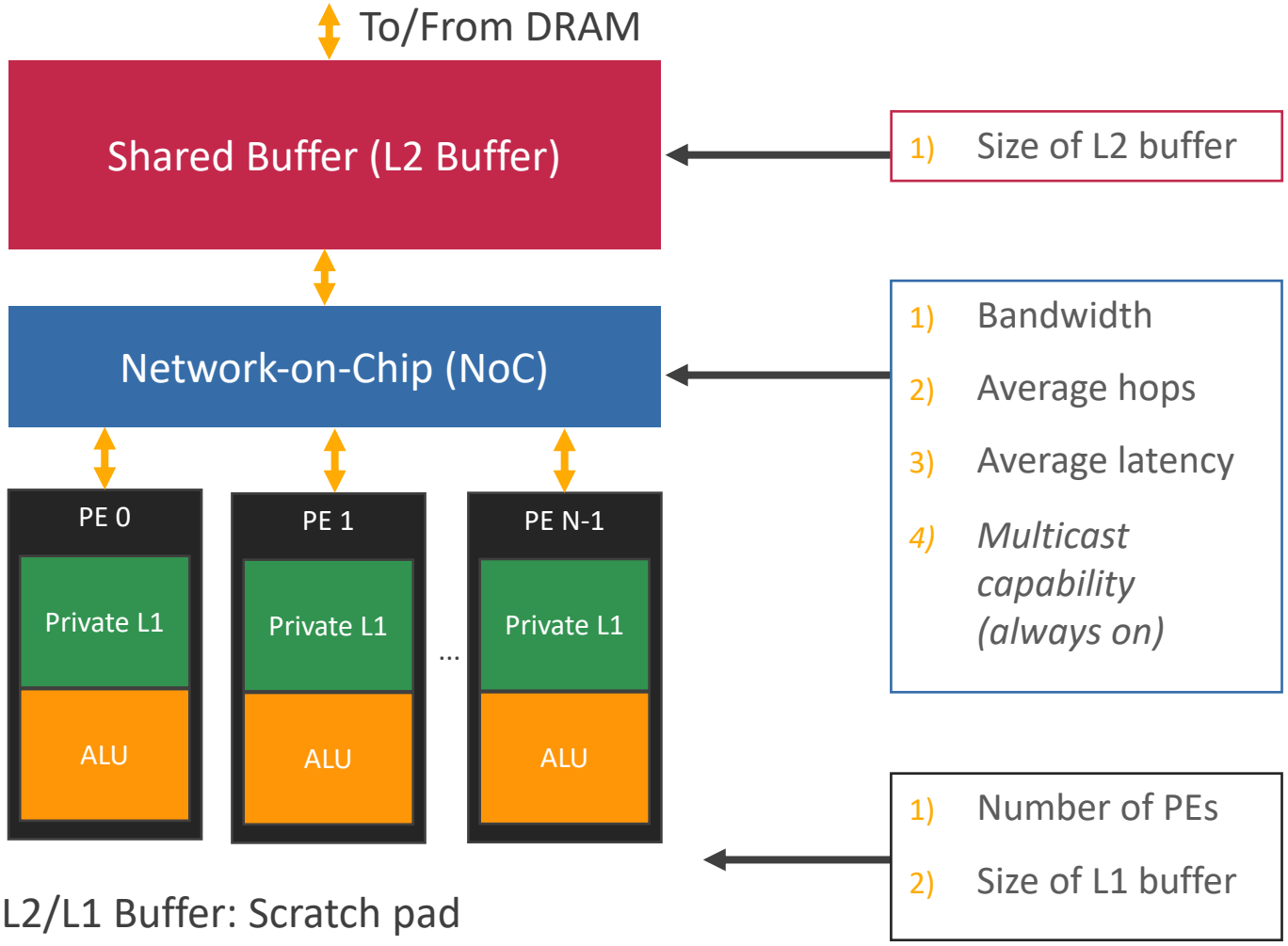
- Now you can run MAESTRO!

```
>> ./maestro --HW_file='data/hw/accelerator_1.m' \
--Mapping_file='data/mapping/Resnet50_kcp_ws.m' \
--print_res=true \
--print_res_csv_file=true \
--print_log_file=false \
```

1. Input files
2. Options

- Open run_example.sh in the top directory to see the command.

