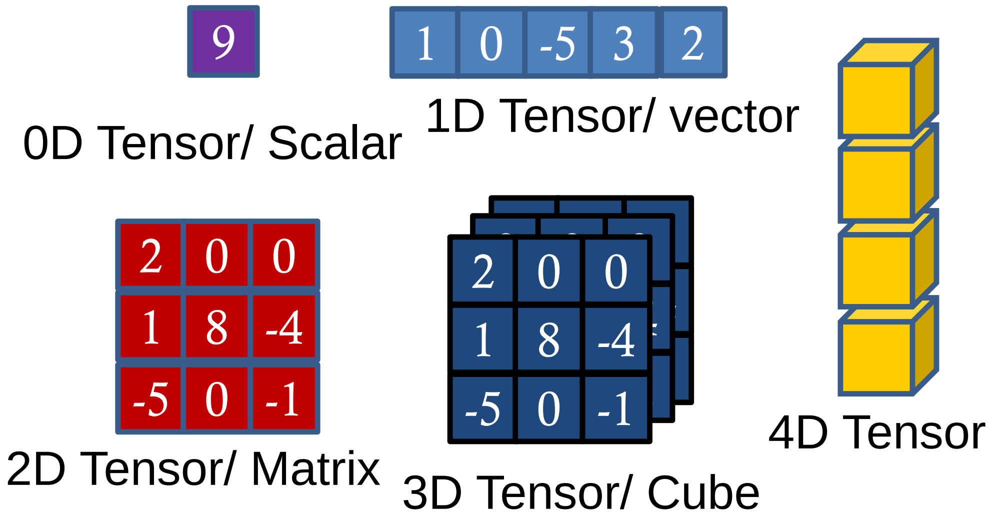
Tensaurus: A Versatile Accelerator for Mixed Sparse-Dense Tensor Computations

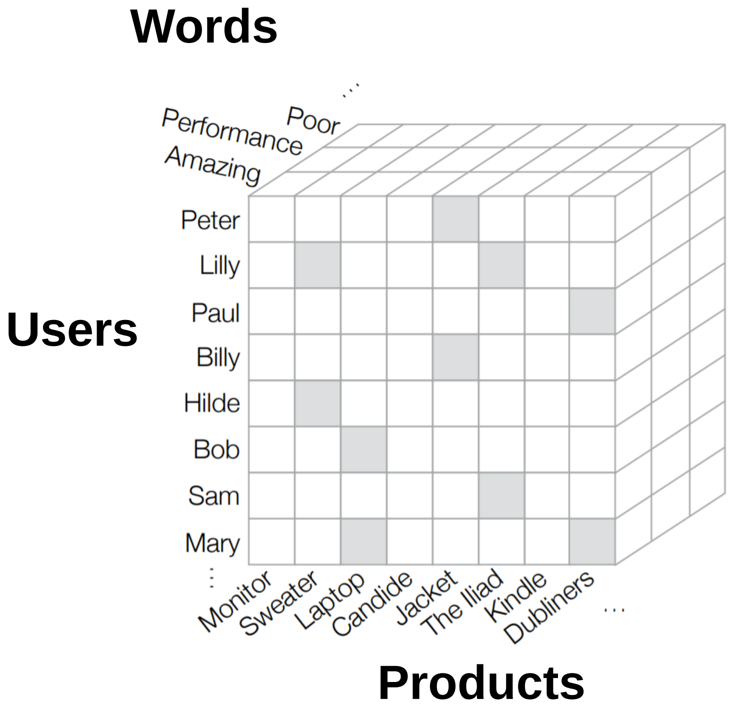
Nitish Srivastava, Hanchen Jin, Shaden Smith[2], Hongbo Rong[3],
David Albonesi, and Zhiru Zhang

Cornell University
[2]Microsoft AI & Research
[3]Intel Parallel Computing Lab

What is a Tensor?

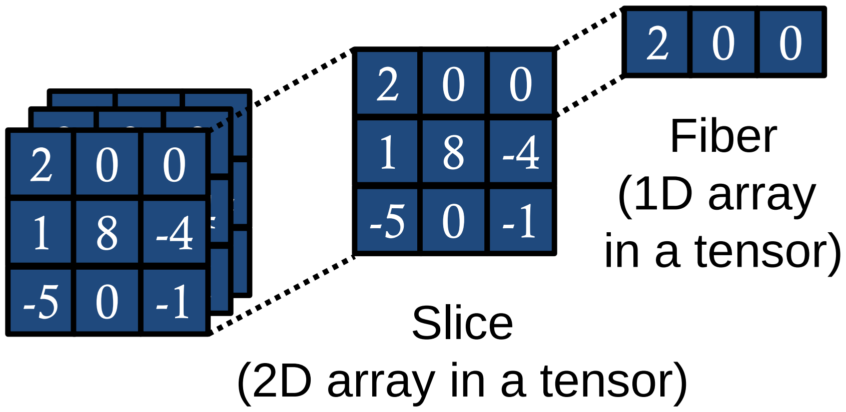
- **Tensors** are generalization of matrices to n dimensions
  - Scalar is tensor with 0 dimensions
  - Vector is tensor with 1 dimension
  - Matrix is tensor with 2 dimensions, and so on





Product Reviews

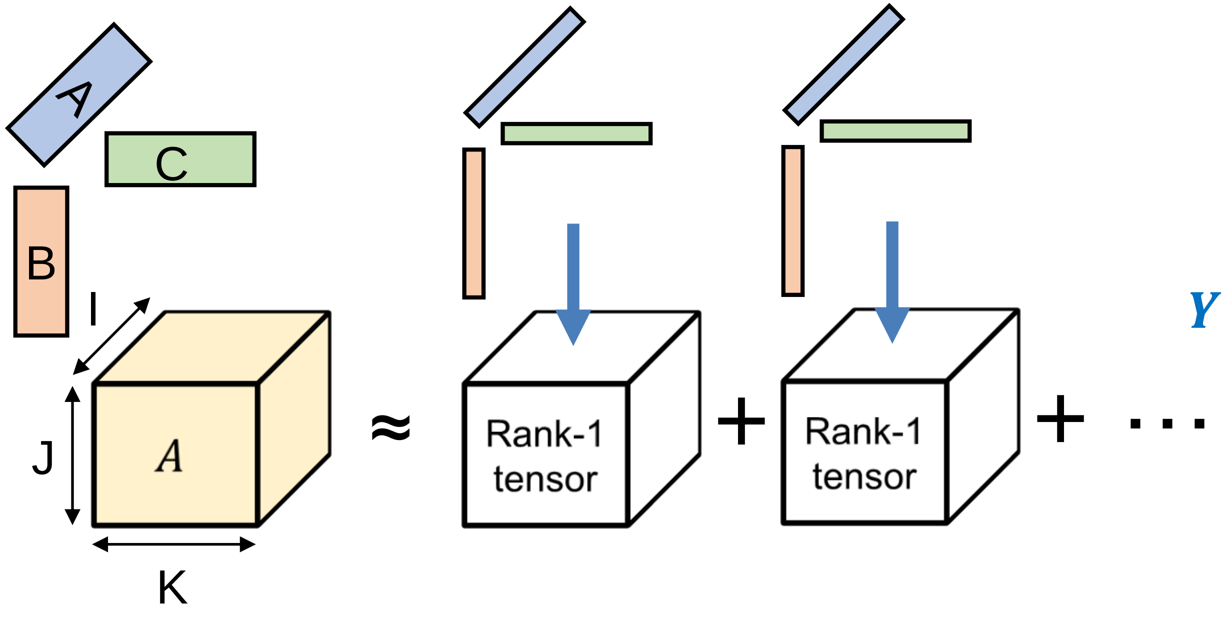
- High-dimensional data
- Density: 10^{-7} %
- Requires low-dimensional representation for ease of analysis



