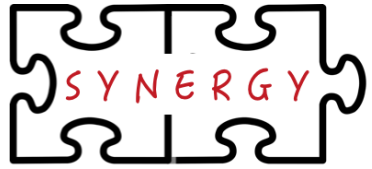


A High-level Overview of MAESTRO Mapping Directives and Cost Model

Synergy Lab, Georgia Tech
Hyoukjun Kwon

Outline

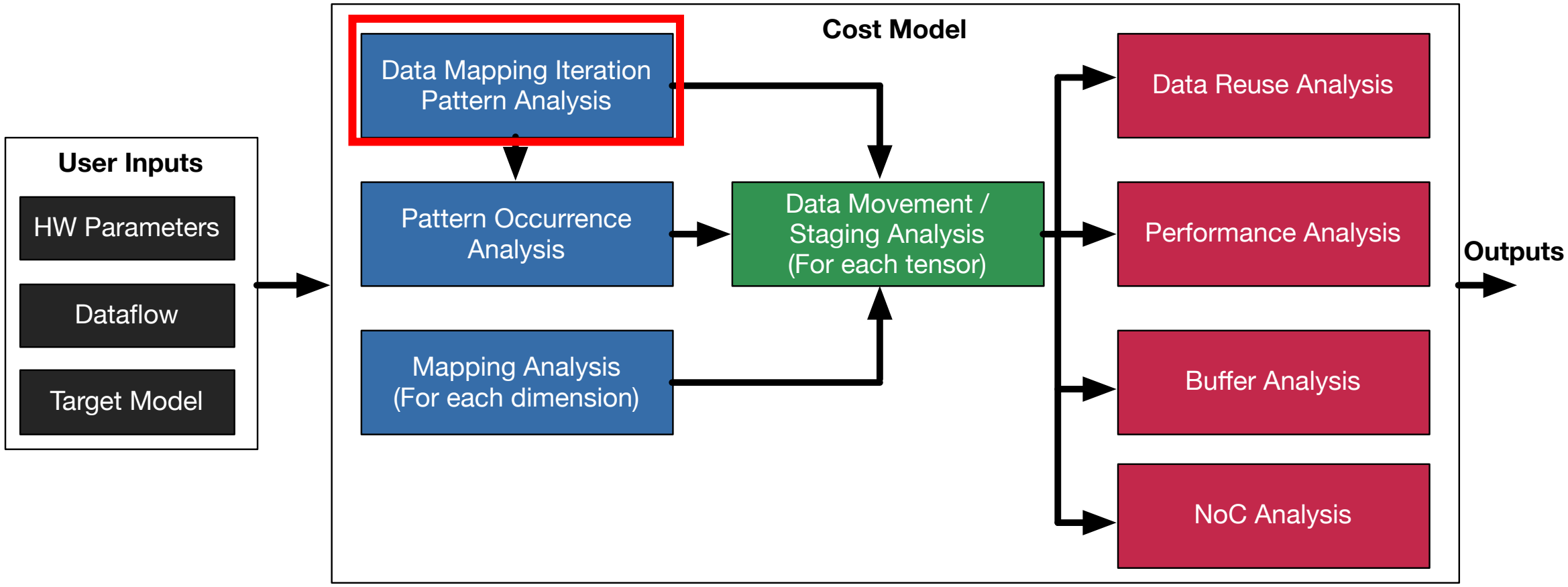
- Mapping Representation: A data-centric representation
  - Computation and Data Space
  - Data-centric Directives
  - Deep-dive Example: Eyeriss-like Dataflow
- **MAESTRO Cost Model – High Level Overview**

Source Code Structure

Cost-model/include/

- base: Base class definitions
- tools: Misc. helper classes (output file generation, etc.)
- user-api: API class (code level APIs)
- dataflow-specification-language: Directive syntax definition, parser, etc.
- abstract-hardware-model: Hardware performance and cost models
- dataflow-analysis: Data reuse analysis
- cost-analysis: Compute the costs using data reuse analysis results
- design-space-exploration: (Will be renamed) Contains base HW cost information

