A High-level Overview of MAESTRO Mapping Directives and Cost Model

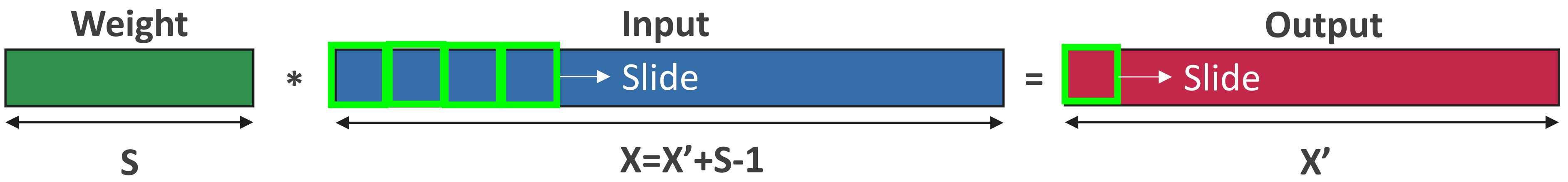
Synergy Lab, Georgia Tech
Hyoukjun Kwon

Outline

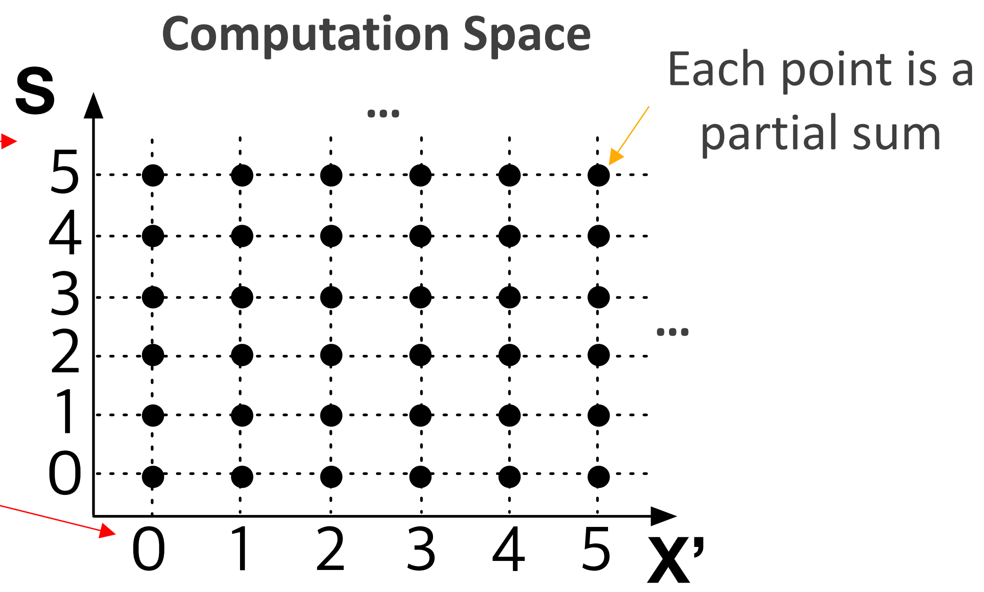
- Mapping Representation: A data-centric representation
  - Computation and Data Space
  - Data-centric Directives
  - Deep-dive Example: Eyeriss-like Dataflow
- MAESTRO Cost Model – High Level Overview

Computation Space of CONV1D

CONV1D Op

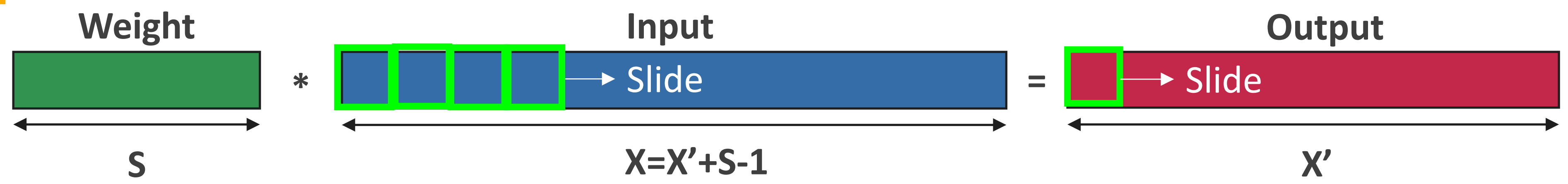



```
for(int s = 0; s < S; s++)
  for(int x’ = 0; x’ < X’; x’++)
    PartialSum[x’][s] = Weight[s] * Input[x’+s]
    Output[x’] += PartialSum[x’][s]
```



Data Space of CONV1D

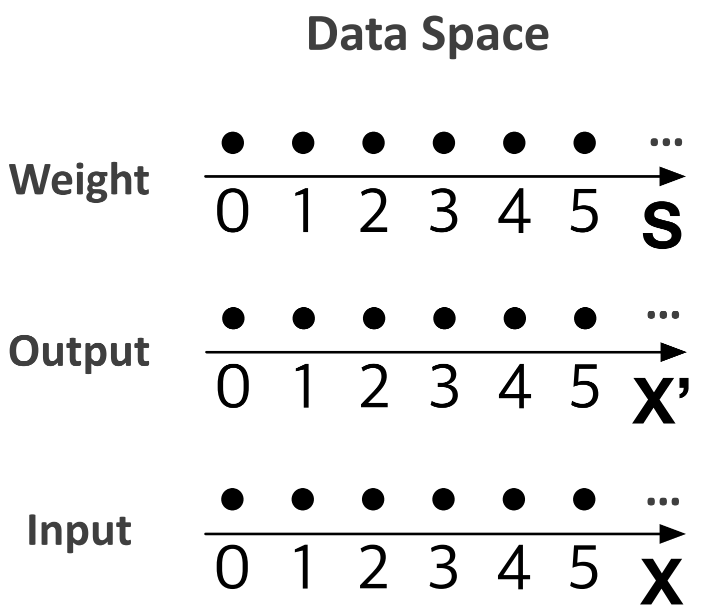
CONV1D Op




```
for(int s = 0; s < S; s++)
  for(int x’ = 0; x’ < X’; x’++)
    PartialSum[x’][s] = Weight[s] * Input[x’+s]
    Output[x’] += PartialSum[x’][s]
```

PartialSum[x’][s]
needs to access:

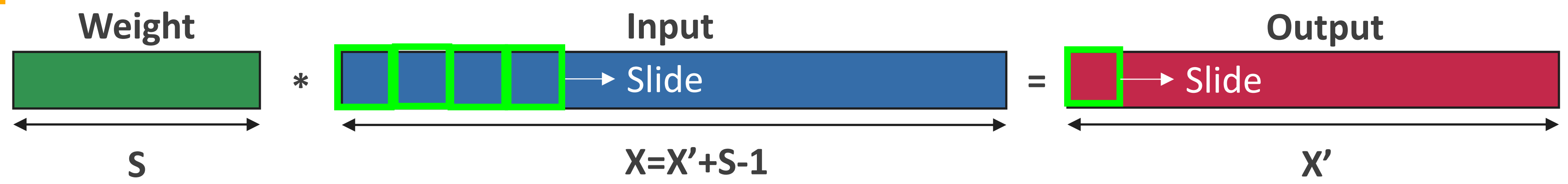
- Weight[s]
- Output[x’]
- Input[x’+s]



Data reuse is behavior in data space!

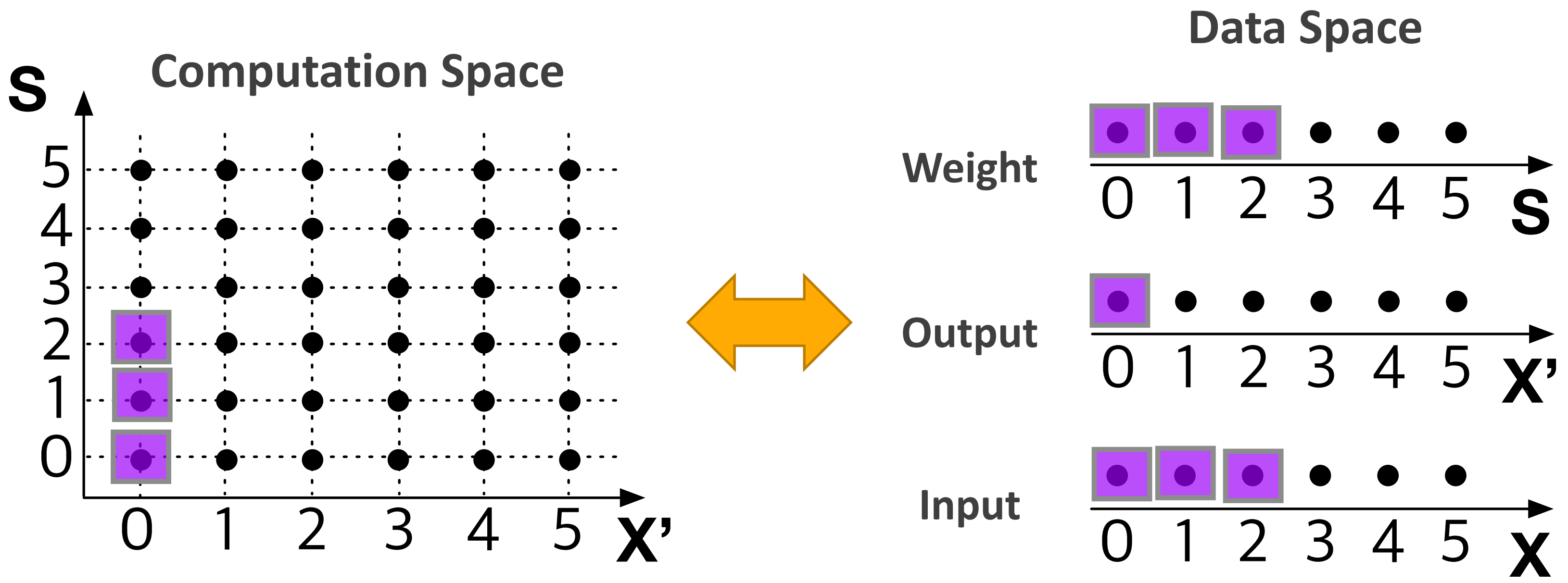
Computation and Data Space

CONV1D Op



PartialSum[x’][s] needs to access:

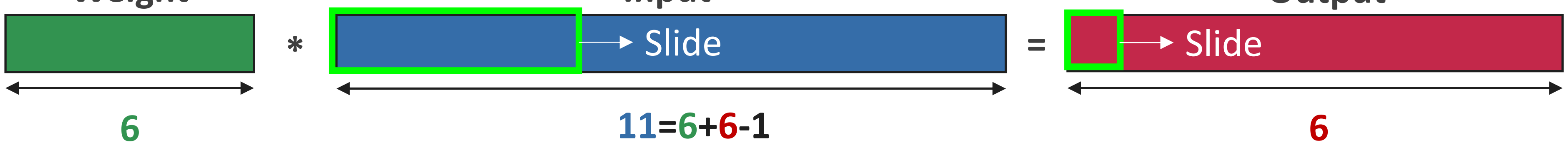
- Weight[s]
- Output[x’]
- Input[x’+s]