Tensor Decompositions for Low-Dimensional Representation

Sparse data

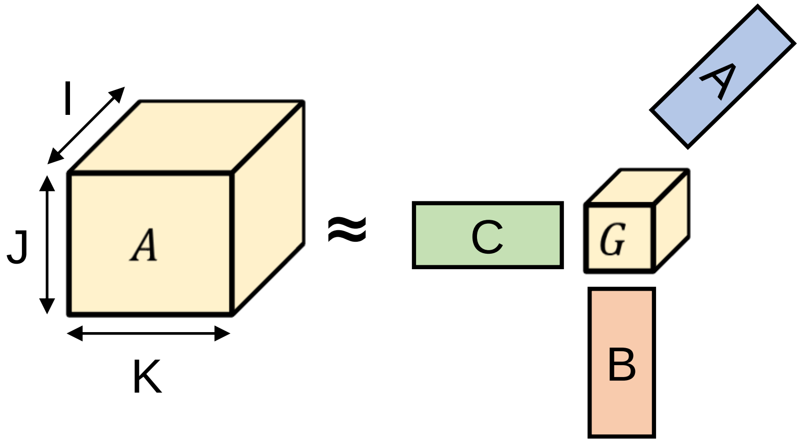
Dense data



CP Decomposition

$$Y(i,f) = \sum_{j=0}^{J} B(j,f) \cdot \sum_{k=0}^{K} A(i,j,k) \cdot C(k,f)$$

MTTKRP

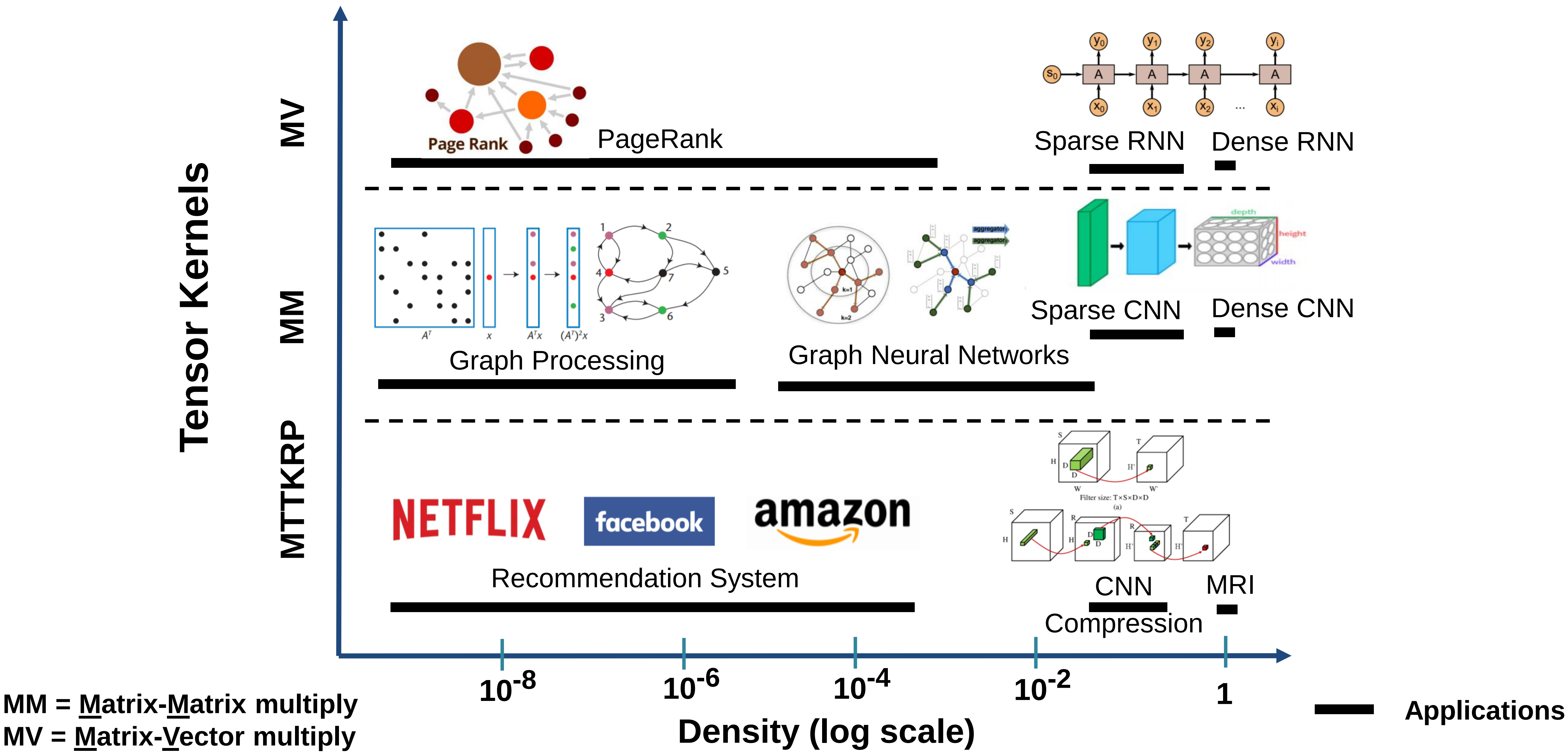


Tucker Decomposition

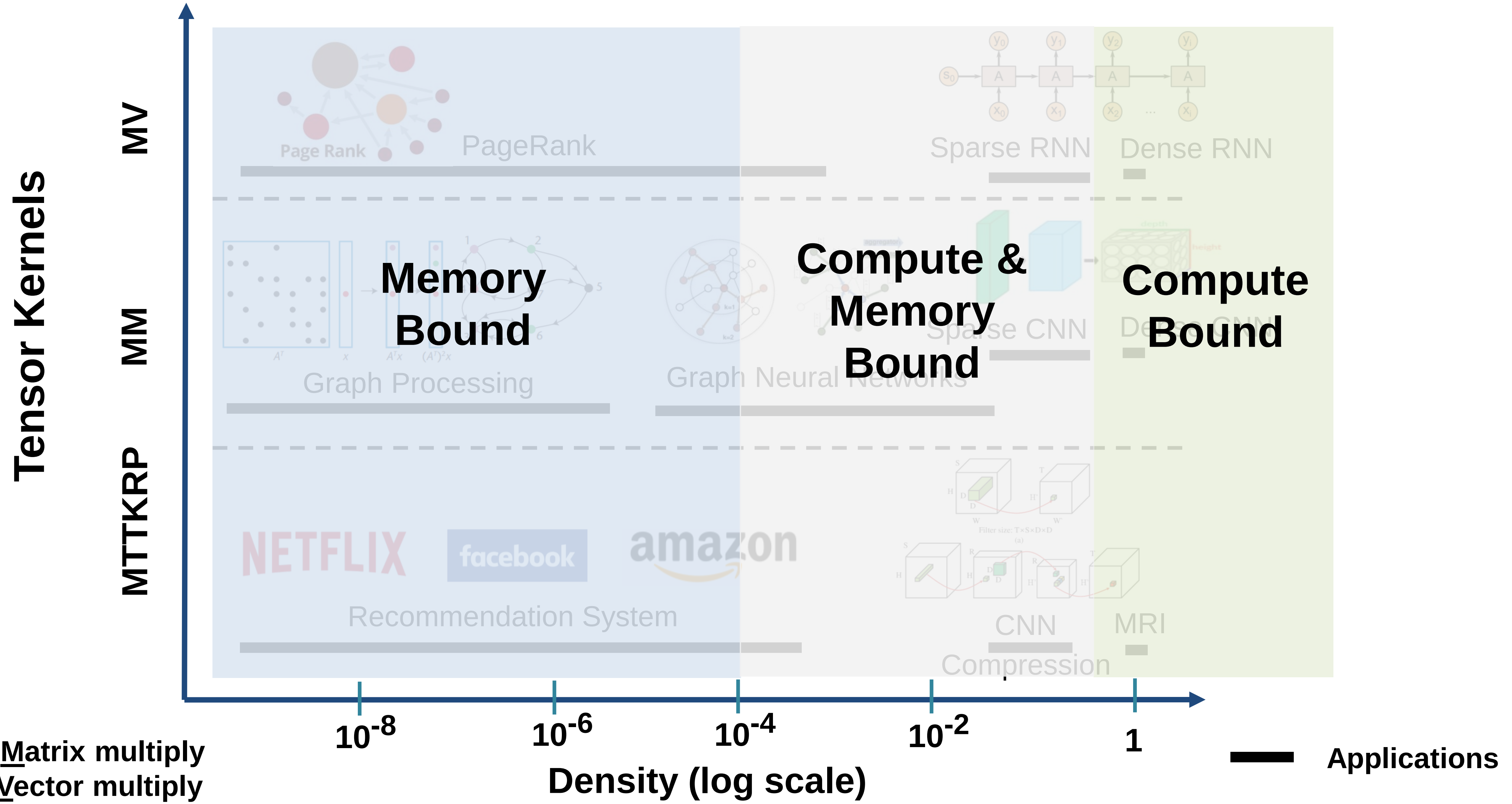
$$Y(i,f,g) = \sum_{j=0}^{J} B(j,f) \cdot \sum_{k=0}^{K} A(i,j,k) \cdot C(k,g)$$

TTMc

Kernel-Sparsity Spectrum of Tensor Applications

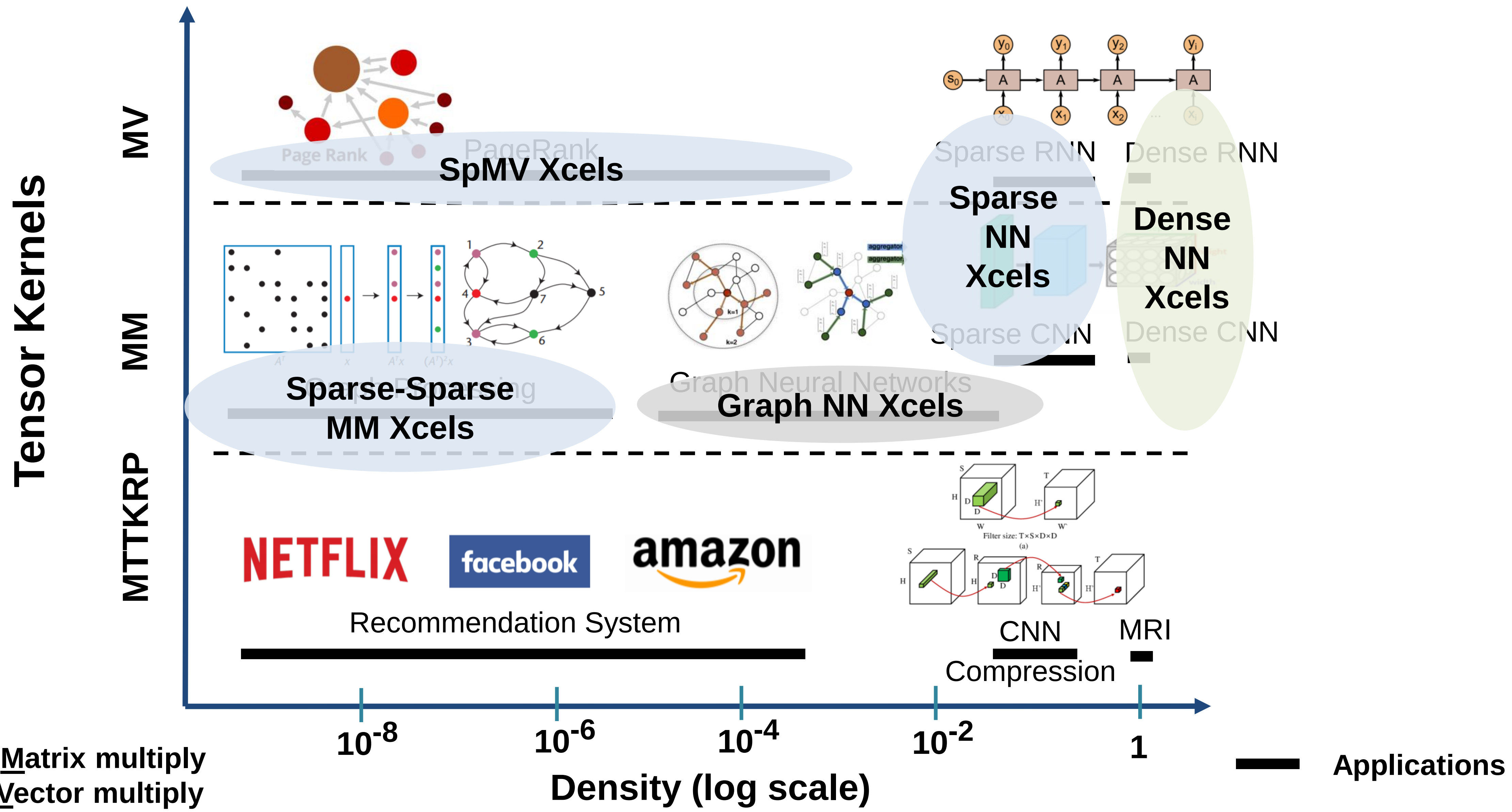


Kernel-Sparsity Spectrum of Tensor Applications



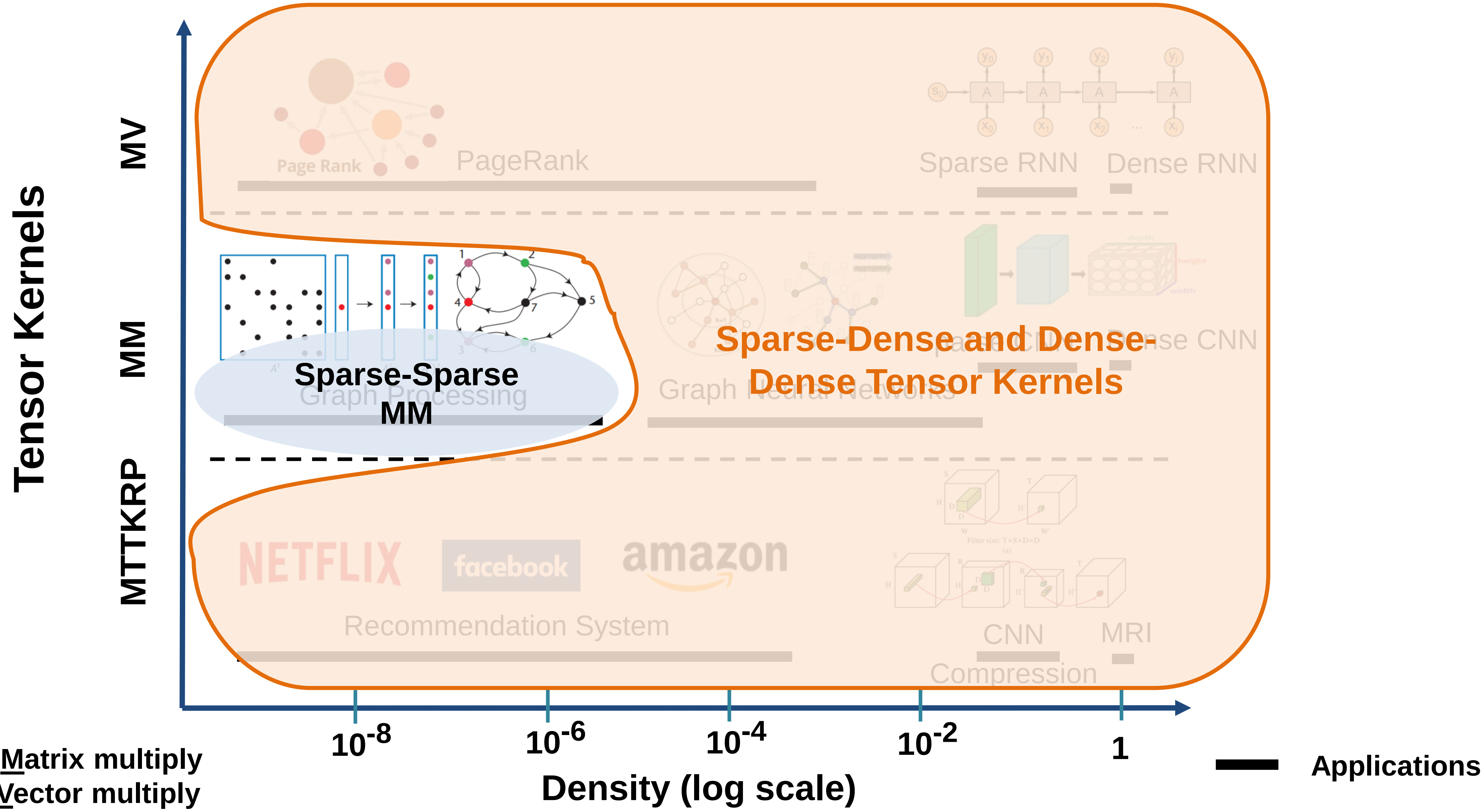
MM = Matrix-Matrix multiply
MV = Matrix-Vector multiply