Understanding HW File

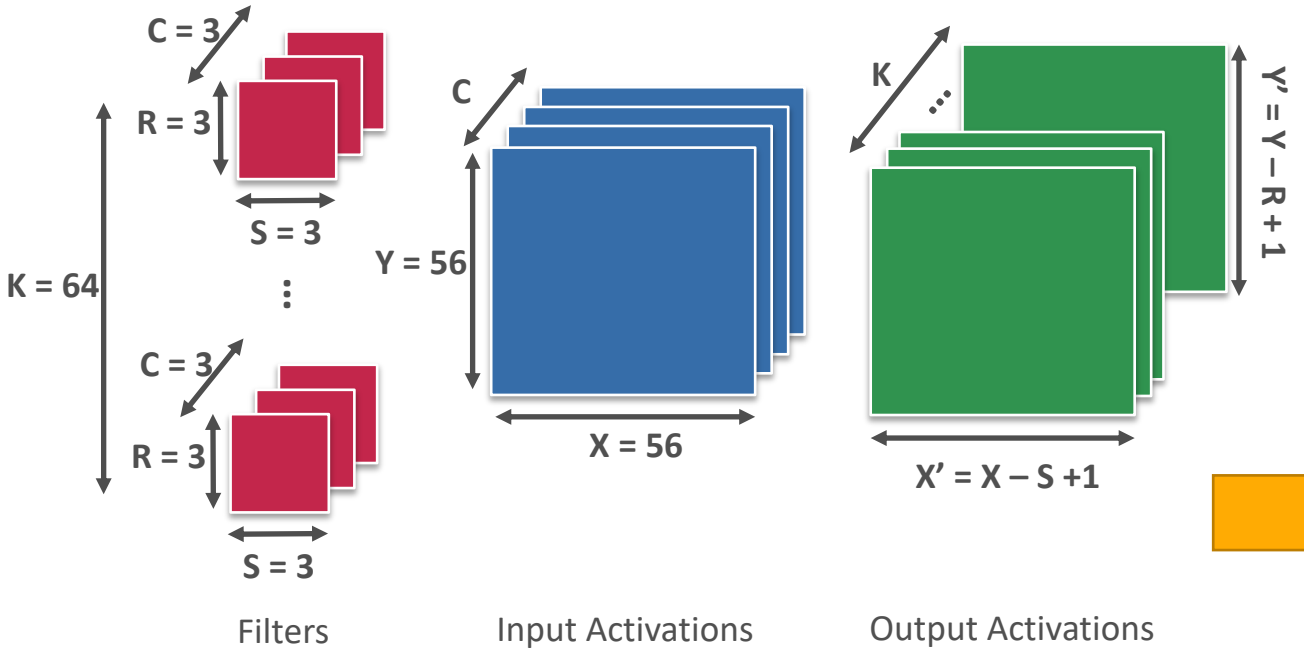


L2/L1 Buffer: Scratch pad
L0 Buffer (in ALUs): Register File

```
NumPEs: 256
L1Size: 512
L2Size: 32768
NoC_BW: 1024
NoC_NumHops: 1
```

maestro/data/hw/accelerator_1.m

Understanding Mapping File



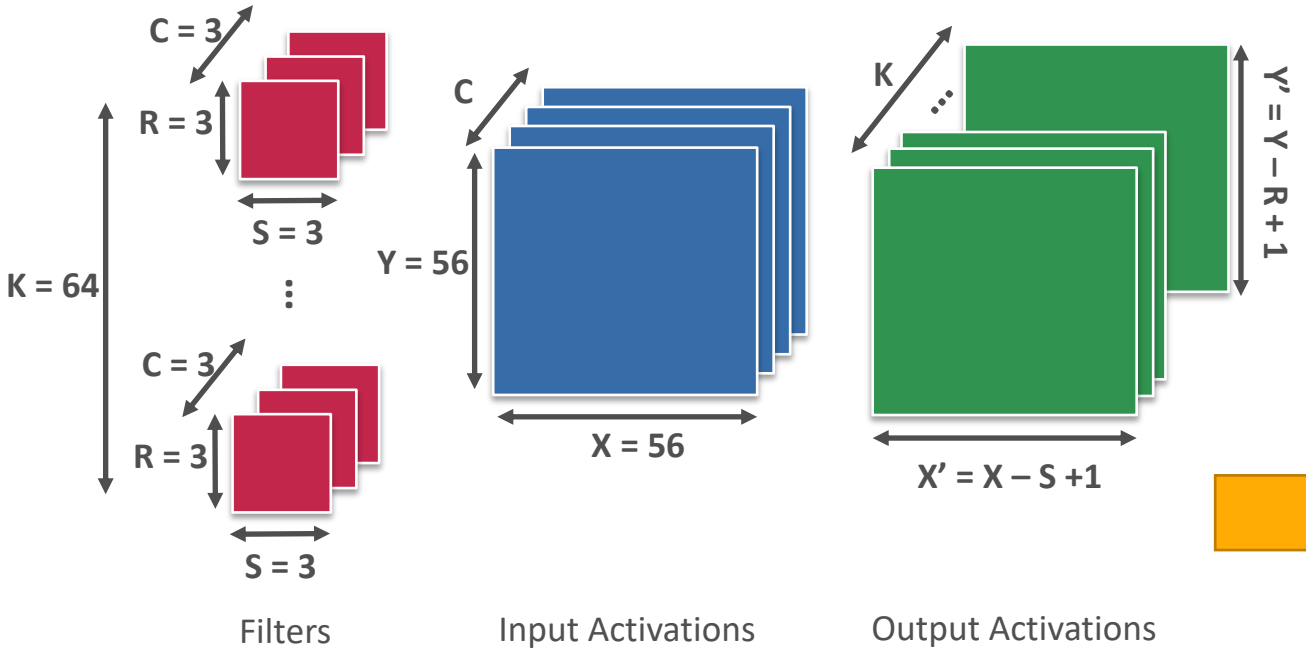
Data Dimensions (Loop Variables)