Partial sum has 1:1 correspondence to each data tensor

Mapping Example 1: Weight Stationary

CONV1D Op



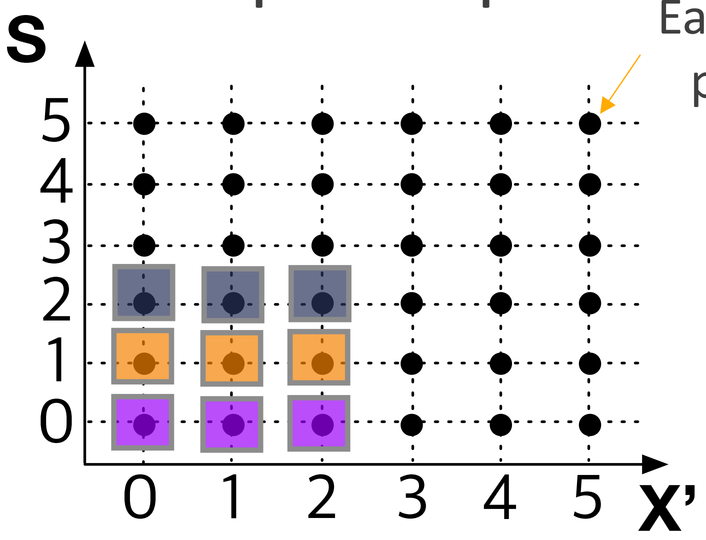
Time = 0

Weight does not change over time

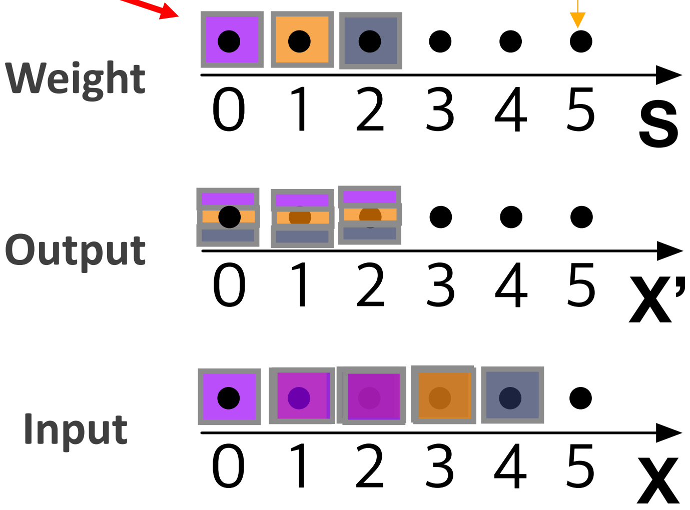
Each point is a data point

Computation Space

Each point is a partial sum



Data Space



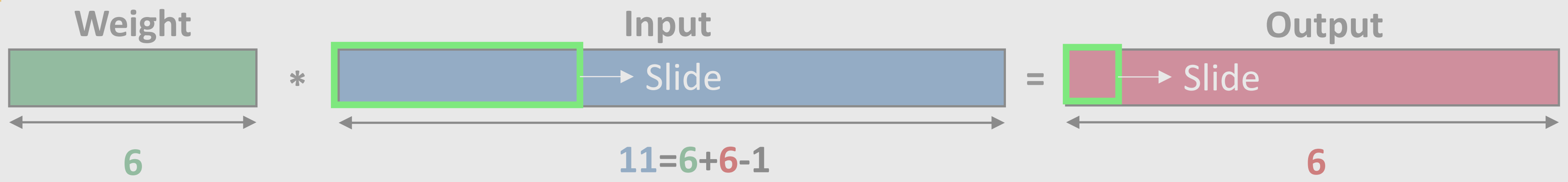
PartialSum[x'][s] needs to access:

- Weight[s]
- Output[x']
- Input[x'+s]

"Weight Stationary" Dataflow

Mapping Example 2: Output Stationary

CONV1D Op



Time = 0

Each point is a data point

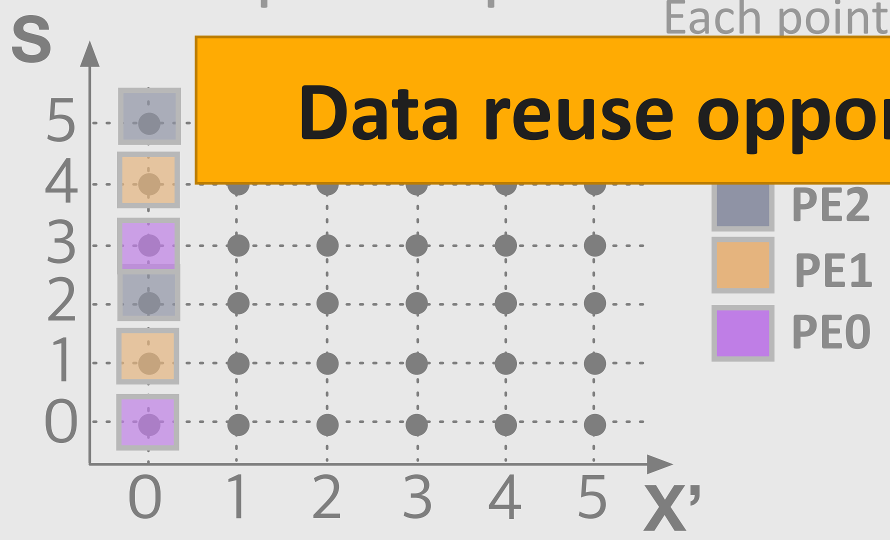
Output does not change over time

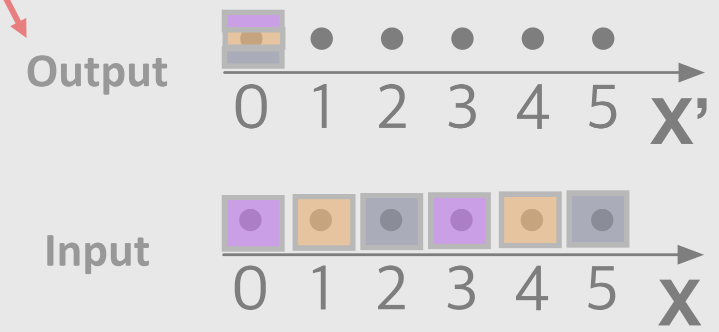
Computation Space

Each point is a

Data Space

Data reuse opportunities are *Explicit* in data space!



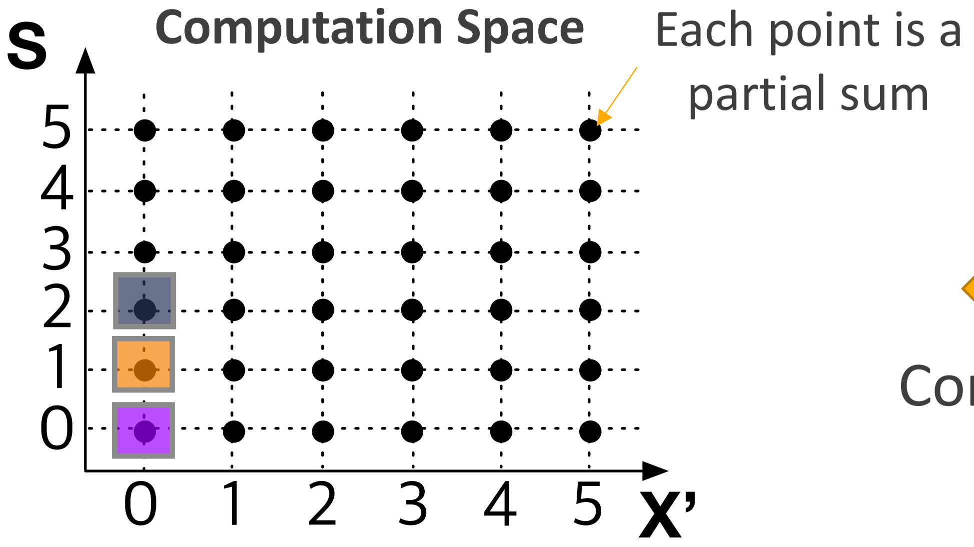


PartialSum[x'][s] needs to access:

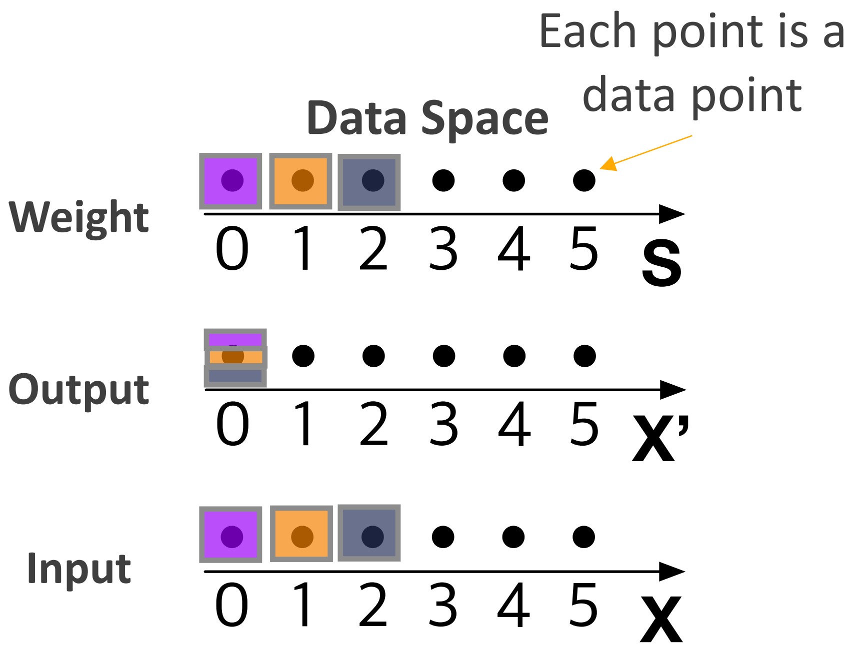
- Weight[s]
- Output[x']
- Input[x'+s]

"Output Stationary" Dataflow

Computation and Data Space



Convertible



- Describes computations (What it does)
- Higher dimensionality than data (CONV2D: 7D loop nest)

Easy for programmers to understand

Representation: *Loop nest*

- Directly describes data mapping (What it uses)

Good to be used as an IR for tools (intermediate representation)!