Kernel-Sparsity Spectrum of Tensor Applications



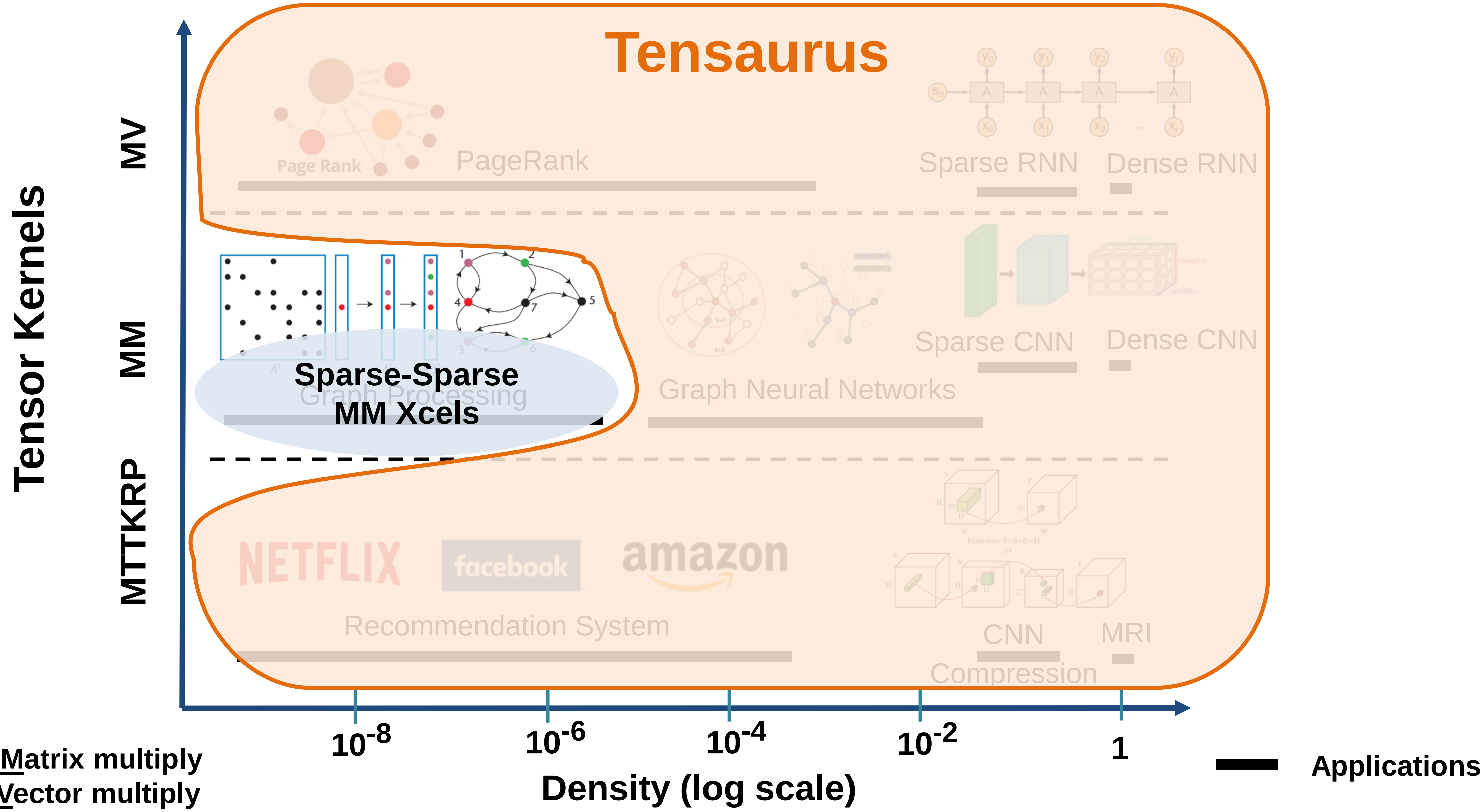
MM = Matrix-Matrix multiply
MV = Matrix-Vector multiply

Kernel-Sparsity Spectrum of Tensor Applications



MM = Matrix-Matrix multiply
MV = Matrix-Vector multiply

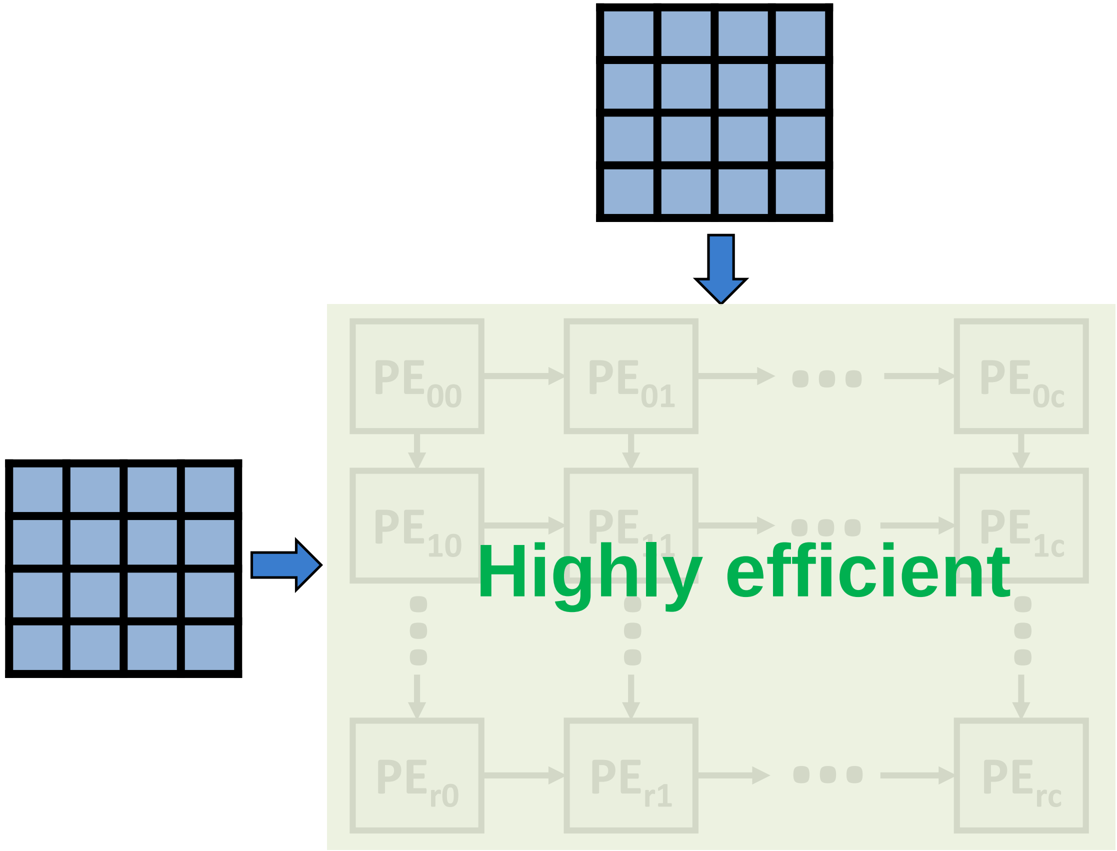
Kernel-Sparsity Spectrum of Tensor Applications



MM = Matrix-Matrix multiply
MV = Matrix-Vector multiply

Challenges with Sparse-Dense Tensor Acceleration

- **Dense Tensor Acceleration**
  - Systolic arrays provide high utilization of both memory and compute



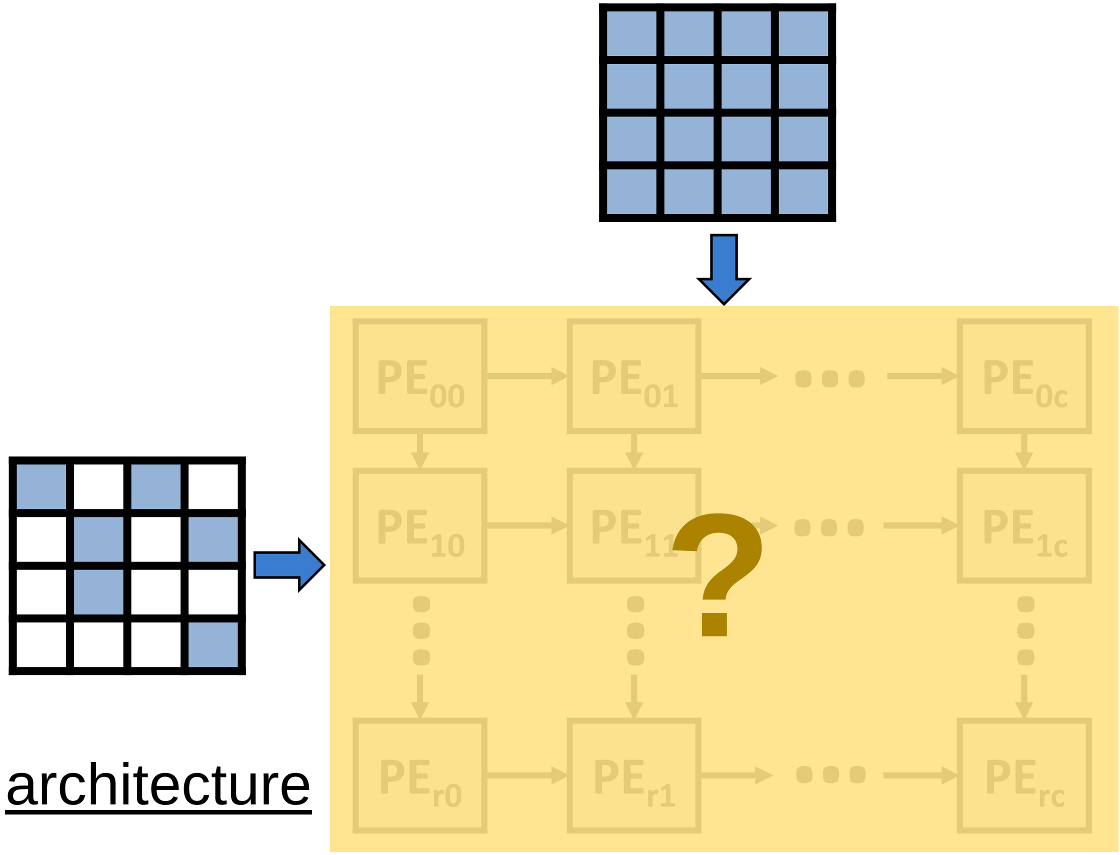
Challenges with Sparse-Dense Tensor Acceleration

- **Dense Tensor Acceleration**
  - Systolic arrays provide high utilization of both memory and compute

- **Mixed Sparse-Dense Tensor Acceleration**
  - Memory bound
  - Hard to achieve high compute and bandwidth utilization

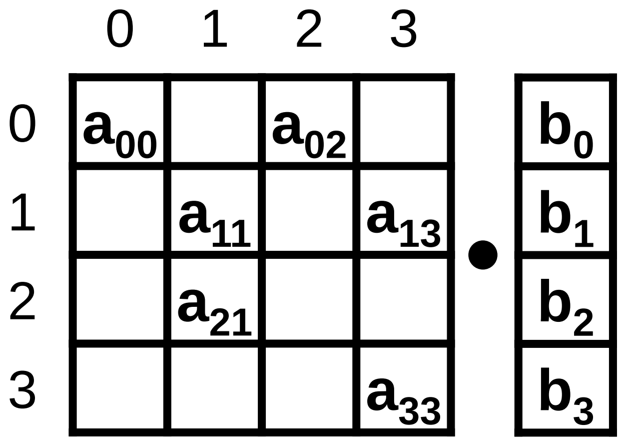
- **Goal:** Leverage a dense accelerator to efficiently perform sparse-dense compute

- **Key approach:** Co-design of accelerator <u>architecture</u> and <u>sparse format</u>
  - Low overhead of supporting sparse compute
  - High compute and bandwidth utilization



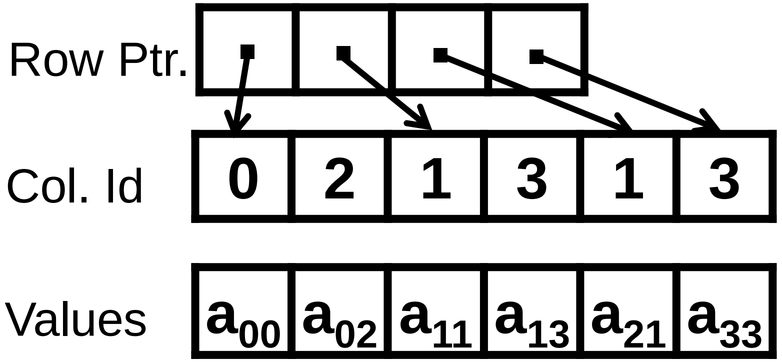
How to push sparse data in dense systolic array?