Cost Model Overview



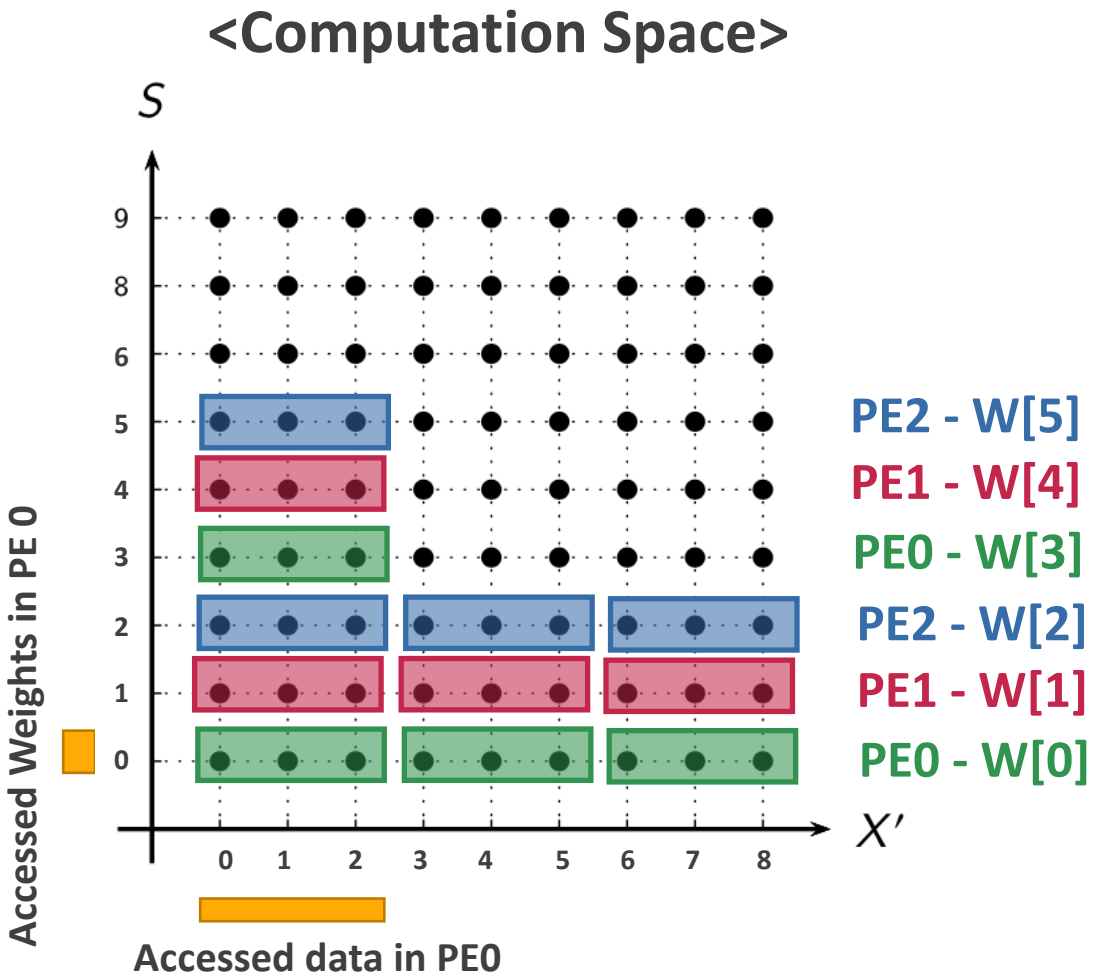
Mapping Iteration Pattern Analysis

SpatialMap(Sz=1, Ofs=1) S
TemporalMap(Sz=3, Ofs=3) X
Cluster(Sz=3)
TemporalMap(Sz=1, Ofs=1) S
SpatialMap(Sz=1, Ofs=1) X