Easy for tools to analyze data reuse

Representation: *??*

How do we describe data space?

Outline

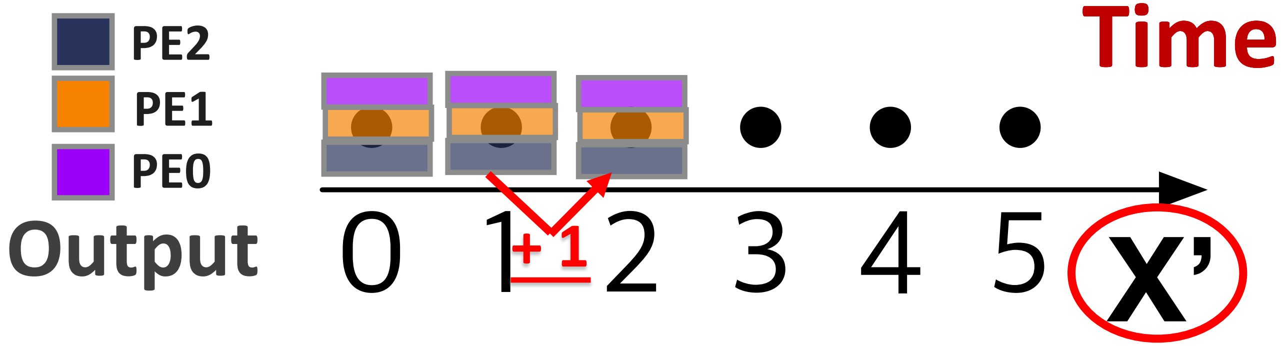
- Mapping Representation: A data-centric representation
  - Computation and Data Space
  - Data-centric Directives
  - Deep-dive Example: Eyeriss-like Dataflow
- MAESTRO Cost Model – High Level Overview

Introducing Data-centric Directives

- **Temporal Map**

Syntax: TemporalMap(Mapping size, Temporal Offset) Dim

Time = 0



Description

TemporalMap(1, 1) *X'*

Mapping is on dimension X' ... PE index + 1

Temporal Offset = 1

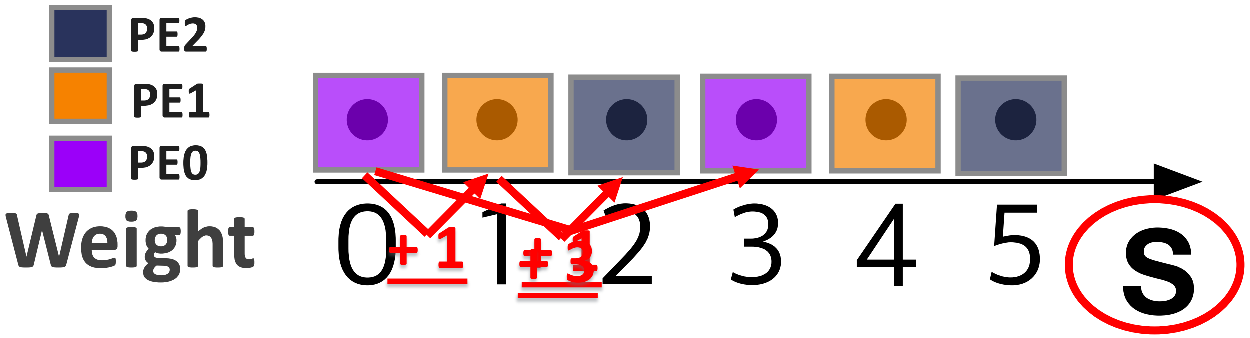
- **High-level Semantics:** Map the <u>same</u> data* across PEs, and update the mapping over time

* When the data dimension is not 1D, maps a dimension of data, not data points

Introducing Data-centric Directives

▪ Spatial Map

```
Syntax: SpatialMap(Mapping size, Spatial Offset) Dim
```



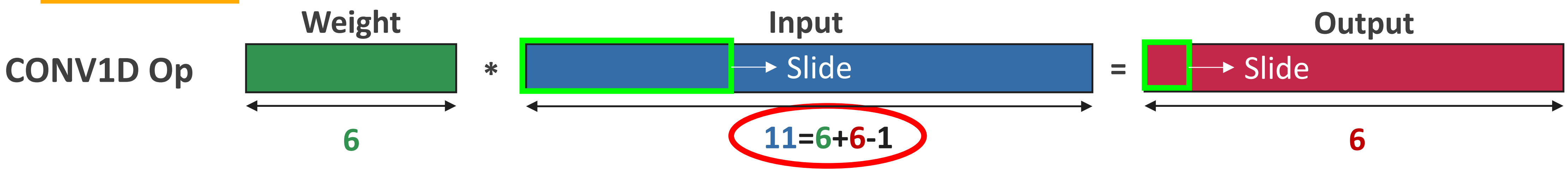
Description

SpatialMap(1, 1) *S*

- **High-level Semantics:** Map different data* across PEs with offset
  ➡ Parallelization!
- **Spatial Folding:** When the number of PEs is not sufficient to cover entire data
  - **Implicit temporal offset:** the number of PEs

* When the data dimension is not 1D, maps a dimension of data, not data points

Describing Dataflows using Data-centric Directives

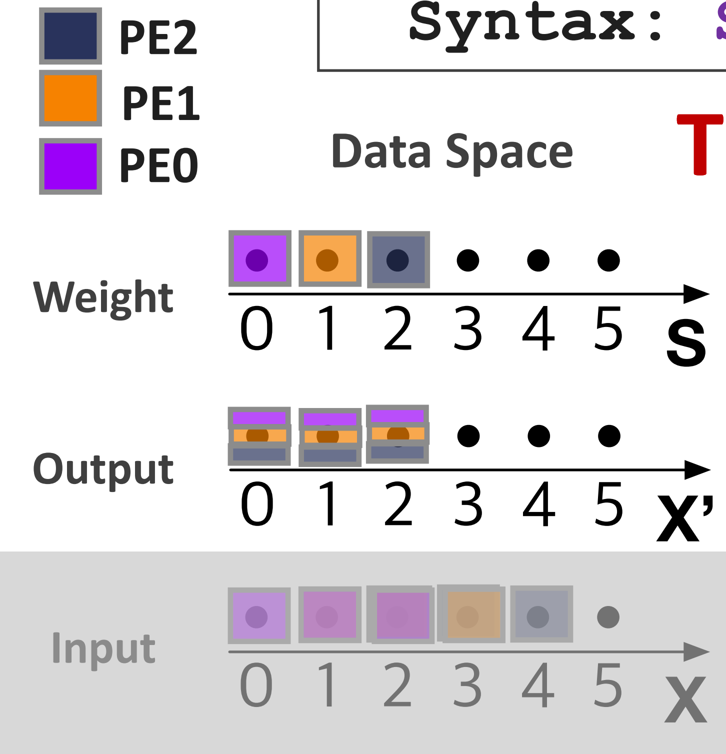



```
Syntax: Sp/TpMap(Mapping size, Sp/Tp Offset) Dim
```

Data Space

Time = 0

What changes faster?



SpatialMap(1,1) S

TemporalMap(1,1) X'

Description

Change slower

Change faster

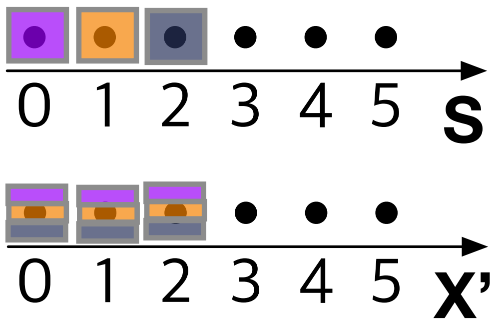
Directive order: relative order of "change" in each data dimension

The Impact of Directive Order

Syntax: Sp/TpMap(Mapping size, Sp/Tp Offset) Dim

PE2
PE1
PE0

Data Space

Weight
Output


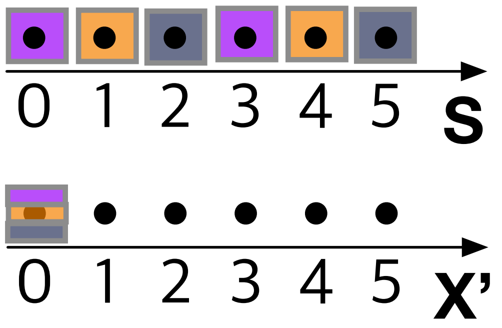
SpatialMap(1,1) S
TemporalMap(1,1) X’

“Weight stationary”

Tensor	Reuse Factor	Minimum Buffer Size
Weight	X’	1
Output	1	1

Change Directive Order

Data Space

Weight
Output


“Output stationary”

Tensor	Reuse Factor	Minimum Buffer Size
Weight	1	1
Output	S/3	1

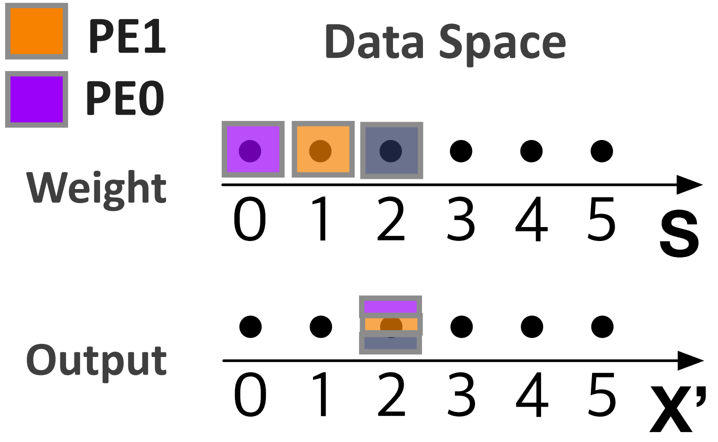
* Reuse factor: The number of acc

Weight Reuse to Output Reuse

The Impact of Spatial/Temporal Directive Choice

Syntax: Sp/TpMap(Mapping size, Sp/Tp Offset) Dim

PE2
PE1
PE0

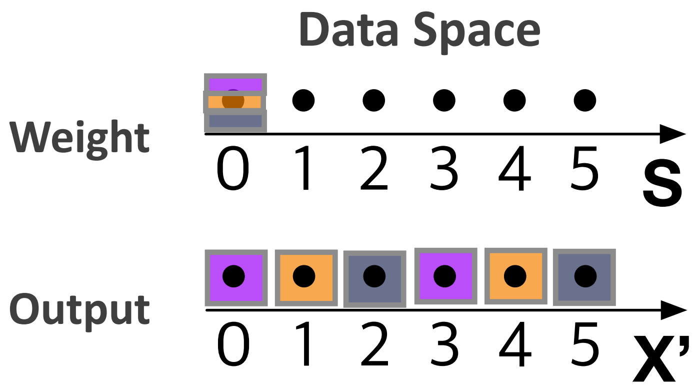


SpatialMap(1,1) S
TemporalMap(1,1) X'

"Weight stationary"

Tensor	Reuse Factor	Minimum Buffer Size
Weight	X'	1
Output	1	1

Change
Spatial/Temporal



TemporalMap(1,1) S
SpatialMap(1,1) X'

"Weight stationary"

Tensor	Reuse Factor	Minimum Buffer Size
Weight	X'/3	1
Output	1	1

* Reuse factor: The number o

Different Weight-Stationary!