Importance of Accelerator Friendly Formats



SpMV
(Sparse Matrix dense Vector multiply)

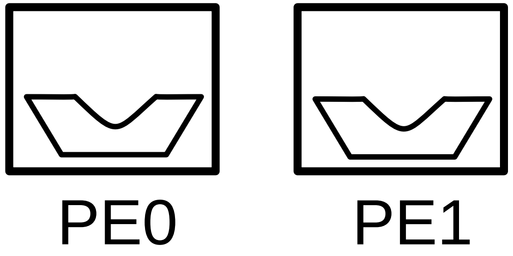
Compressed Sparse Row Format (CSR)



Importance of Accelerator Friendly Formats

$$
\begin{array}{cc}
\text{PE0} & 0 \\
\text{PE1} & 1 \\
\text{PE0} & 2 \\
\text{PE1} & 3
\end{array}
\begin{bmatrix}
a_{00} & & a_{02} & \\
 & a_{11} & & a_{13} \\
 & a_{21} & & \\
 & & & a_{33}
\end{bmatrix}
\bullet
\begin{bmatrix}
b_0 \\ b_1 \\ b_2 \\ b_3
\end{bmatrix}
$$

SpMV
(Sparse Matrix dense Vector multiply)

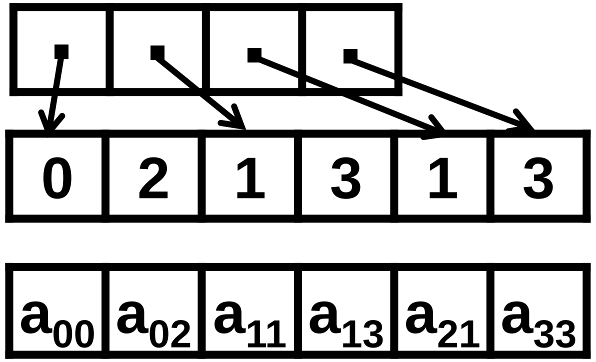


Compressed Sparse Row Format (CSR)

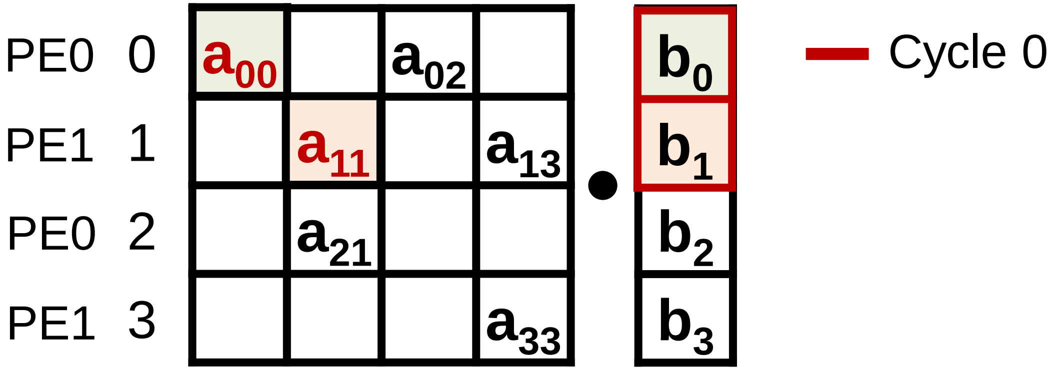
Row Ptr.

Col. Id

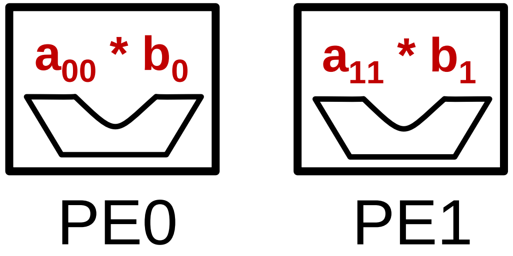
Values



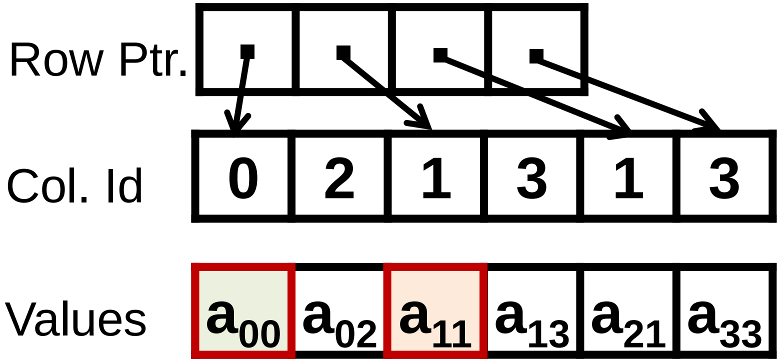
Importance of Accelerator Friendly Formats



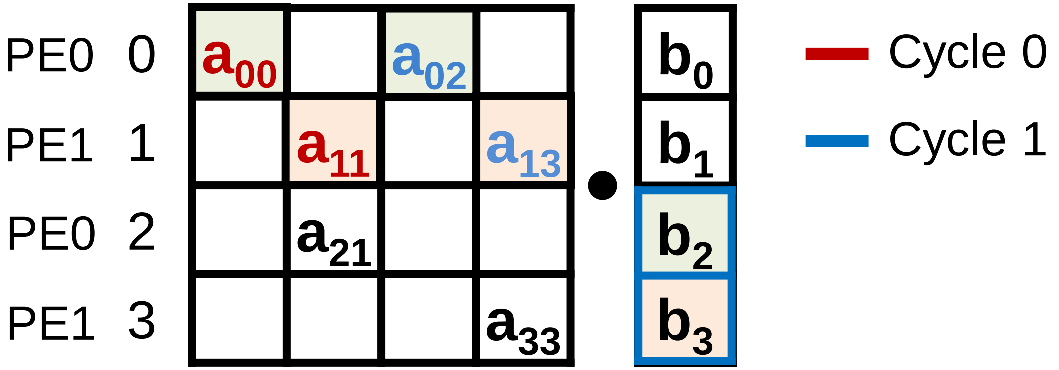
SpMV
(Sparse Matrix dense Vector multiply)

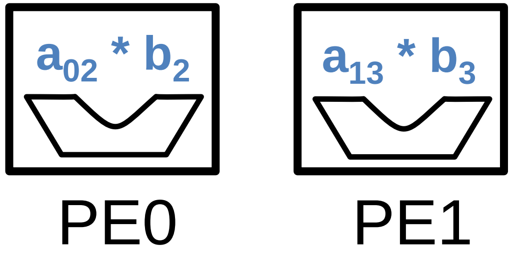


Compressed Sparse Row Format
(CSR)



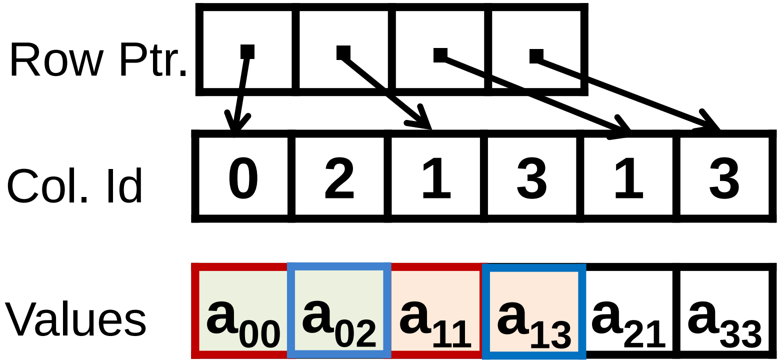
Importance of Accelerator Friendly Formats



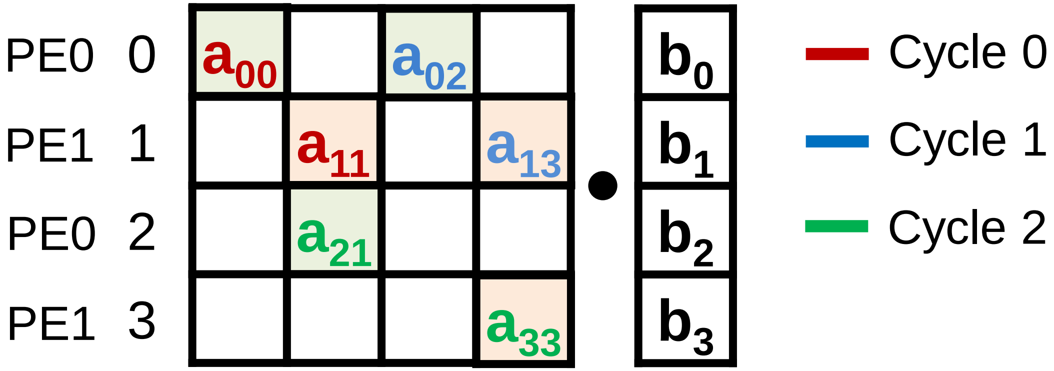


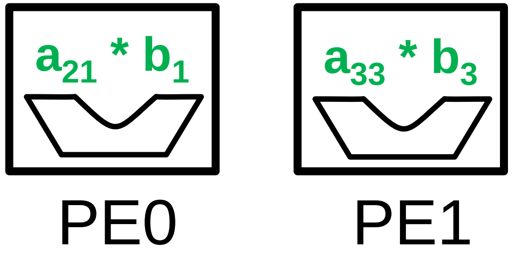
SpMV
(Sparse Matrix dense Vector multiply)

Compressed Sparse Row Format (CSR)



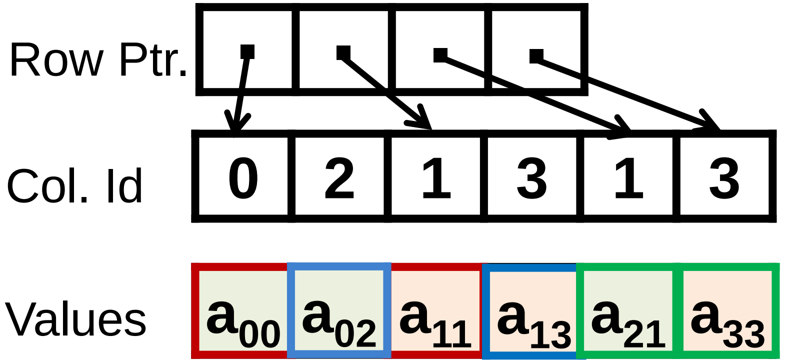
Importance of Accelerator Friendly Formats