Filter: Init
Input: Init

Filter: Init
Input: Steady

Filter: Steady
Input: Init

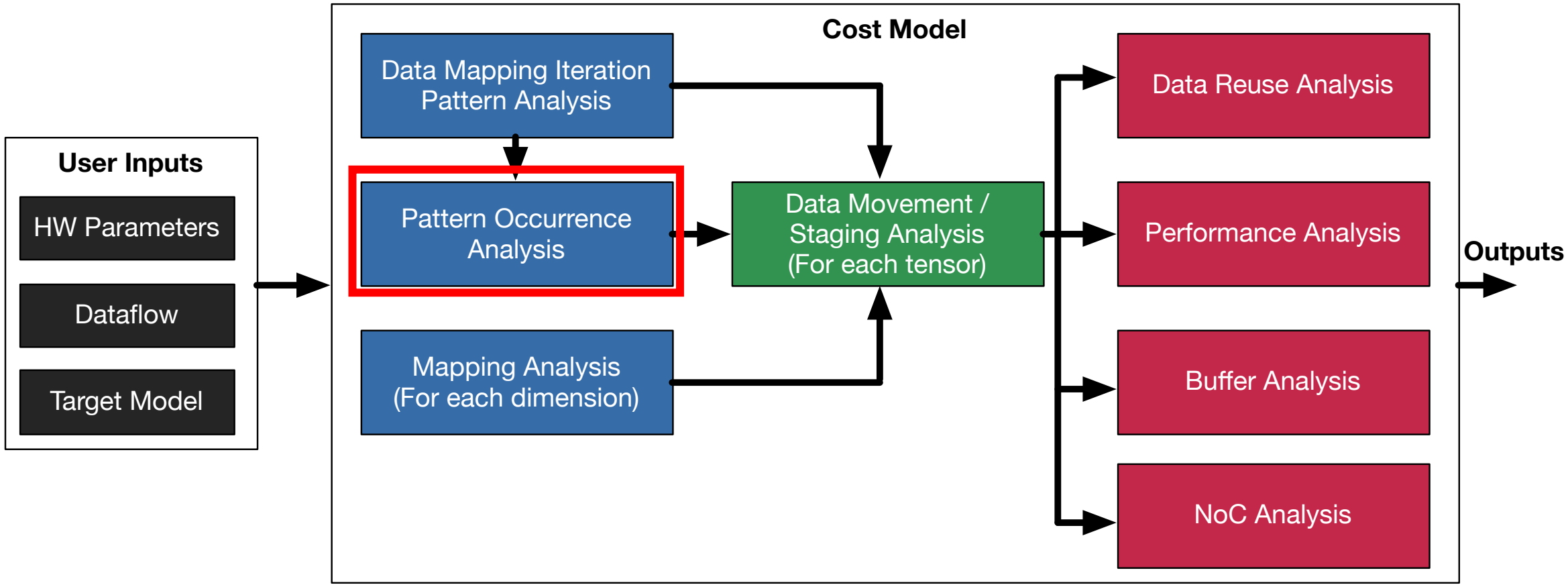


Mapping iteration Status = {Init, Steady, Edge}

Analysis is performed on each dimension in higher dimension cases