- **K/C:** Input/output Channel
- **Y/X:** Input Height/Width
- **R/S:** Filter Height/Width
- **N**: Batch

```
Network Resnet50 {
    Layer CONV2_1_2 {
        Type: CONV
        Stride { X: 1, Y: 1 }
        Dimensions { K: 64, C: 64, R: 3, S: 3, Y: 56, X: 56 }
        Dataflow {
            SpatialMap(1,1) K;
            TemporalMap(64,64) C;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            Cluster(64, P);
            SpatialMap(1,1) C;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
        }
    }
}
```

Part of maestro/data/mapping/Resnet50_kcp_ws.m

Understanding Mapping File



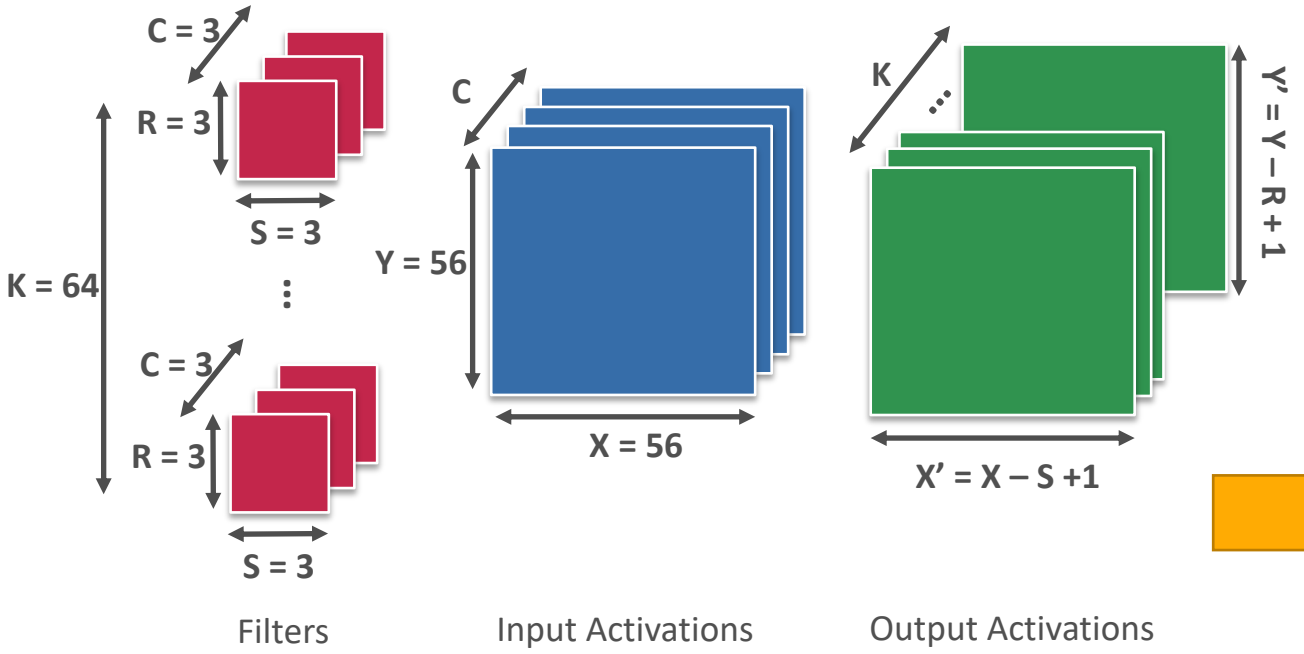
Data Dimensions (Loop Variables)