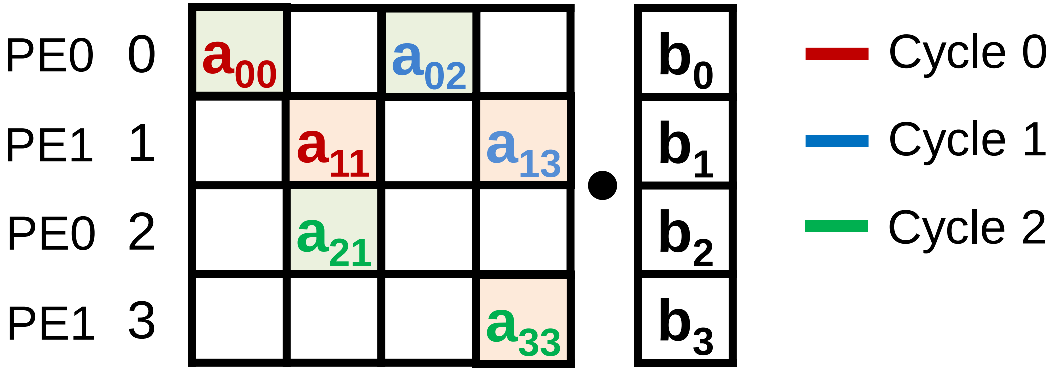


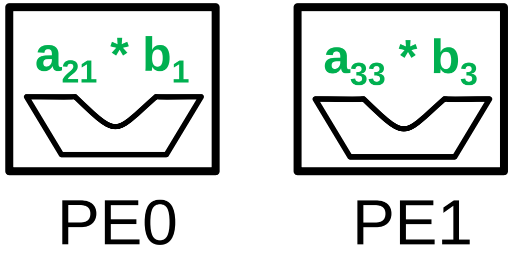
SpMV
(Sparse Matrix dense Vector multiply)

Compressed Sparse Row Format (CSR)



Importance of Accelerator Friendly Formats

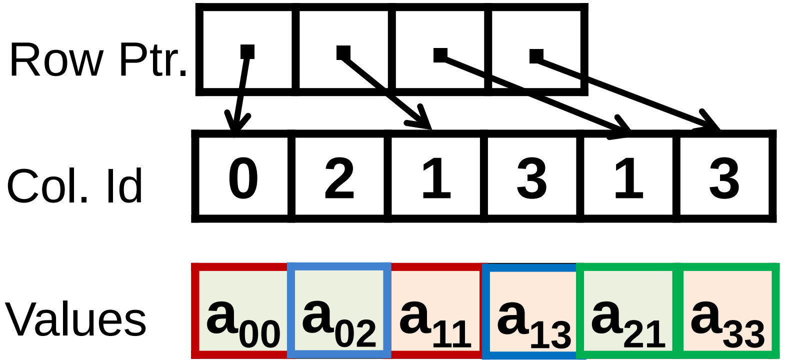




SpMV

(Sparse Matrix dense Vector multiply)

Compressed Sparse Row Format (CSR)



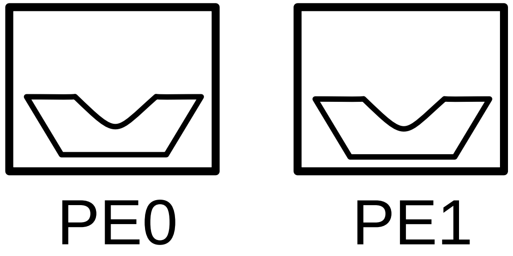
Problems with CSR

- **Non-streaming & non-vectorized accesses**
- **Indirect memory access**

Importance of Accelerator Friendly Formats

		0	1	2	3		
PE0	0	a_{00}		a_{02}			b_0
PE1	1		a_{11}		a_{13}	$\bullet$	b_1
PE0	2		a_{21}				b_2
PE1	3				a_{33}		b_3

SpMV
(Sparse Matrix dense Vector multiply)



Compressed Interleaved Sparse Row (CISR) [1]

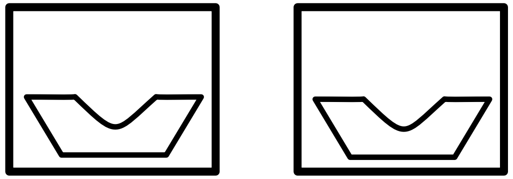
PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}

[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Importance of Accelerator Friendly Formats

		0	1	2	3		
PE0	0	a_{00}		a_{02}			b_0
PE1	1		a_{11}		a_{13}	$\bullet$	b_1
PE0	2		a_{21}				b_2
PE1	3				a_{33}		b_3

SpMV
(Sparse Matrix dense Vector multiply)



PE0 PE1

Compressed Interleaved Sparse Row (CISR) [1]

PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}

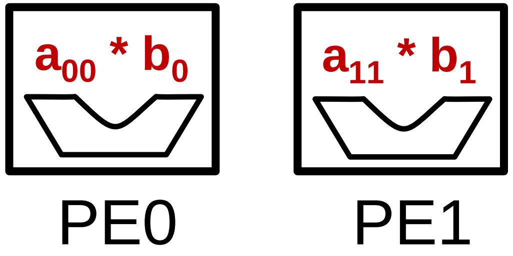
[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Importance of Accelerator Friendly Formats

		0	1	2	3
PE0	0	a_{00}		a_{02}	
PE1	1		a_{11}		a_{13}
PE0	2		a_{21}		
PE1	3				a_{33}

$$\bullet \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Cycle 0



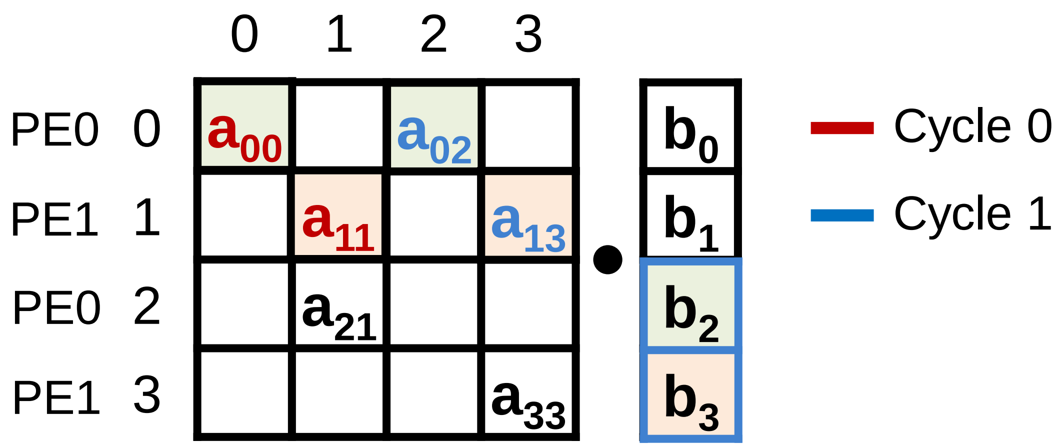
SpMV
(Sparse Matrix dense Vector multiply)

Compressed Interleaved Sparse Row (CISR) [1]

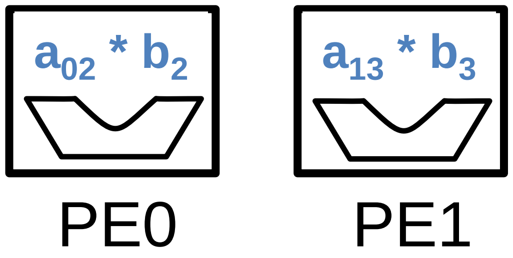
PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}

[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Importance of Accelerator Friendly Formats



SpMV
(Sparse Matrix dense Vector multiply)

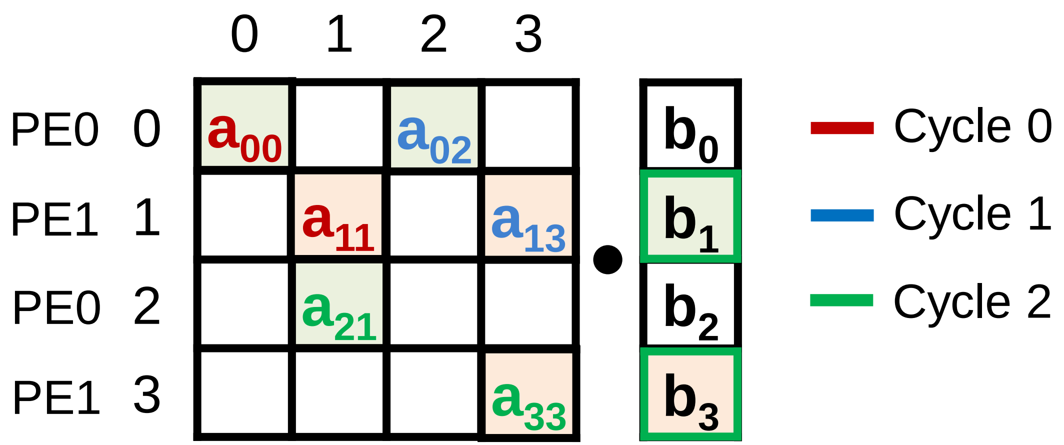


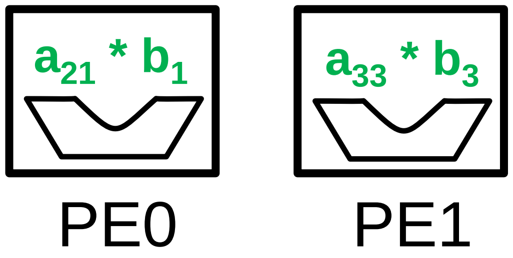
Compressed Interleaved Sparse Row (CISR) [1]

PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}

[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Importance of Accelerator Friendly Formats





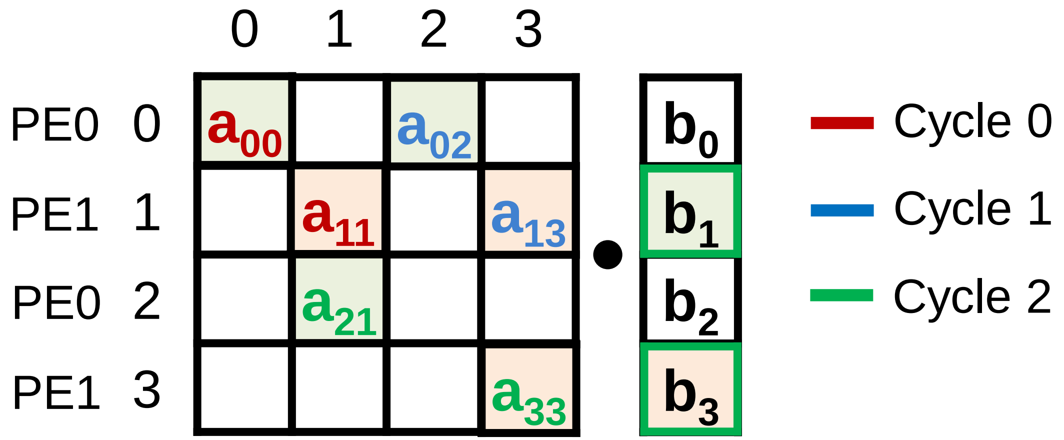
SpMV
(Sparse Matrix dense Vector multiply)

Compressed Interleaved Sparse Row (CISR) [1]

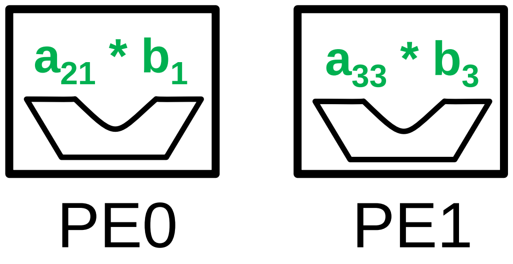
PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}

[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Importance of Accelerator Friendly Formats