Identifies which tensor changes for each iteration pattern

Cost Model Overview

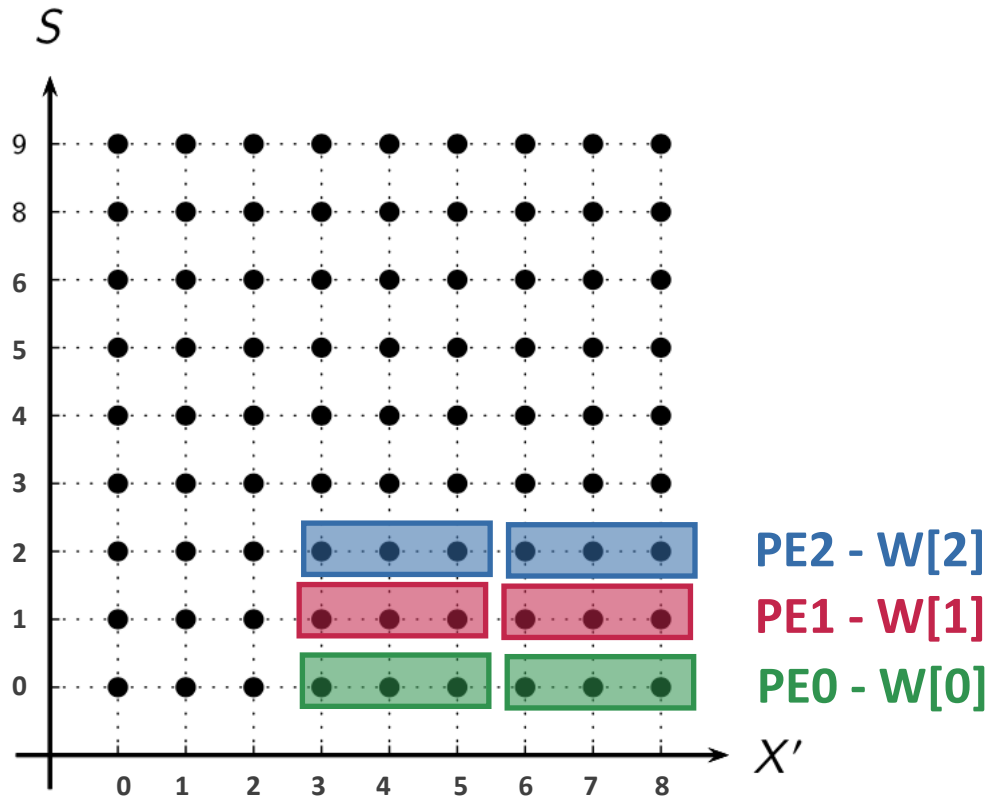


Mapping Iteration Pattern Count Analysis

SpatialMap(Sz=1, Ofs=1) S
TemporalMap(Sz=3, Ofs=3) X
Cluster(Sz=3)
TemporalMap(Sz=1, Ofs=1) S
SpatialMap(Sz=1, Ofs=1) X