The Impact of Mapping Size

Syntax: Sp/TpMap(Mapping size, Sp/Tp Offset) Dim

PE2
PE1
PE0

Data Space

Weight

Output

Data Space

Even in a simple CONV1D, we observe complex trade-off space based on dataflow

Directives can precisely describe dataflows

Mapping Size

"Weight stationary"

Tensor	Reuse Factor	Minimum Buffer Size
Weight	X'	1
Output	1	1

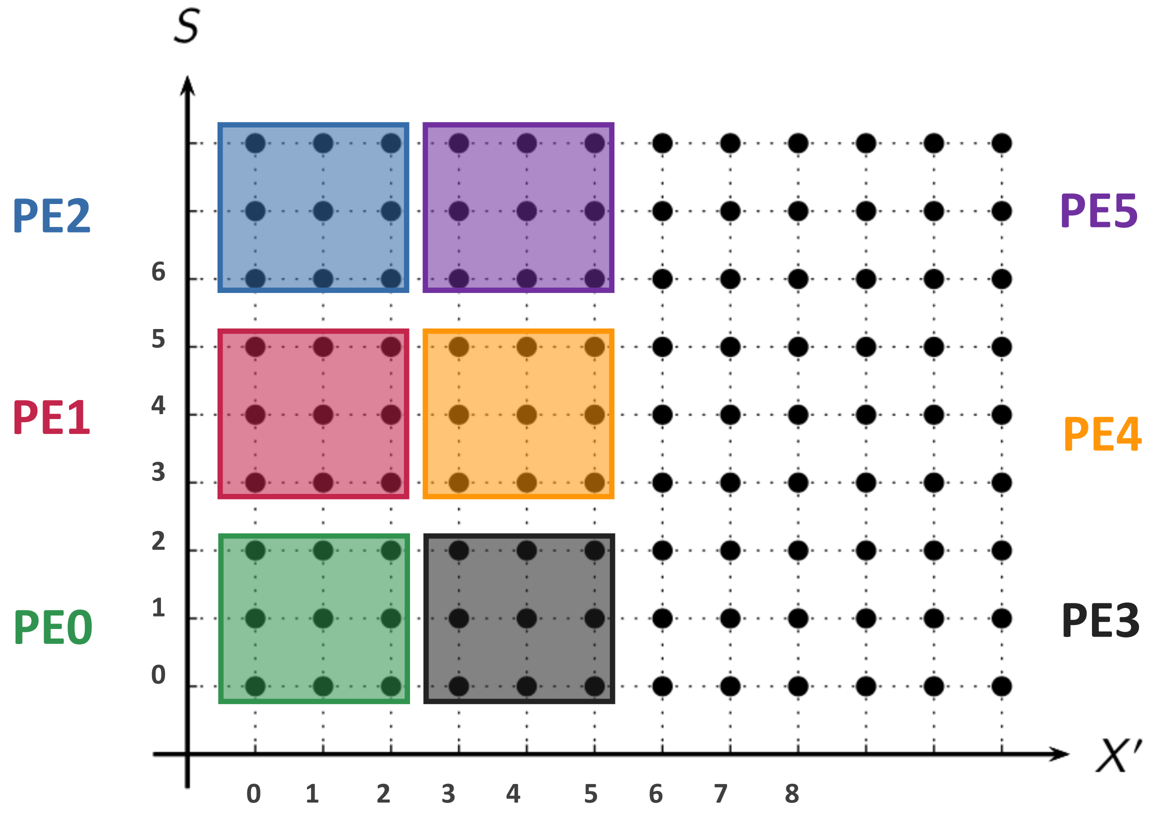
"Weight stationary"

Tensor	Reuse Factor	Minimum Buffer Size
Weight	X'/3	1
Output	1	3

* Reuse factor: The number o

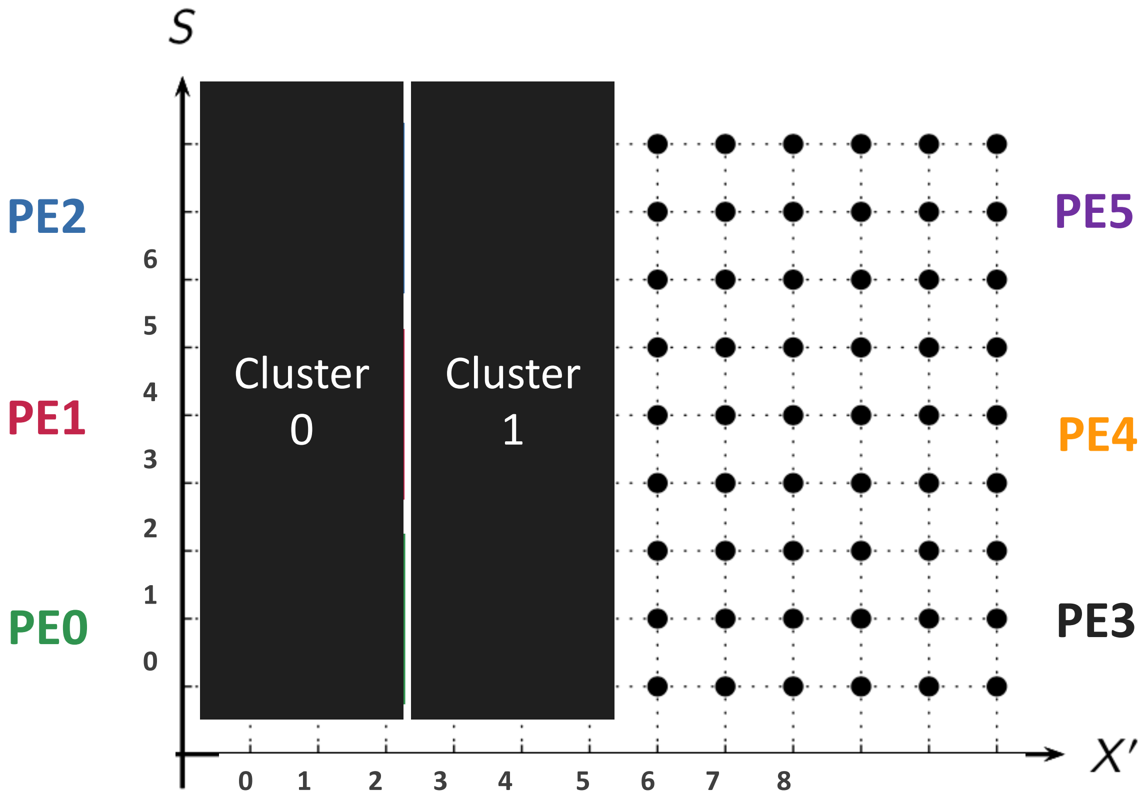
Another Weight-Stationary!

Describing Mappings with Multiple Parallel Dimensions



How to describe this dataflow?

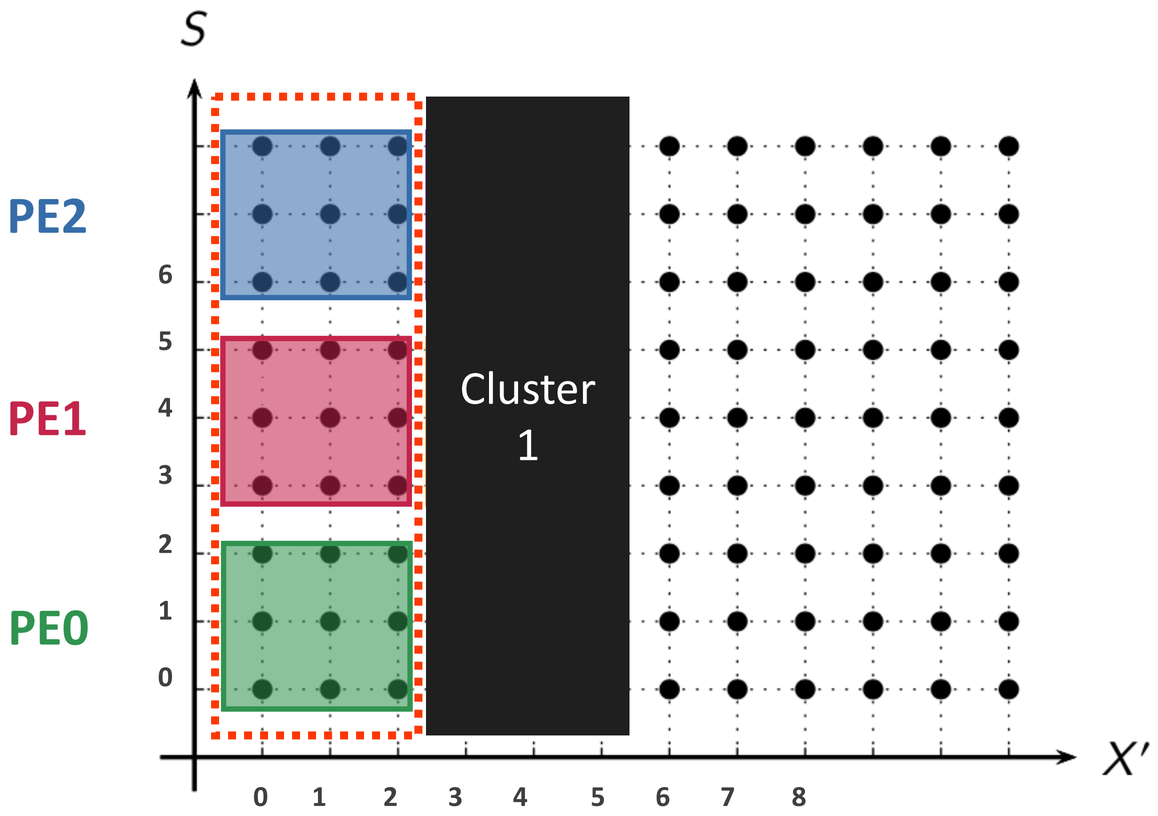
Multi-level Parallelism via Clustering



TemporalMap(size=9, offset=9) *S*
SpatialMap(size=3, offset=3) *X'*

Mapping target: Clusters!