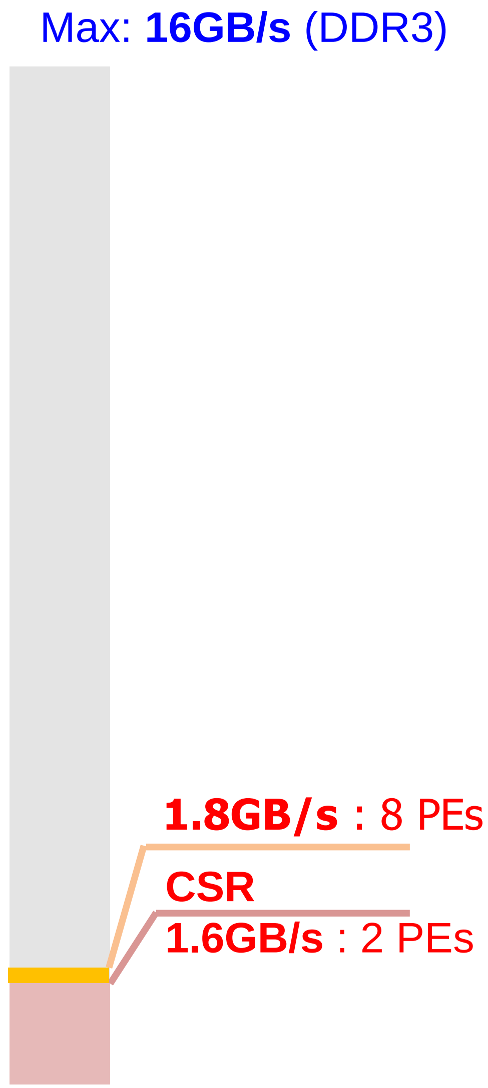


SpMV
(Sparse Matrix dense Vector multiply)



Compressed Interleaved Sparse Row (CISR) [1]

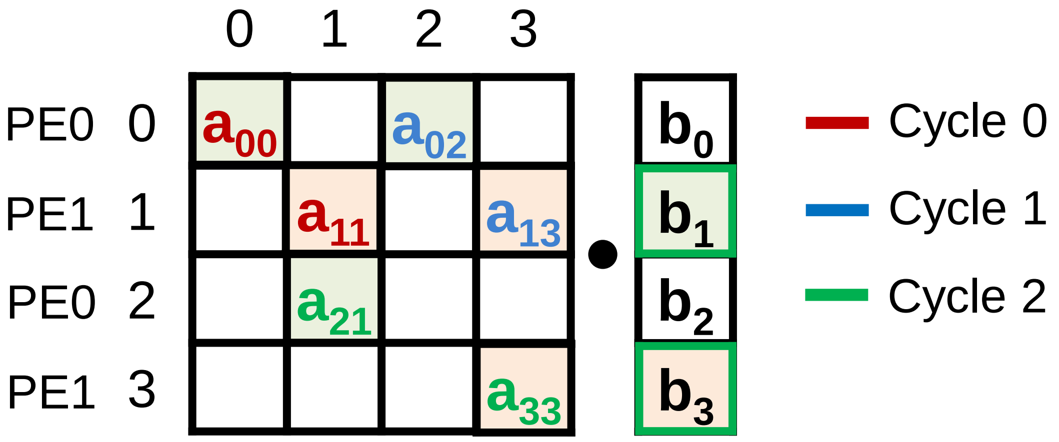
PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}



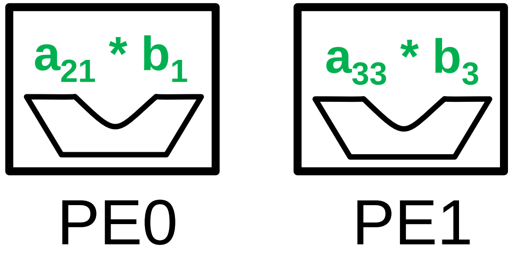
Utilized Bandwidth vs. # PEs

[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Importance of Accelerator Friendly Formats

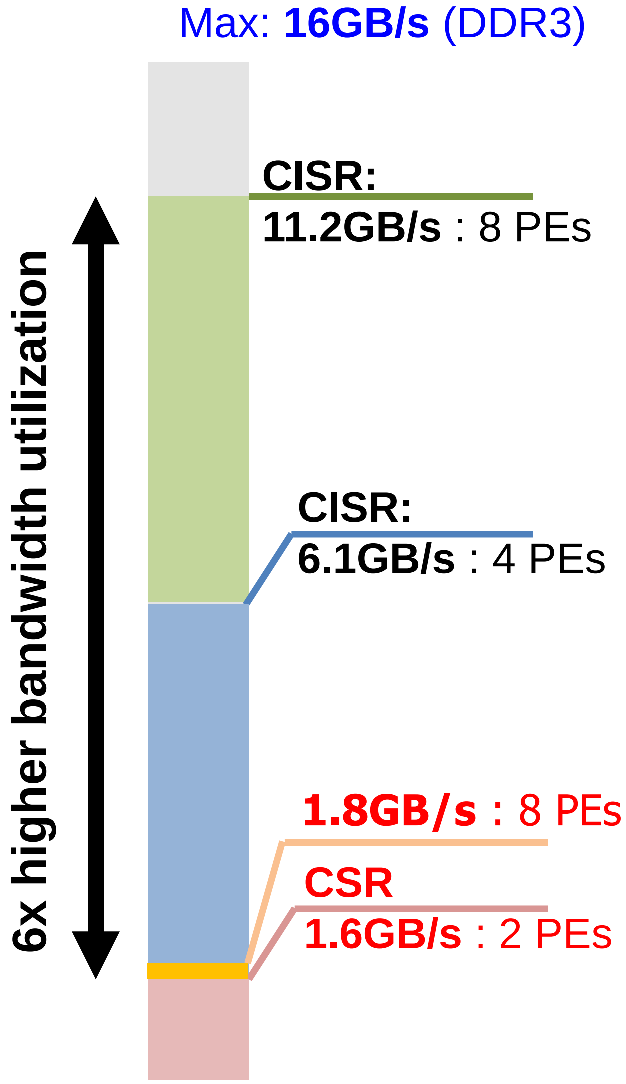


SpMV
(Sparse Matrix dense Vector multiply)



Compressed Interleaved Sparse Row (CISR) [1]

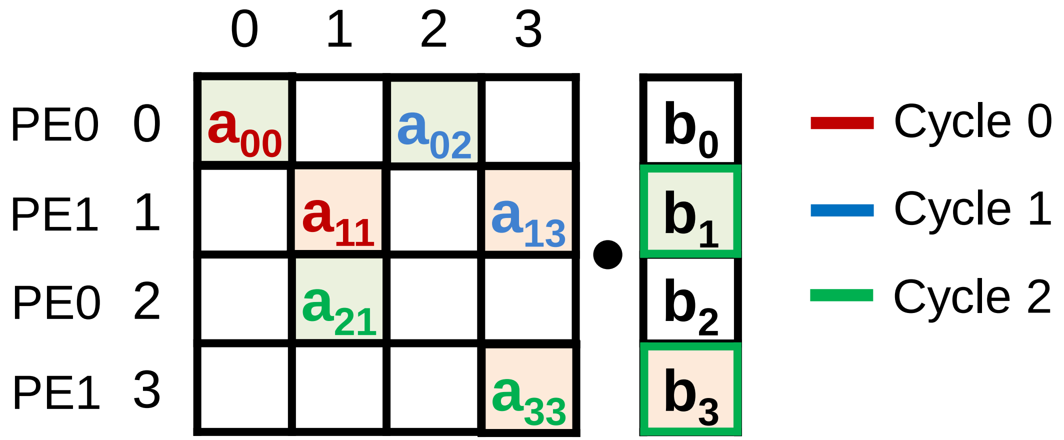
PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}



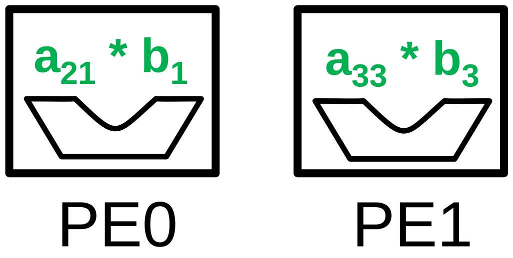
Utilized Bandwidth vs. # PEs

[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Importance of Accelerator Friendly Formats



SpMV
(Sparse Matrix dense Vector multiply)

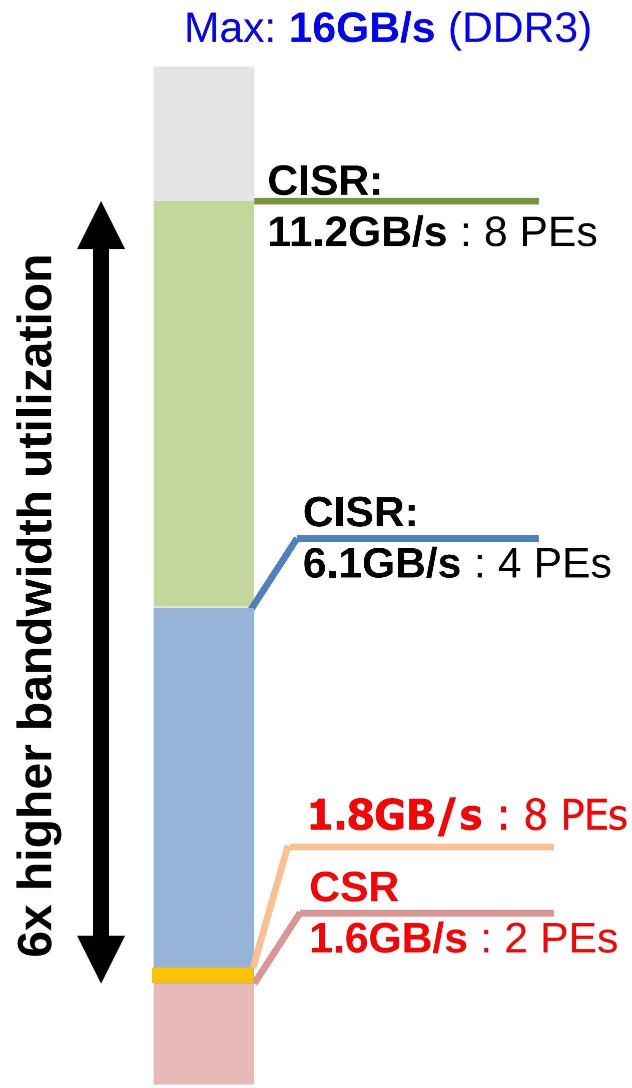


Compressed Interleaved Sparse Row (CISR) [1]

PE	0	1	0	1	0	1
Row Len.	2	2	1	1		
Col. Id	0	1	2	3	1	3
Values	a_{00}	a_{11}	a_{02}	a_{13}	a_{21}	a_{33}

Benefits of CISR

- **Streaming & vectorized accesses**



Utilized Bandwidth vs. # PEs

[1] Fowers, et al. A high memory bandwidth FPGA accelerator for sparse matrix-vector multiplication, Int'l Symp. On *Field-Programmable Custom Computing Machines (FCCM), 2014*

Extending CISR to CISS for Tensors

		0	1	2	3
PE0	0	a_{00}		a_{02}	
PE1	1		a_{11}		a_{13}
PE0	2		a_{21}		
PE1	3				a_{33}