Filter: Init
Input: Steady

<Computation Space>



Input

Dimension(X') / MapSz(X') = 3

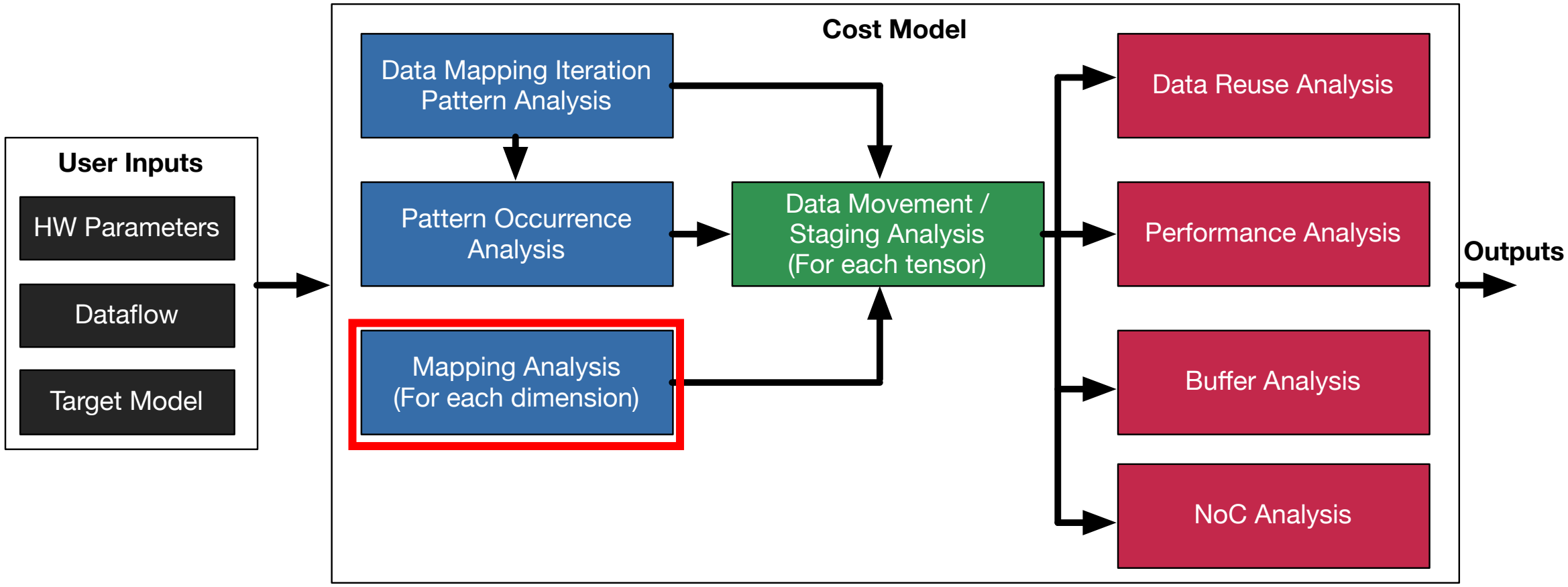
Subtract init case; 3-1 =2

Filter

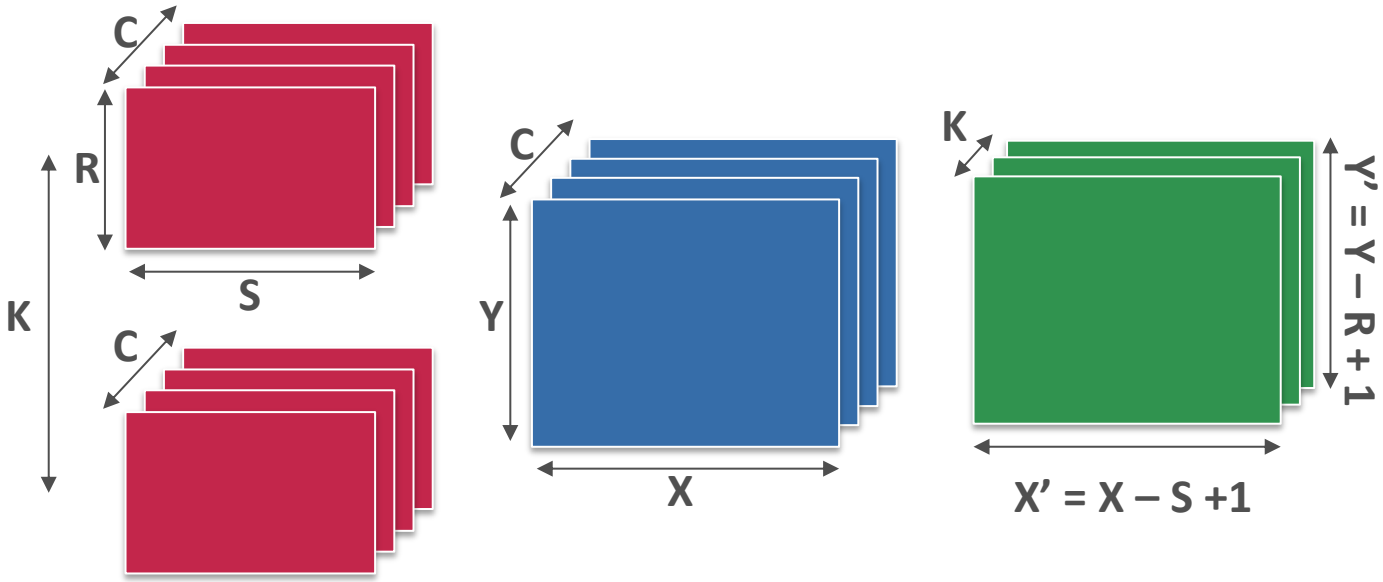
Init case = 1

2x1 = twice

Cost Model Overview



Mapping Analysis - Convention



Fliters Input Activations Output Activations

Data Dimensions (Loop Variables)

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

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

Mapping Analysis

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

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

How many *weight pixels* do we map on each PE?