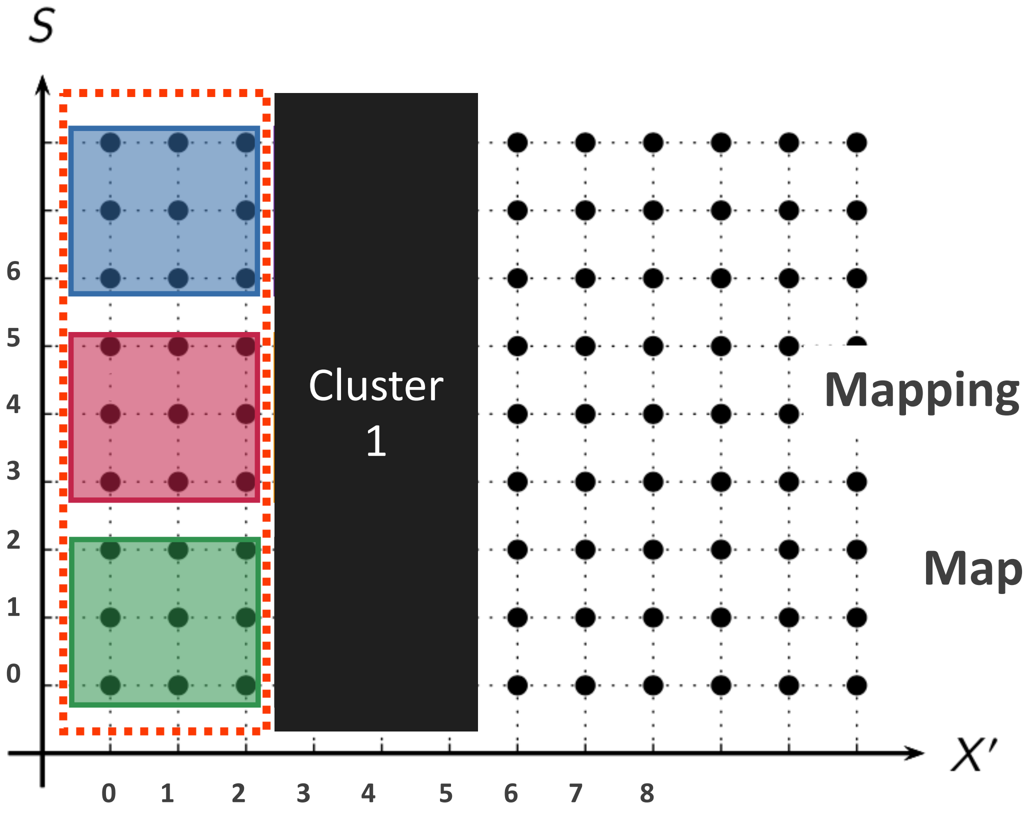
Multi-level Parallelism via Clustering



SpatialMap(size=3, offset=3) *S*
TemporalMap(size=3, offset=3) *X'*

Mapping target: PEs!

Multi-level Parallelism via Clustering



Mapping target: Clusters
- Temporal_Map(size=9, offset=9) *S*
- Spatial_Map(size=3, offset=3) *X'*

Cluster (size=3)

Mapping target: PEs
- Spatial_Map(size=3, offset=3) *S*
- Temporal_Map(size=3, offset=3) *X'*

Outline

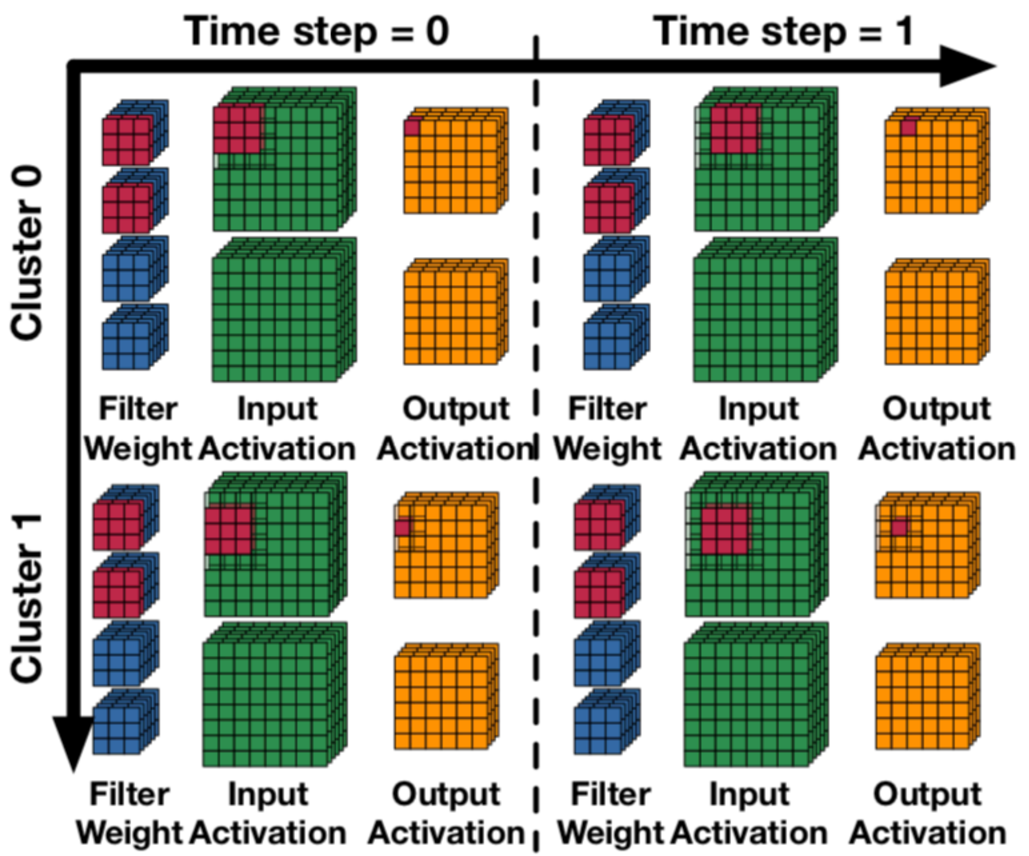
- **Mapping Representation: A data-centric representation**
  - **Computation and Data Space**
  - **Data-centric Directives**
  - **Deep-dive Example: Eyeriss-like Dataflow**
- **MAESTRO Cost Model – High Level Overview**

Full CONV2D Mapping Overview (Eyeriss-like)

TemporalMap(1,1) N
TemporalMap(2,2) K
TemporalMap(3,3) C
SpatialMap(3,1) Y
TemporalMap(3,1) X
Cluster(3, L)
SpatialMap(1,1) Y
* SpatialMap(1,1) R
* TemporalMap(3,3) S

* Dimension Fully Covered by One Iteration

(b) Example dataflow



(c) Visualized data mapping over time

Eyeriss-like Mapping

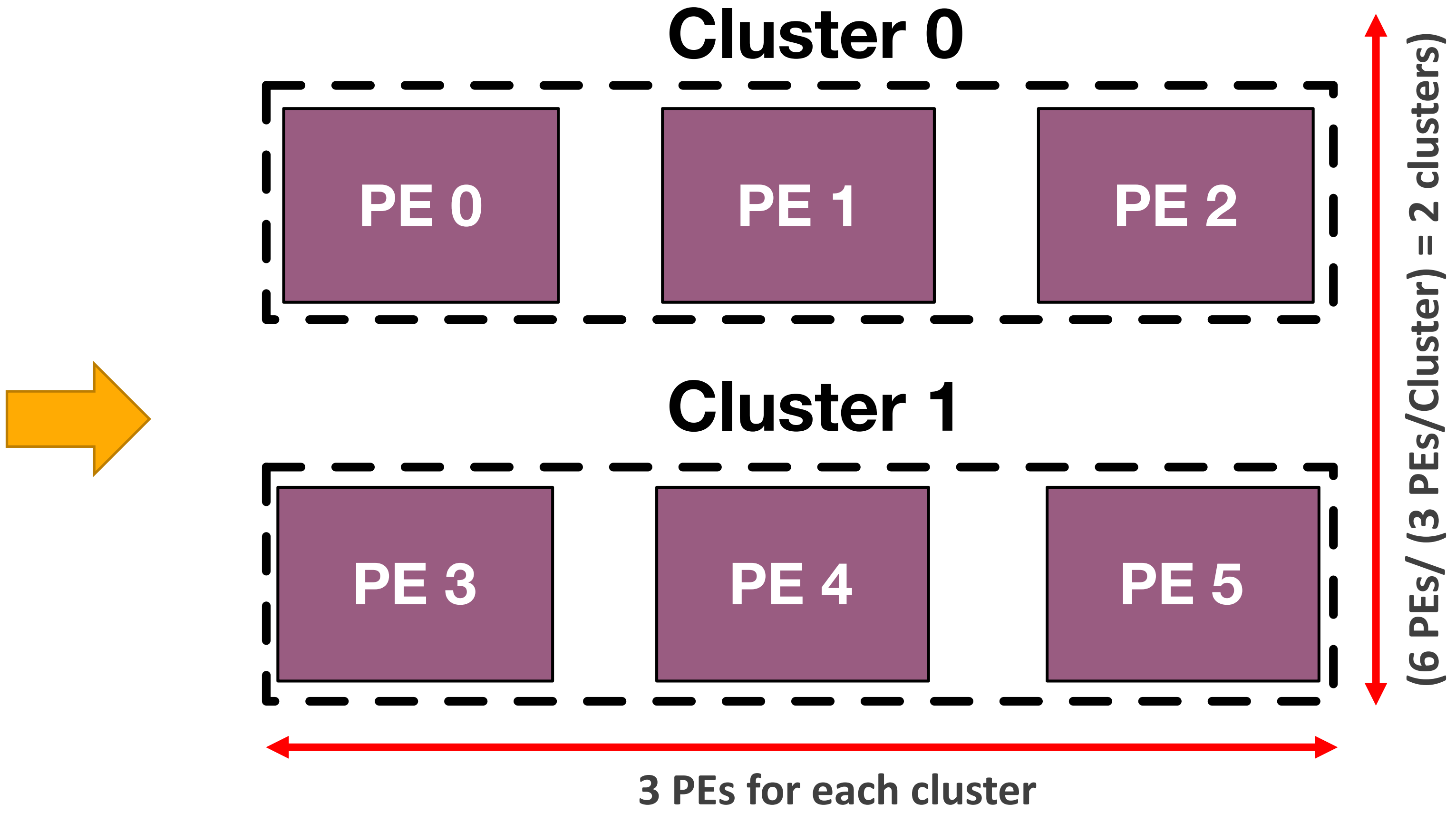
Free Variables

TemporalMap(TileSz(K),TileSz(K)) **K**
TemporalMap(TileSz(C),TileSz(C)) **C**
SpatialMap(Sz(R),1) **Y**
TemporalMap(Sz(S),1) **X**
Cluster(Sz(R)) 2D PE array
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(Sz(S),Sz(S)) **S**