- **K/C:** Input/output Channel
- **Y/X:** Input Height/Width
- **R/S:** Filter Height/Width
- **N**: Batch

```
Network Resnet50 {
    Layer CONV2_1_2 {
        Type: CONV
        Stride { X: 1, Y: 1 }
        Dimensions { K: 64, C: 64, R: 3, S: 3, Y: 56, X: 56 }
        Dataflow {
            SpatialMap(1,1) K;
            TemporalMap(64,64) C;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            Cluster(64, P);
            SpatialMap(1,1) C;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
        }
    }
}
```

Part of maestro/data/mapping/Resnet50_kcp_ws.m

Understanding Mapping File



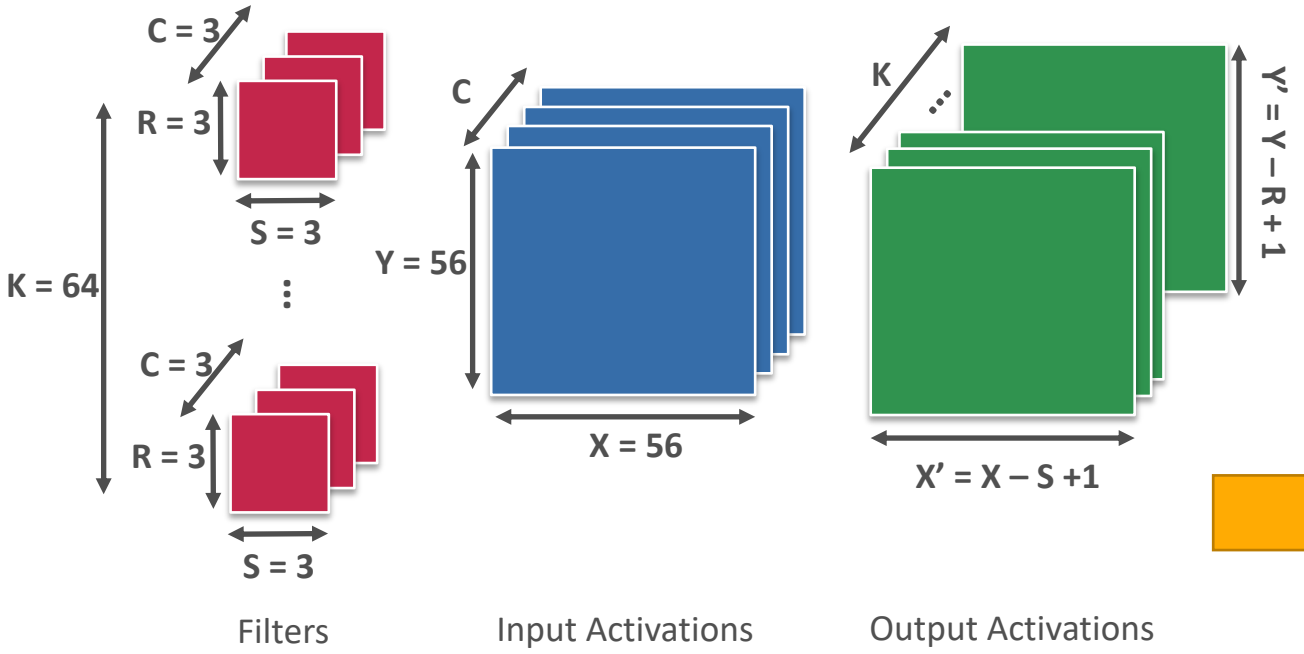
Data Dimensions (Loop Variables)

- **K/C:** Input/output Channel
- **Y/X:** Input Height/Width
- **R/S:** Filter Height/Width
- **N**: Batch

```
Network Resnet50 {
    Layer CONV2_1_2 {
        Type: CONV
        Stride { X: 1, Y: 1 }
        Dimensions { K: 64, C: 64, R: 3, S: 3, Y: 56, X: 56 }
        Dataflow {
            SpatialMap(1,1) K;
            TemporalMap(64,64) C;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            Cluster(64, P);
            SpatialMap(1,1) C;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
        }
    }
}
```