2 x 2 x 3 x 3 = 36 pixels

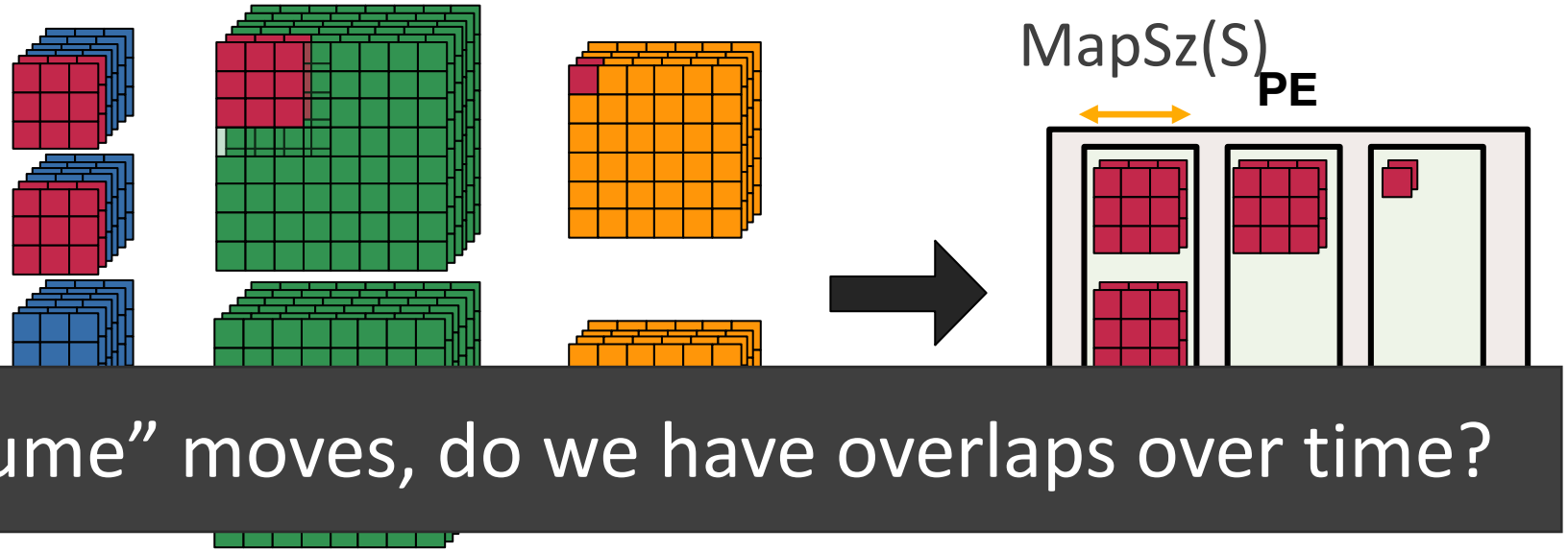
Syntax

TemporalMap (MapSz, Ofs) *Var*
SpatialMap (MapSz, Ofs) *Var*

Mapping Analysis

TemporalMap (MapSz, Ofs) *Var*
SpatialMap (MapSz, Ofs) *Var*

TemporalMap (*1*, 1) **N**
TemporalMap (2, 2) **K**
TemporalMap (*2*, 2) **C**
TemporalMap (3, 3) **R**
TemporalMap (3, *1*) **Y**



When a mapped “Volume” moves, do we have overlaps over time?

How many? (How much data reuse do we have?)

MV(Weight) = MapSz(K) x MapSz(C) x MapSz(R) x MapSz(S)

How many over time? And, how many among PEs at the same time?