Eyeriss Hardware Implied by Mapping

- Will Assume 3x3 filter and 6 PEs in total

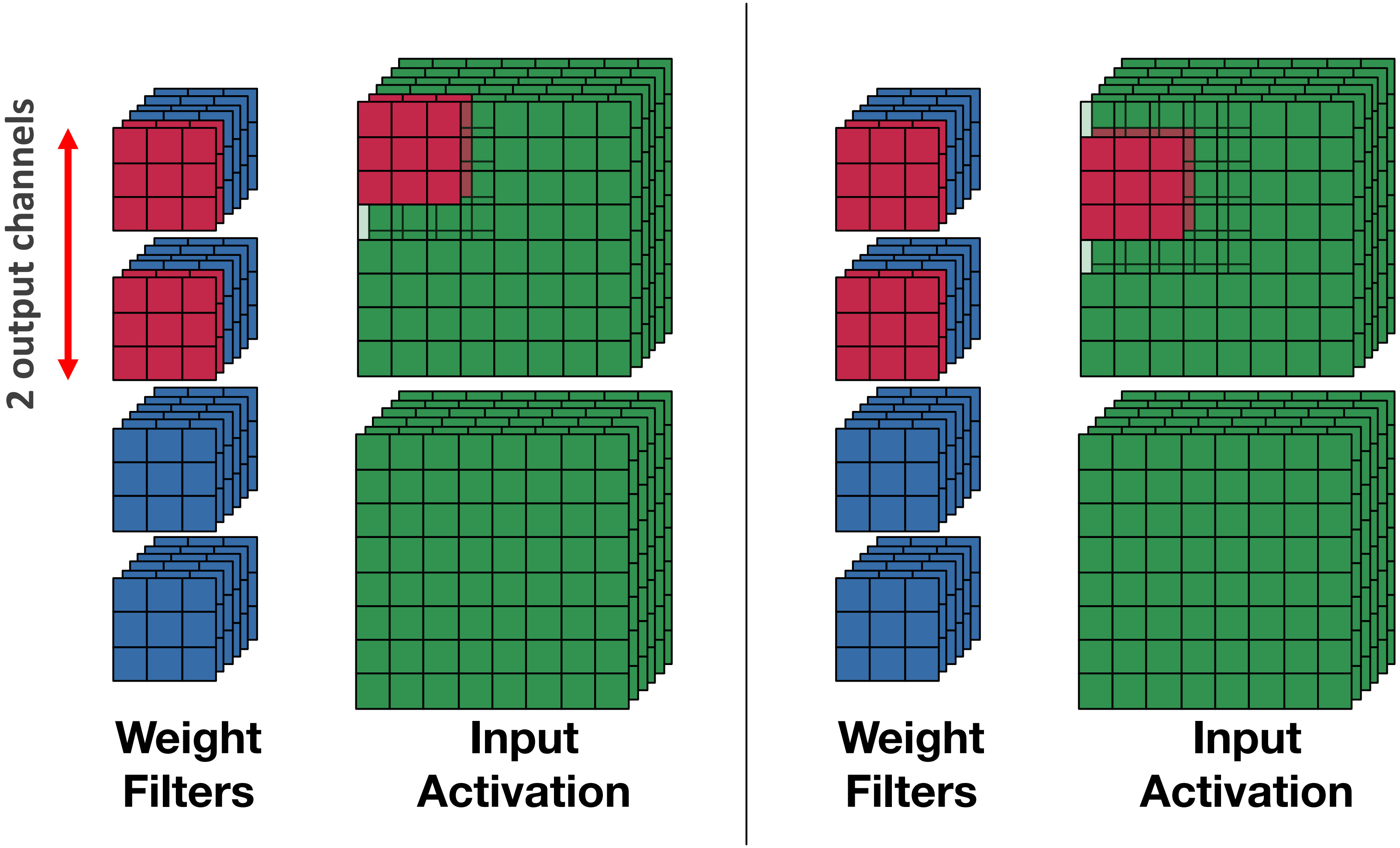
TemporalMap(2,2) **K**
TemporalMap(2,2) **C**
SpatialMap(3,1) **Y**
TemporalMap(3,1) **X**
Cluster(3)
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(3,3) **S**



Tile Size / Offset Analysis

TemporalMap (Map size, Offset) *Dim*
SpatialMap (Map size, Offset) *Dim*

TemporalMap(2,2) **K**
TemporalMap(2,2) **C**
SpatialMap(3,1) **Y**
TemporalMap(3,1) **X**
Cluster(3)
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(3,3) **S**



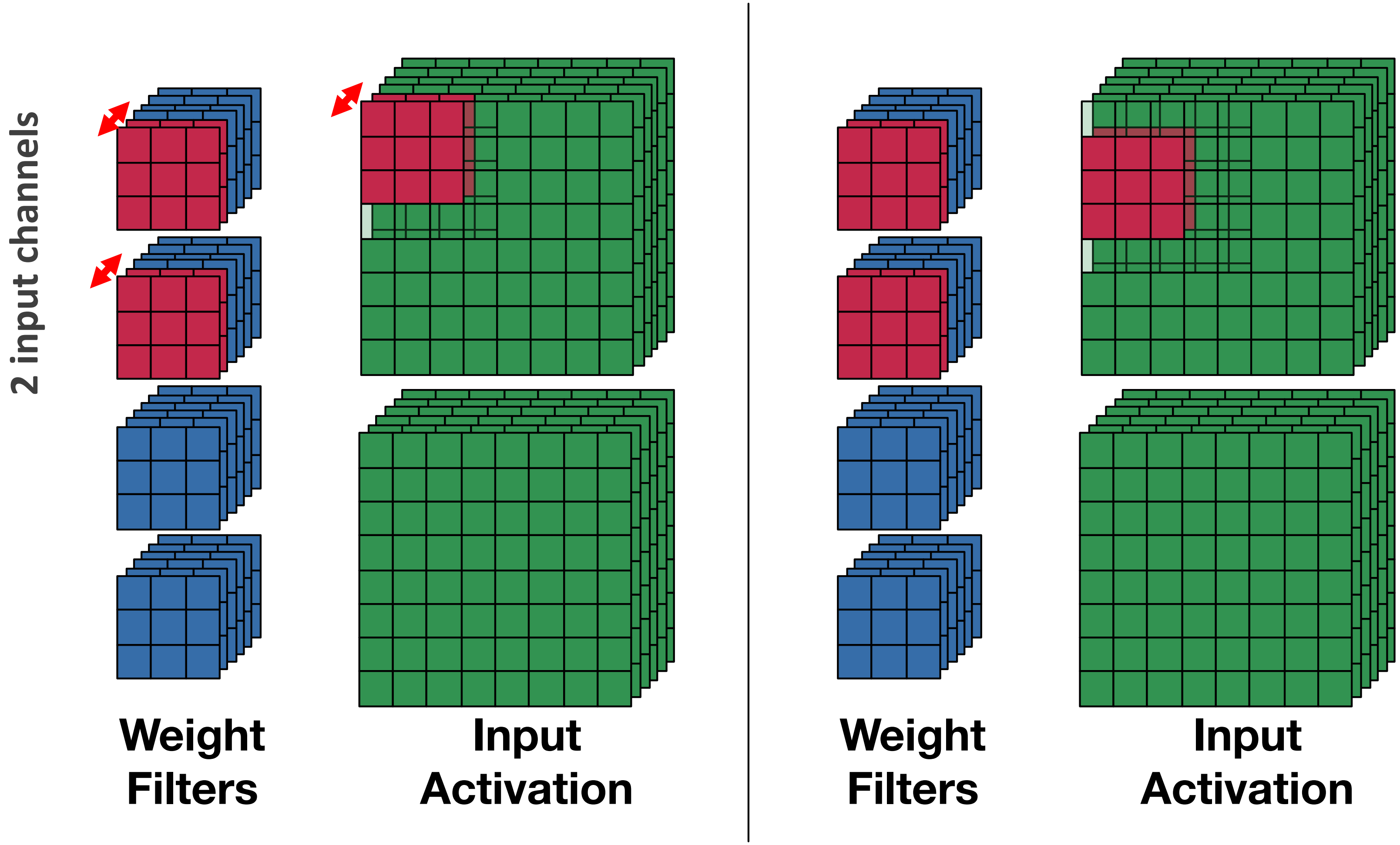
Mapping over cluster 0
<Time Step 0>

Mapping over cluster 1
<Time Step 0>

Tile Size / Offset Analysis

TemporalMap (Map size, Offset) *Dim*
SpatialMap (Map size, Offset) *Dim*

TemporalMap(2,2) **K**
TemporalMap($\boxed{2}$,2) **C**
SpatialMap(3,1) **Y**
TemporalMap(3,1) **X**
Cluster(3)
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(3,3) **S**