CISR+

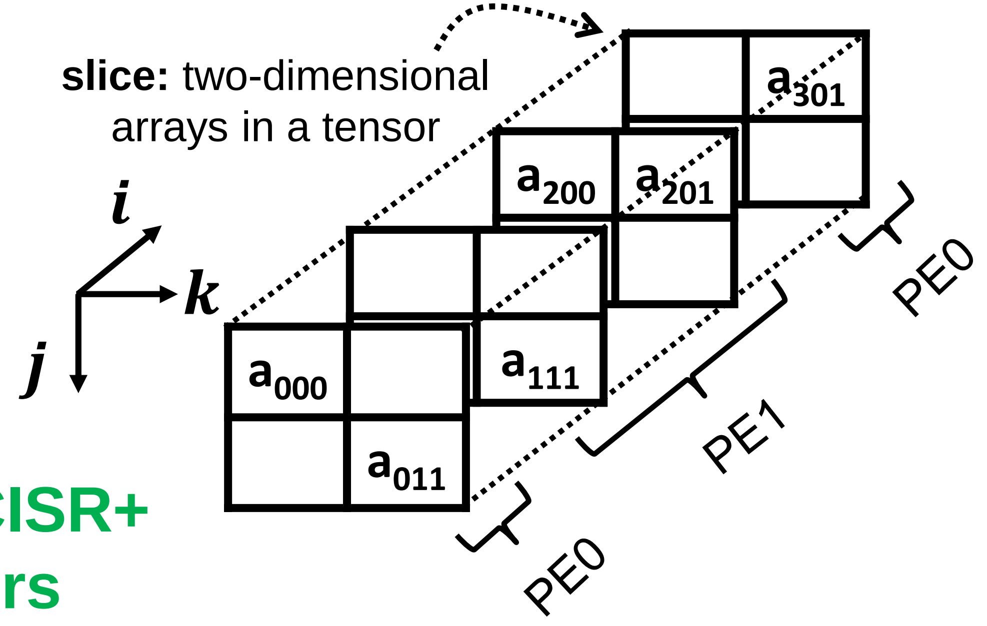
PE	0	1	0	1	0	1	0	1	0	1
Row Id/ Col. Id	0	1	0	1	2	3	2	3	1	3
Values	0	0	a_{00}	a_{11}	a_{02}	a_{13}	0	0	a_{21}	a_{33}

CISR+ avoids row-decoding

Adds extra zeros to make design more scalable ⇨ HW/SW co-design

Extending CISR+ to Tensors

slice: two-dimensional arrays in a tensor



Compressed Interleaved Sparse Slice (CISS)

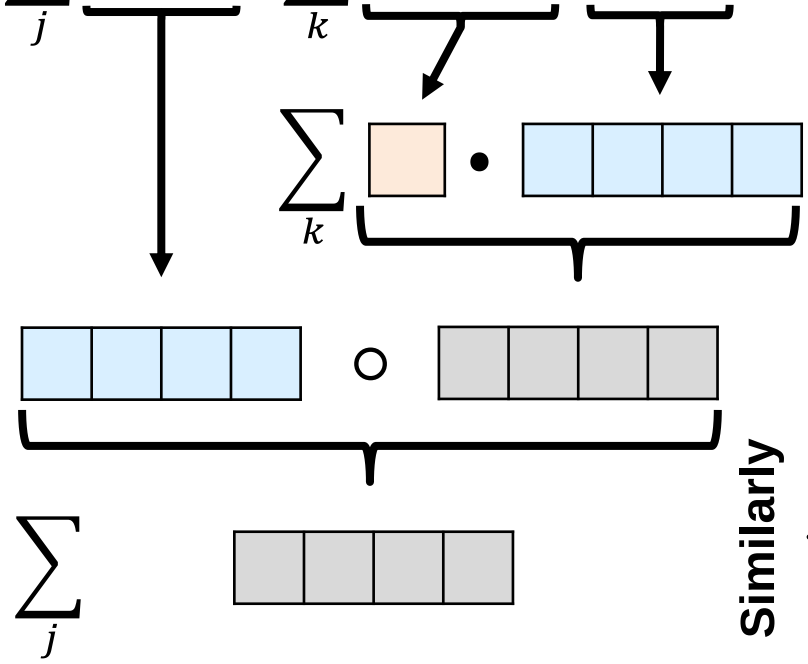
PE	0	1	0	1	0	1	0	1	0	1
k	x	x	0	1	1	x	x	0	1	1
i/j	0	1	0	1	1	2	3	0	0	0
Values	0	0	a_{000}	a_{111}	a_{011}	0	0	a_{200}	a_{301}	a_{201}

Computation Pattern for Tensor Kernels

MTTKRP

fiber fiber op scalar fiber

$$\forall i \quad Y_{(i,:)} = \sum_j B(j,:) \circ \sum_k A(i,j,k) \cdot C(k,:)$$



Scalar-Fiber product followed by Fiber-Fiber product (SF³) Pattern

$$\boxed{fibers_{out} = \sum_{D_1} fiber_1 \; op \sum_{D_0} (scalar \cdot fiber_0)}$$

For MTTKRP

D_1: $[0, J)$ **or** $\{j \mid \exists k \; st. \; A(i,j,k) \neq 0\}$

D_0: $[0, K)$ **or** $\{k \mid A(i,j,k) \neq 0\}$

dense

sparse

Similarly

TTMc $\forall i \quad Y(i,:) = \sum_j B(j,:) \otimes \sum_k (A(i,j,k) \cdot C(k,:))$

MM $\forall i \quad Y(i,:) = \sum_{\phi} null \;\; op \sum_j (A(i,j) \cdot B(j,:))$

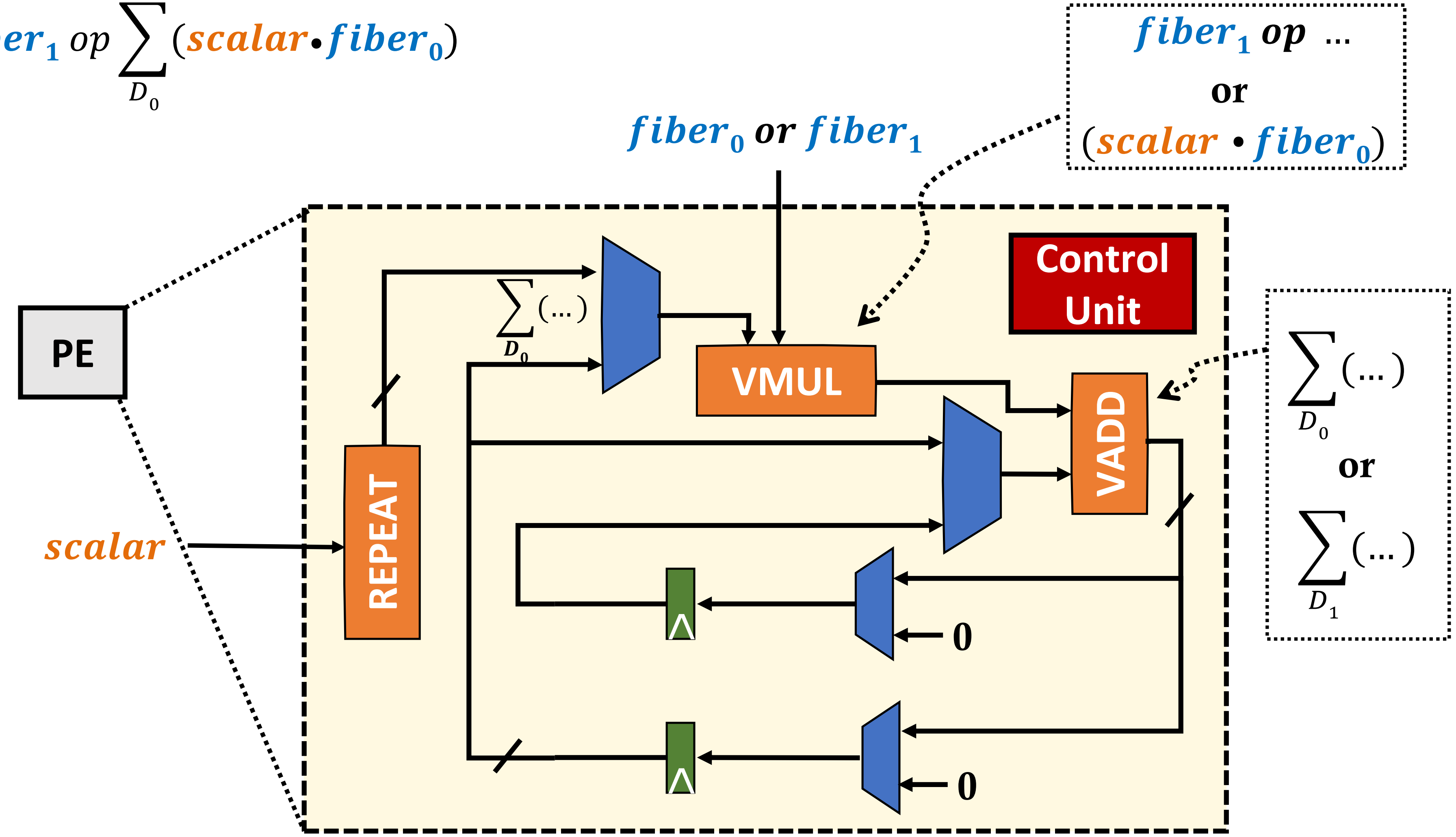
MV $\forall i \quad Y(i,:) = \sum_{\phi} null \;\; op \sum_j (A(i,j) \cdot b(j,:))$

SF³ compute pattern can express all the common dense and mixed sparse-dense tensor kernels

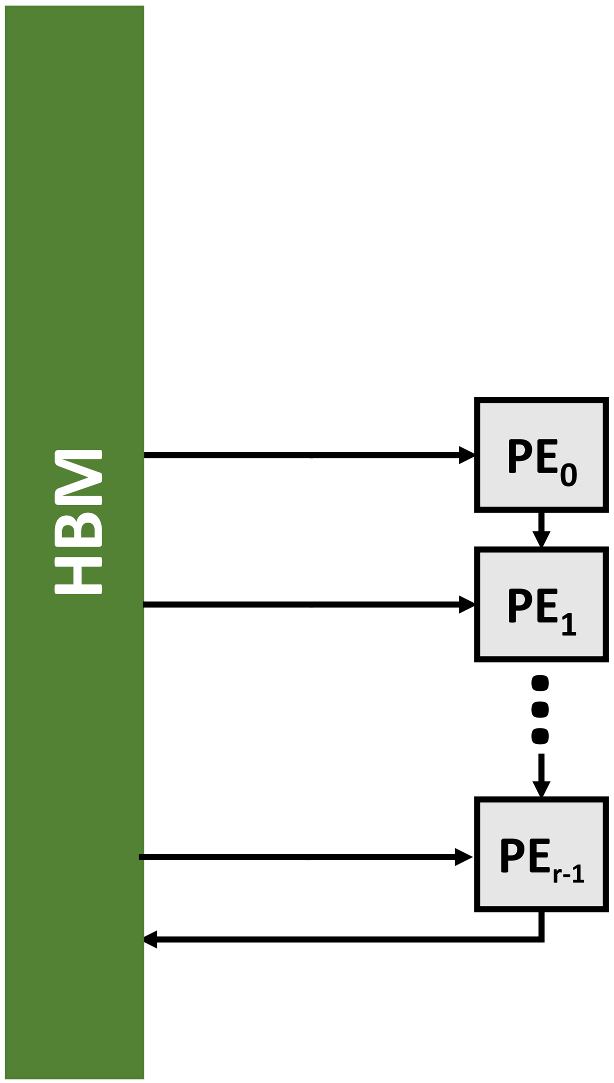
PE for SF³ Compute Pattern

$$\underbrace{fibers_{out}}_{} = \sum_{D_1} fiber_1 \; op \sum_{D_0} (scalar \bullet fiber_0)$$