MV: Mapped Volume

Extending to 6D case

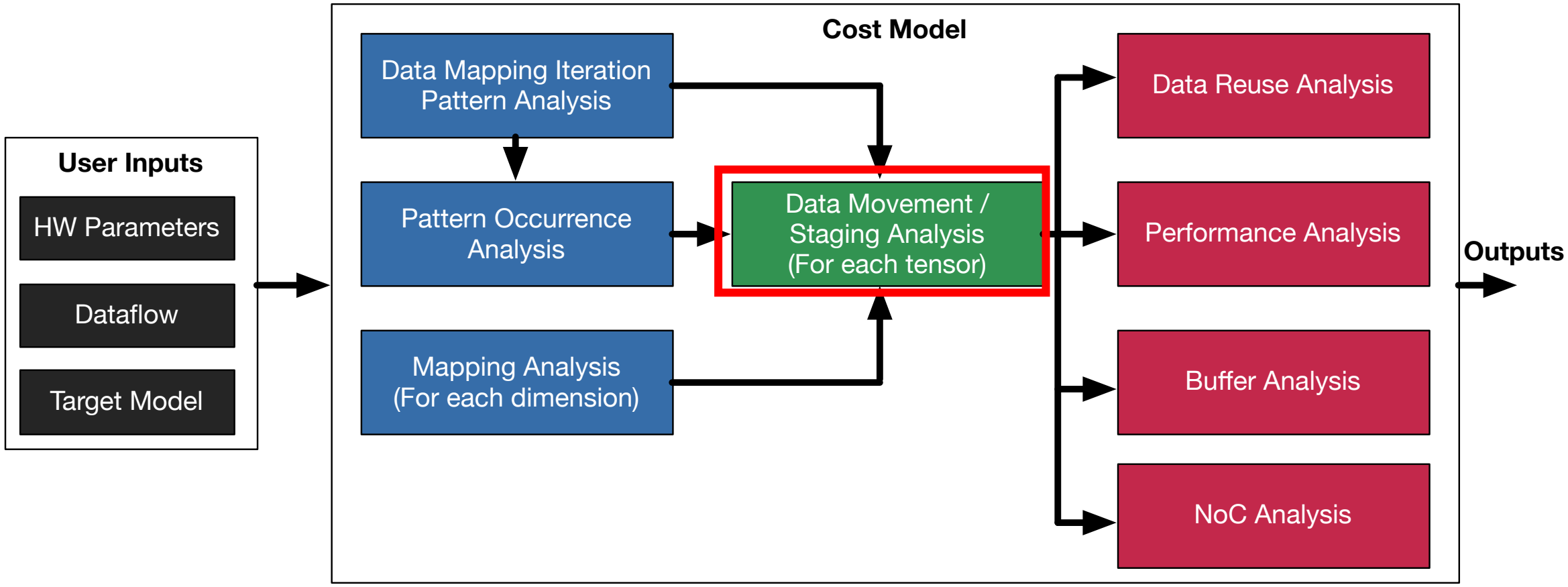
Terms

TU: Temporally unique values
SU: Spatially unique values
SPUSz: Spatially unique values
TUV: Temporally unique volume
SUV: Spatially unique volume

```
//MV: Mapped volume
MV[Weights]  = M(K) x M(C) x M(R) x M(S)
MV[Inputs]   = M(C) x M(Y) x M(X)
MV[Outputs]  = M(K) x M(Y') x M(X')

//MSUV: Mapped spatially unique volume
MSUV[Weights]  = GetSpUSz(K) x GetSpUSz(C) x GetSpU
MSUV[Inputs]   = GetSpUSz(C) x GetSpUSz(Y) x GetSpUSz(X)
MSUV[Outputs]  = GetSpUSz(K) x GetSpUSz(C) x GetSpUSz(Y') x GetSpUSz(X')

//MTUV: Mapped temporally unique volume
MTUV[Weights]  = TU(K) x TU(C) x TU(R) x TU(S)
M
M

//M
MSTUV[Weights] = GetSTpUSz(K) x GetSTpUSz(C) x GetSTpUSz(R) x GetSTpUSz(S)
MSTUV[Inputs]  = GetSTpUSz(C) x GetSTpUSz(Y) x GetSTpUSz(X)
MSTUV[Outputs] = GetSTpUSz(K) x GetSTpUSz(C) x GetSTpUSz(Y') x GetSTpUSz(X')

* GetSpUSz(V) = (V.pragma.class == TemporalMap)? M(V) : SU(V);
* GetSTpUSz(V) = (V.pragma.class == SpatialMap)? SU(V) : TU(V);
```

Analyze the number of unique/reused pixels in each data class for each mapping iteration pattern

Cost Model Overview



Intuition for Rest of Cost Model

- **Iteration pattern analysis provides information regarding**
  - Which tensor changes in between two mappings
  - How many times is each case repeated
- **For details, please see the source code and web page (http://maestro.ece.gatech.edu)**
  - Overall, how many data points are mapped over each PE
  - If a tensor changes, how many data points are reused

Combining information we can extract..

1) Amount of data to be transferred from global buffer to PE array
2) Amount of computation to be done in each mapping

…