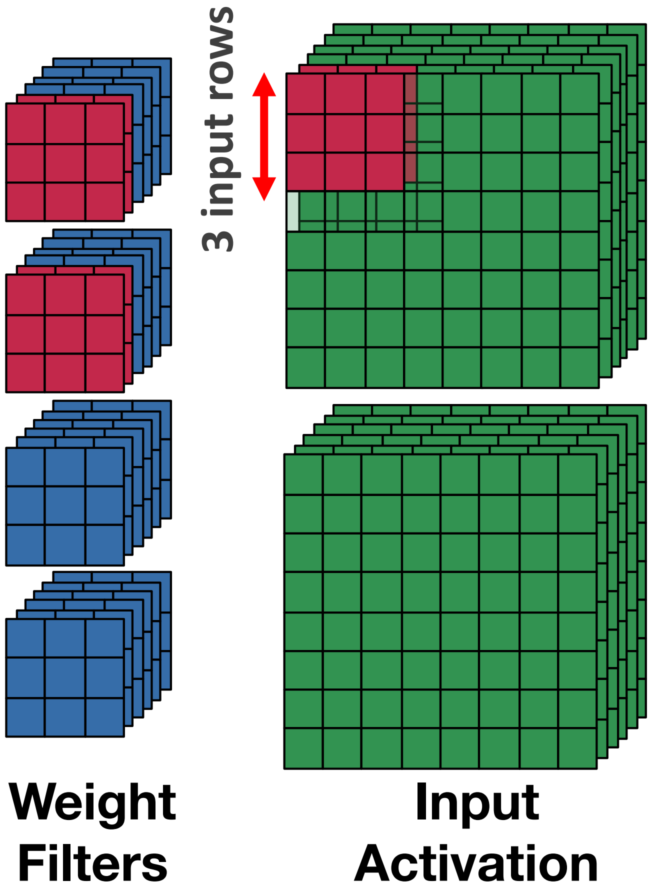
Mapping over cluster 0
<Time Step 0>

Mapping over cluster 1
<Time Step 0>

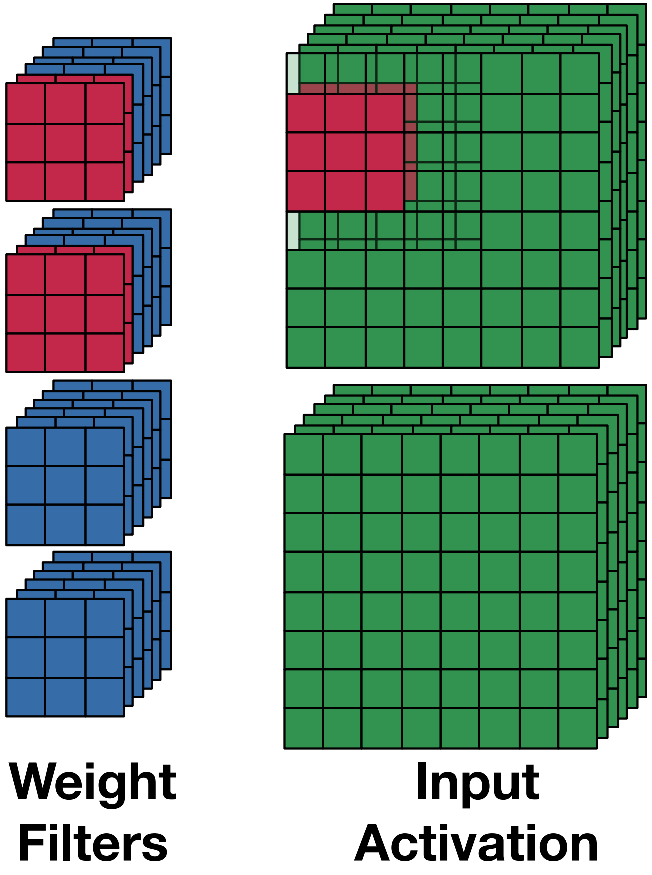
Tile Size / Offset Analysis

TemporalMap (Map size, Offset) *Dim*
SpatialMap (Map size, Offset) *Dim*

TemporalMap(2,2) **K**
TemporalMap(2,2) **C**
SpatialMap($\boxed{3}$,1) **Y**
TemporalMap(3,1) **X**
Cluster(3)
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(3,3) **S**



Mapping over cluster 0
<Time Step 0>

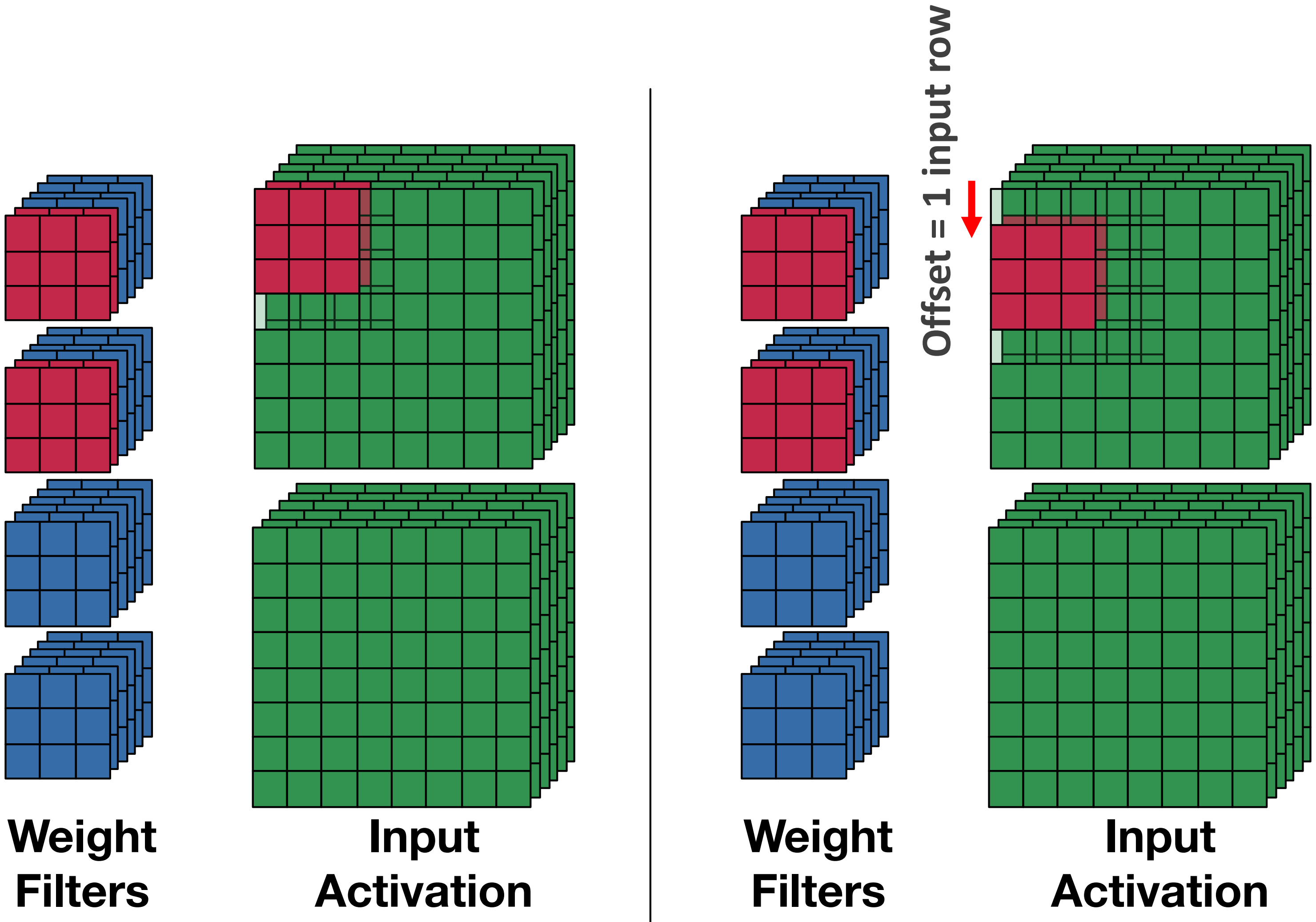


Mapping over cluster 1
<Time Step 0>

Tile Size / Offset Analysis

TemporalMap (Map size, Offset) *Dim*
SpatialMap (Map size, Offset) *Dim*

TemporalMap(2,2) **K**
TemporalMap(2,2) **C**
SpatialMap(3,$\boxed{1}$) **Y**
TemporalMap(3,1) **X**
Cluster(3)
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(3,3) **S**



Mapping over cluster 0
<Time Step 0>

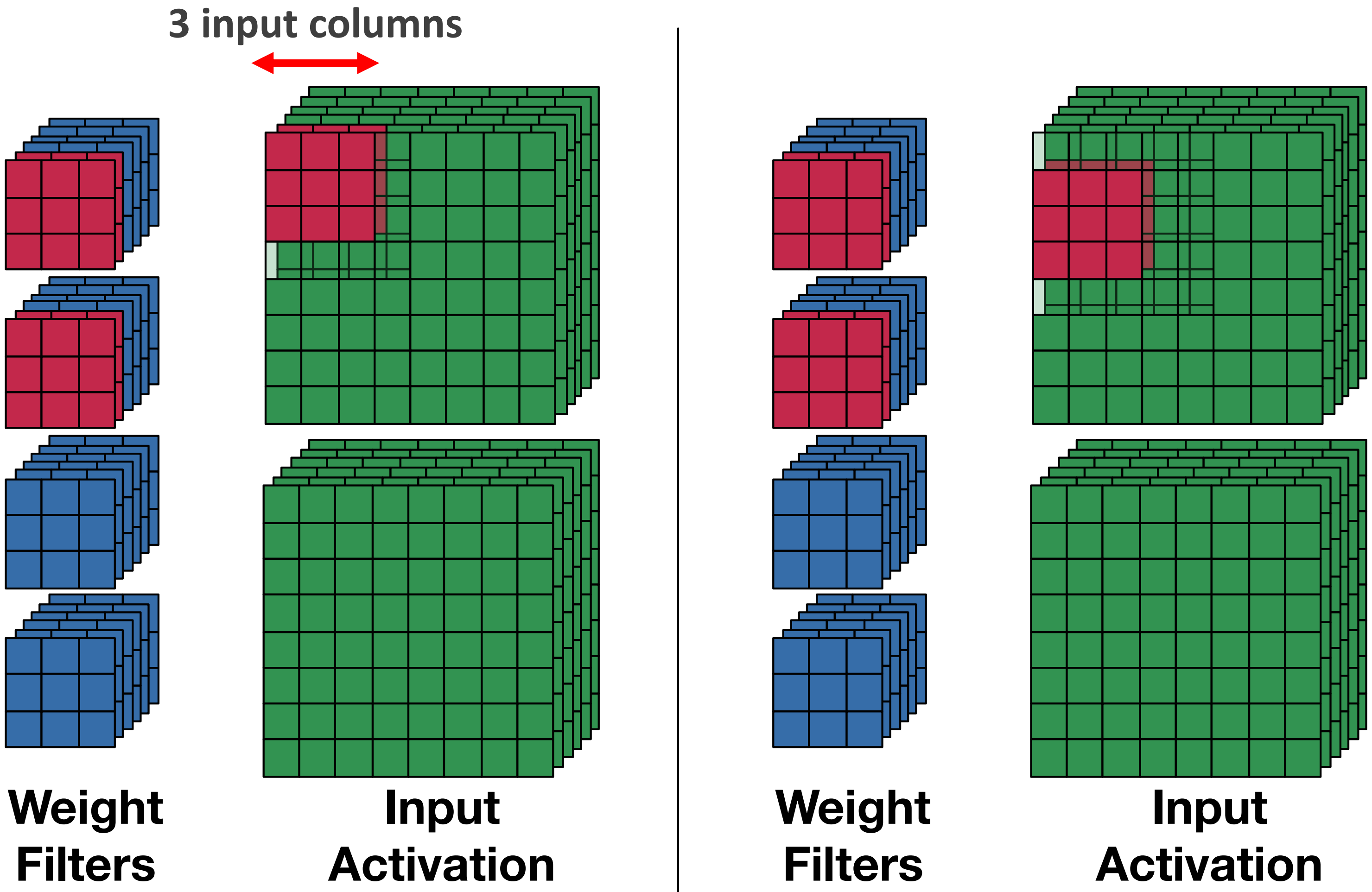
Mapping over cluster 1
<Time Step 0>

Tile Size / Offset Analysis

TemporalMap (Map size, Offset) *Dim*
SpatialMap (Map size, Offset) *Dim*

- **Tile Size Analysis**

TemporalMap(2,2) **K**
TemporalMap(2,2) **C**
SpatialMap(3,1) **Y**
TemporalMap($\boxed{3}$,1) **X**
Cluster(3)
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(3,3) **S**