Part of maestro/data/mapping/Resnet50_kcp_ws.m

Understanding Mapping File



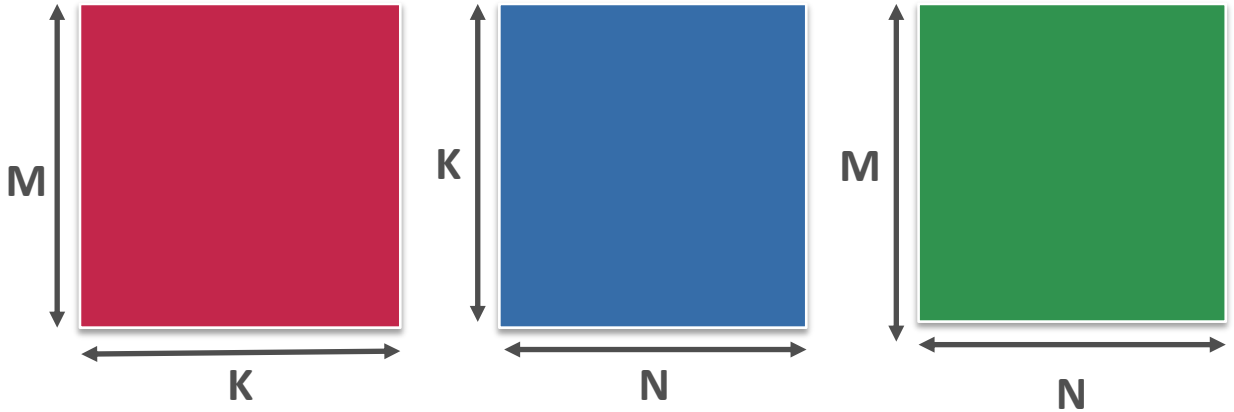
Data Dimensions (Loop Variables)

- **K/C:** Input/output Channel
- **Y/X:** Input Height/Width
- **R/S:** Filter Height/Width
- **N**: Batch

```
Network Resnet50 {
    Layer CONV2_1_2 {
        Type: CONV
        Stride { X: 1, Y: 1 }
        Dimensions { K: 64, C: 64, R: 3, S: 3, Y: 56, X: 56 }
        Dataflow {
            SpatialMap(1,1) K;
            TemporalMap(64,64) C;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            Cluster(64, P);
            SpatialMap(1,1) C;
            TemporalMap(Sz(R),1) Y;
            TemporalMap(Sz(S),1) X;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
        }
    }
}
```

Part of maestro/data/mapping/Resnet50_kcp_ws.m

Understanding Mapping File for GEMM



Data Dimensions

- **Matrix A :** M rows, K cols
- **Matrix B:** K rows, N cols
- **Matrix C:** M rows, N cols

Dense Matrix-Dense Matrix multiplication

A x **B** = **C**

```
Constant SzM 100;
Constant SzN 100;
Constant SzK 100;

Constant MTileSz 1;
Constant NTileSz 1;
Constant KTileSz 100;


Network BLAS3 {
    Layer BLAS {
        Type: GEMM
        Dimensions { K: SzK, M: SzM, N: SzN }
        Dataflow {
            TemporalMap(NTileSz,NTileSz) N;
            SpatialMap(MTileSz,MTileSz) M;
            TemporalMap(KTileSz,KTileSz) K;
            Cluster(KTileSz, P);
            TemporalMap(NTileSz,NTileSz) N;
            TemporalMap(MTileSz,MTileSz) M;
            SpatialMap(1,1) K;
        }
    }
}
```

maestro/data/mapping/GEMM_Example.m

Demo

- Now let’s begin from the MAESTRO website.
  - http://maestro.ece.gatech.edu/

Stay tuned

- Support for non-linear operation
- More NoC topologies for distribution and reduction
  - Bus, Tree, …
- MLIR Integration

Resources

- Tutorial Website
  http://maestro.ece.gatech.edu/docs/build/html/tutorials/micro2020.html

- MAESTRO Website http://maestro.ece.gatech.edu/

- Publications

1. Understanding Reuse, Performance, and Hardware Cost of DNN Dataflows: A Data-Centric Approach
   Hyoukjun Kwon, Prasanth Chatarasi, Micheal Pellauer, Angshuman Parashar, Vivek Sarkar, and Tushar Krishna, **MICRO 2019**
2. MAESTRO: A Data-Centric Approach to Understand Reuse, Performance, and Hardware Cost of DNN Mappings
   Hyoukjun Kwon, Prasanth Chatarasi, Micheal Pellauer, Angshuman Parashar, Vivek Sarkar, and Tushar Krishna, **IEEE Micro Top Picks 2020**
3. GAMMA: Automating the HW Mapping of DNN Models on Accelerators via Genetic Algorithm
   Sheng-Chun Kao and Tushar Krishna, **ICCAD 2020**
4. ConfuciuX: Autonomous Hardware Resource Assignment for DNN Accelerators using Reinforcement Learning
   Sheng-Chun Kao, Geonhwa Jeong, and Tushar Krishna, **MICRO 2020**

Q&A

- Thank you for listening!