Mapping over cluster 0
<Time Step 0>

Mapping over cluster 1
<Time Step 0>

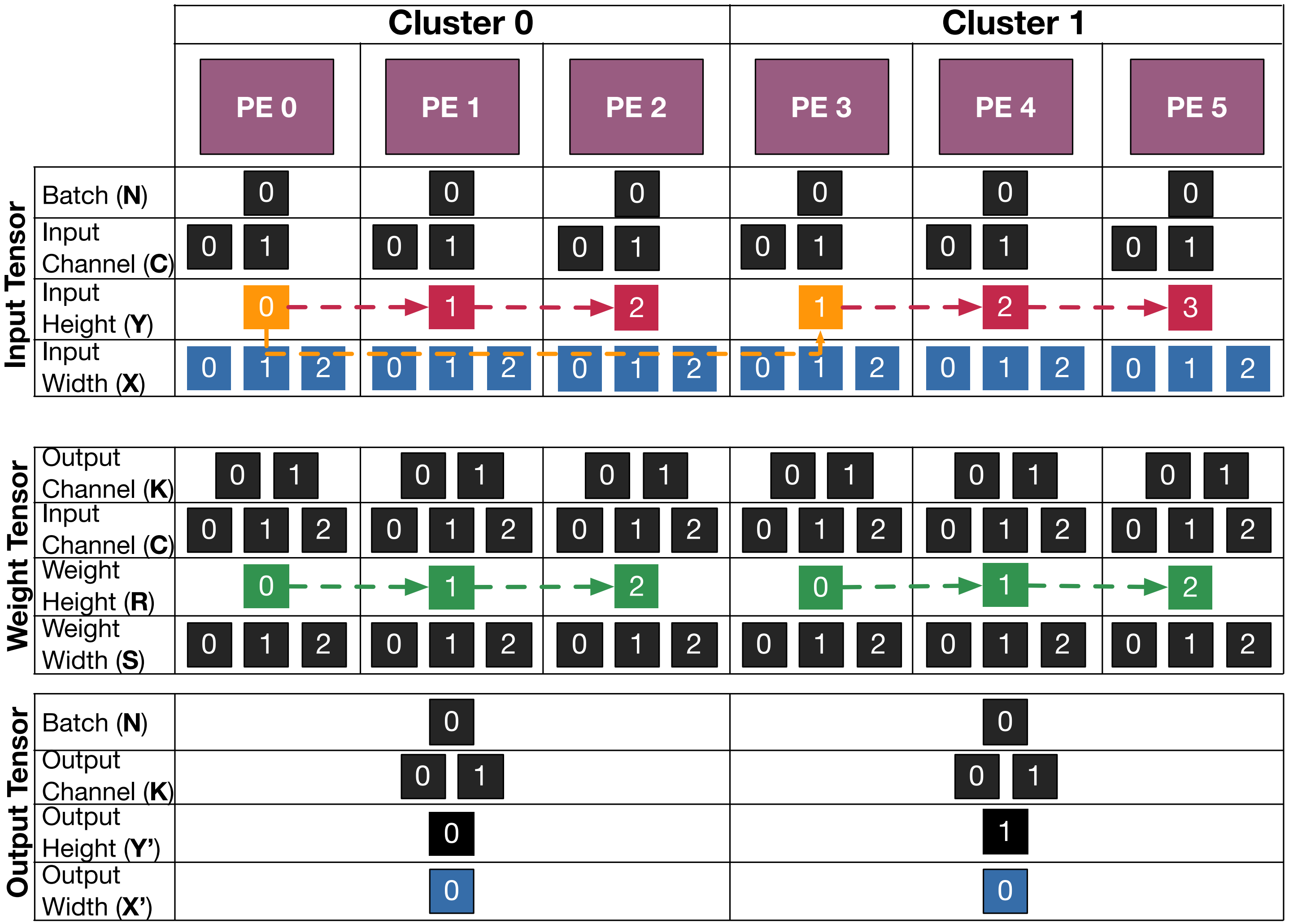
Data Reuse Analysis

TemporalMap (Map size, Offset) *Dim*
SpatialMap (Map size, Offset) *Dim*

TemporalMap(2,2) **K**
TemporalMap(2,2) **C**
SpatialMap(3,1) **Y**
TemporalMap(3,1) **X**
Cluster(3)
SpatialMap(1,1) **Y**
SpatialMap(1,1) **R**
TemporalMap(3,3) **S**

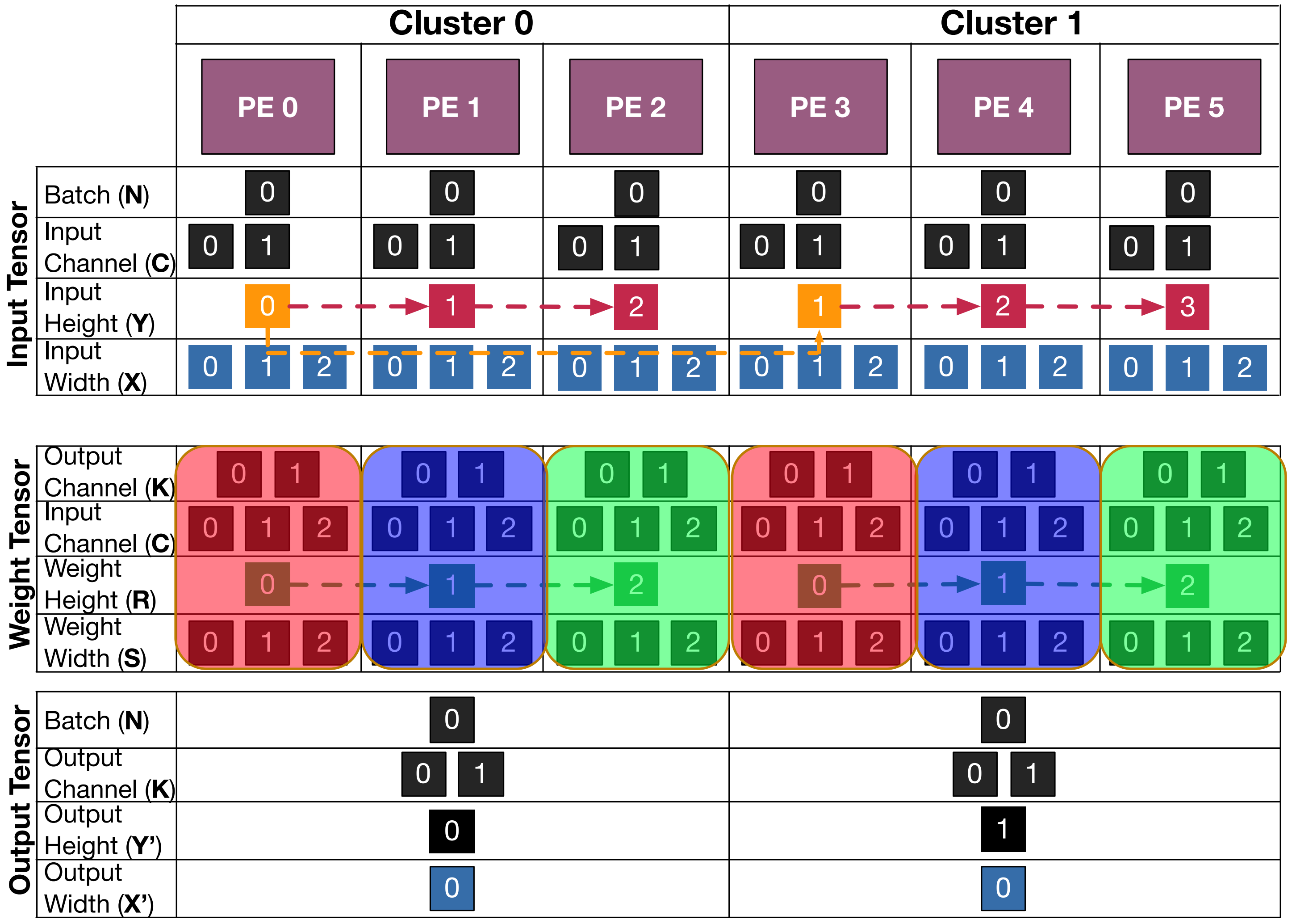
Variable / Data class	Output Channel (K)	Input Channel (C)	Filter Row (R)	Filter Column (S)	Input Row (Y)	Input Column (X)
Output Activation	X		X	X	X	X
Input Activation		X			X	X
Filter Weights	X	X	X	X		

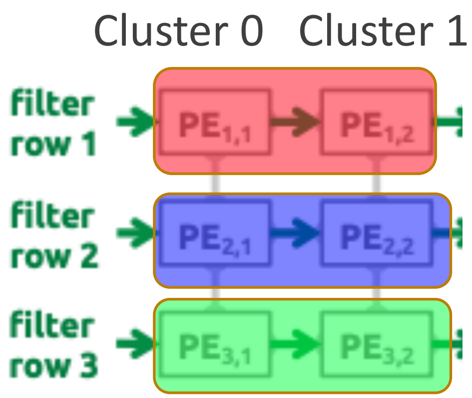
* Output row(Y') = Y-R+1, Output column(X') = X-S+1



<Time Step 0>

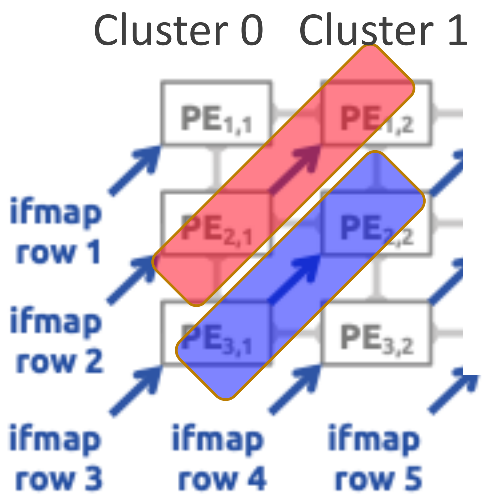
Data Reuse Analysis





Data Reuse Analysis

		Cluster 0			Cluster 1		
		PE 0	PE 1	PE 2	PE 3	PE 4	PE 5
Input Tensor	Batch (**N**)	0	0	0	0	0	0
	Input Channel (**C**)	0 1	0 1	0 1	0 1	0 1	0 1
	Input Height (**Y**)	0	1	2	1	2	3
	Input Width (**X**)	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2
Weight Tensor	Output Channel (**K**)	0 1	0 1	0 1	0 1	0 1	0 1
	Input Channel (**C**)	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2
	Weight Height (**R**)	0	1	2	0	1	2
	Weight Width (**S**)	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2	0 1 2
Output Tensor	Batch (**N**)	0			0		
	Output Channel (**K**)	0 1			0 1		
	Output Height (**Y'**)	0			1		
	Output Width (**X'**)	0			0		



Data Reuse Analysis