Different output fibers can be computed in parallel



Vertical Scaling Using Coarse-Grained Parallelism

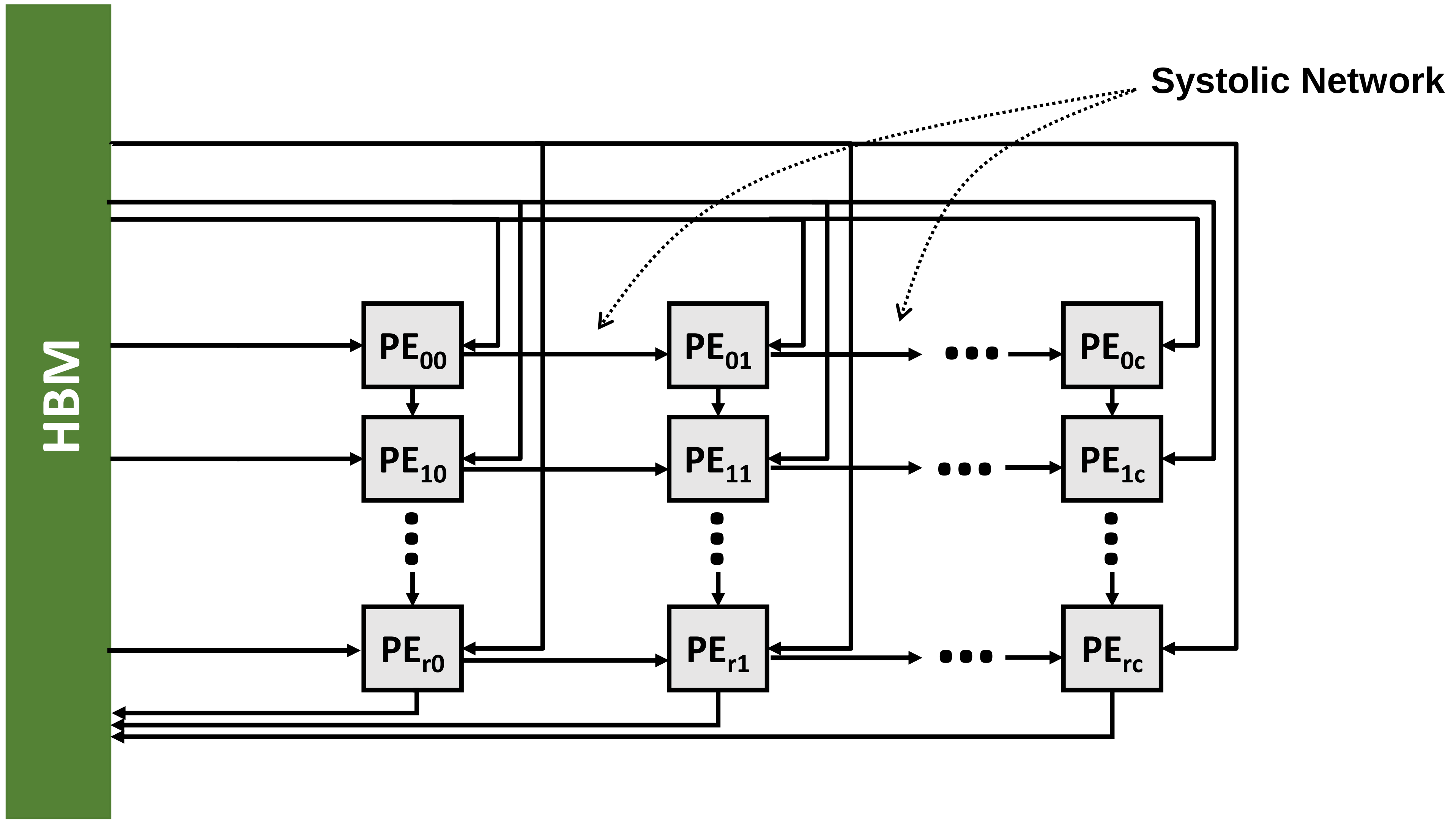


$$\mathrm{PE}_0: \quad fiber_{out}[0] = \sum_{D'_1} fiber_1 \; op \sum_{D'_0} (scalar \cdot fiber_0)$$

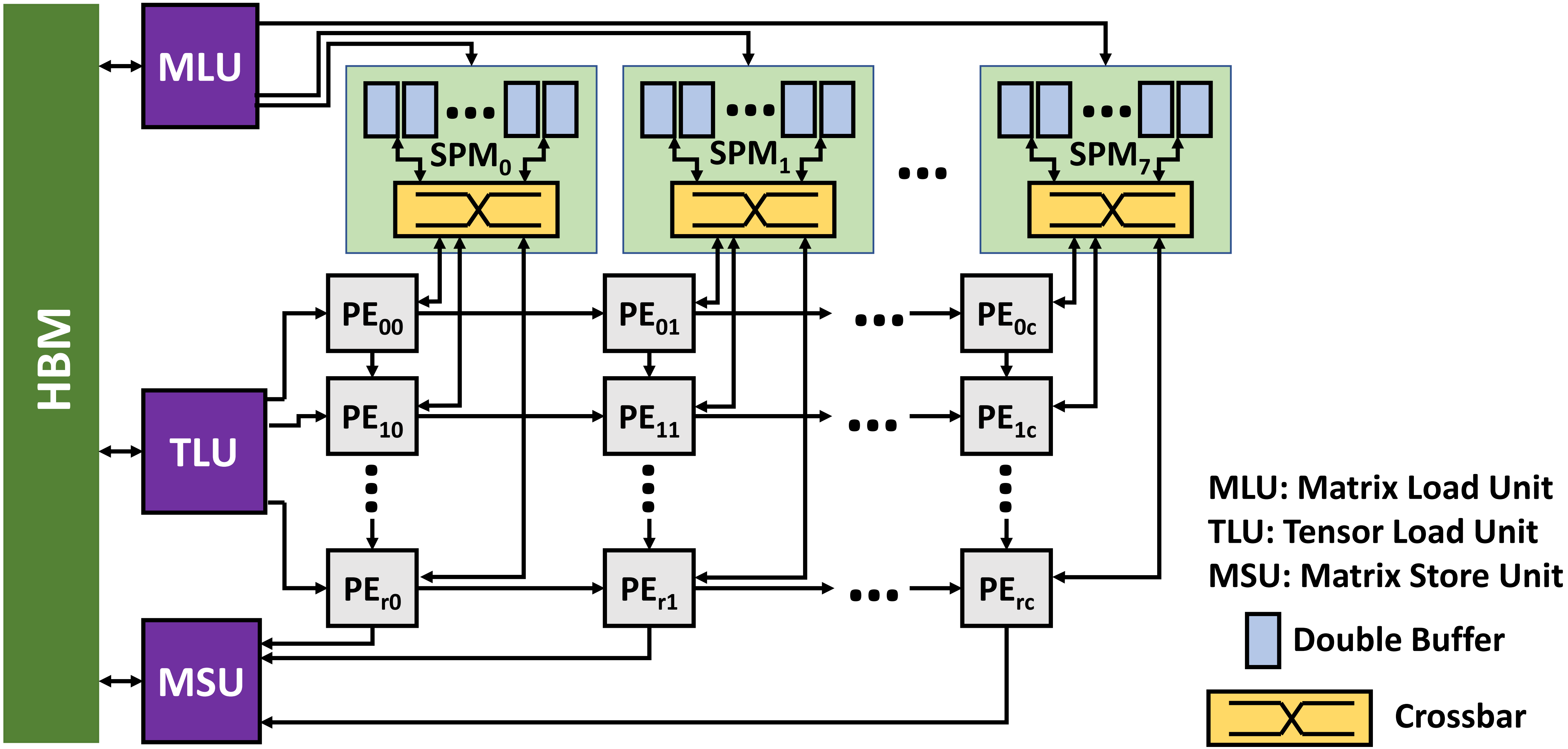
$$\mathrm{PE}_1: \quad fiber_{out}[1] = \sum_{D''_1} fiber_1 \; op \sum_{D''_0} (scalar \cdot fiber_0)$$

$$\mathrm{PE}_{r-1}: \quad fiber_{out}[r-1] = \underbrace{\sum_{D'''_1} fiber_1 \; op \sum_{D'''_0} (scalar \cdot fiber_0)}_{\text{Long vector SIMD compute}}$$

Horizontal Scaling Using SIMD-Vector Parallelism

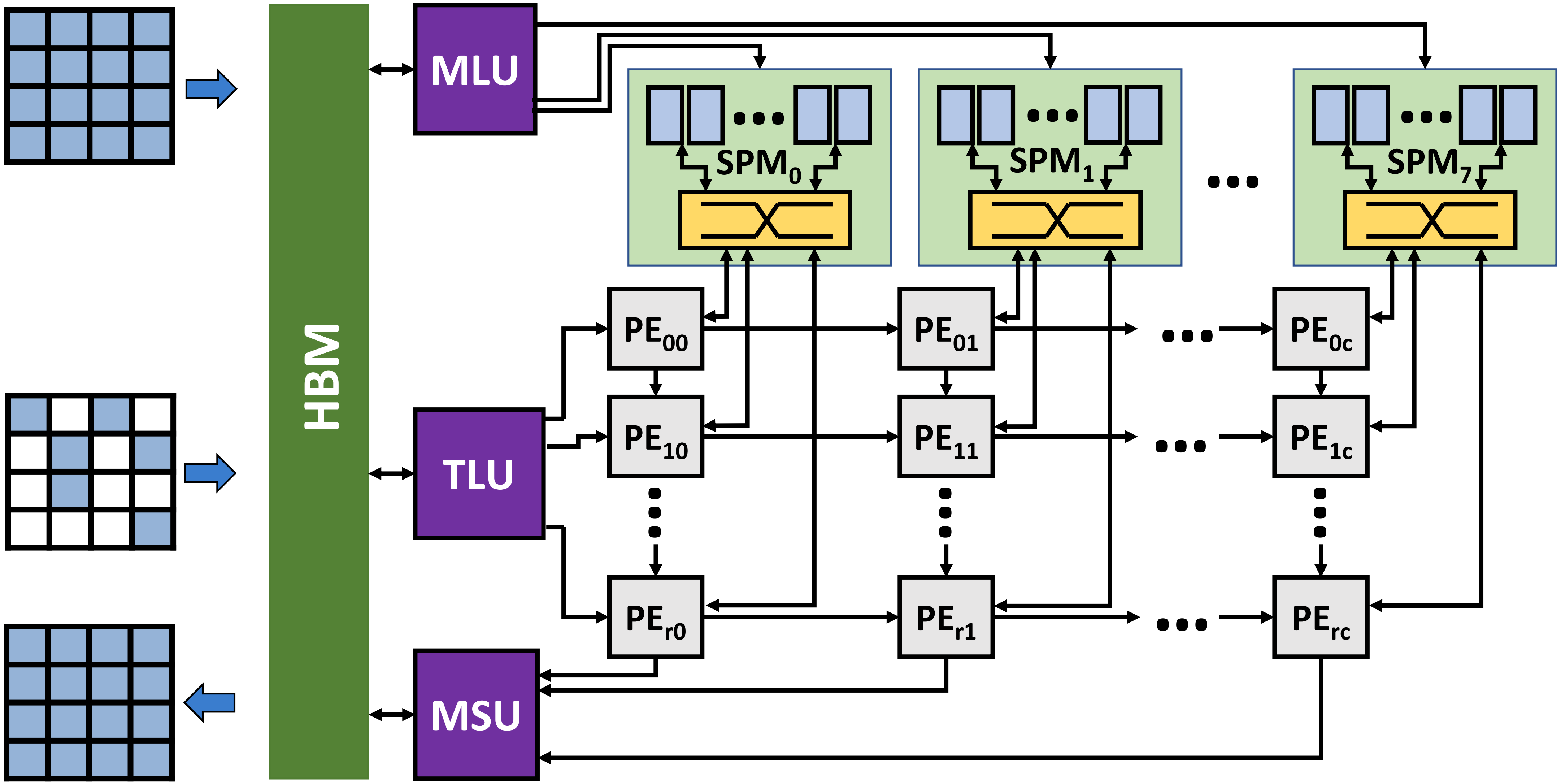


Tensaurus Architecture



Tensaurus Architecture

Accelerator for both dense and sparse-dense!!



Evaluation Methodology

- **Cycle-level simulation in gem5**
  - 8 x 8 PE array, VLEN = 8
  - 8 16KB RAMs per SPM
  - HBM: 8 128-bit physical channels (128 GB/s peak bandwidth)

- **RTL Modeling of a PE using PyMTL**
  - 28 nm (Synopsys & Cadence Tools)

Area and Power Breakdown

Component	Area(mm^2)	%	Power (mW)	%
PE	0.625	27.2 %	402.30	40.9 %
Xbar	0.066	2.8 %	24.27	2.5%
SPM	0.832	36.2 %	296.05	30.1 %
MSU	0.759	33.0 %	247.03	25.2 %
TLU	0.009	0.4 %	6.28	0.6%
MLU	0.009	0.4 %	6.28	0.6 %
Total	2.3	100 %	982.21	100 %

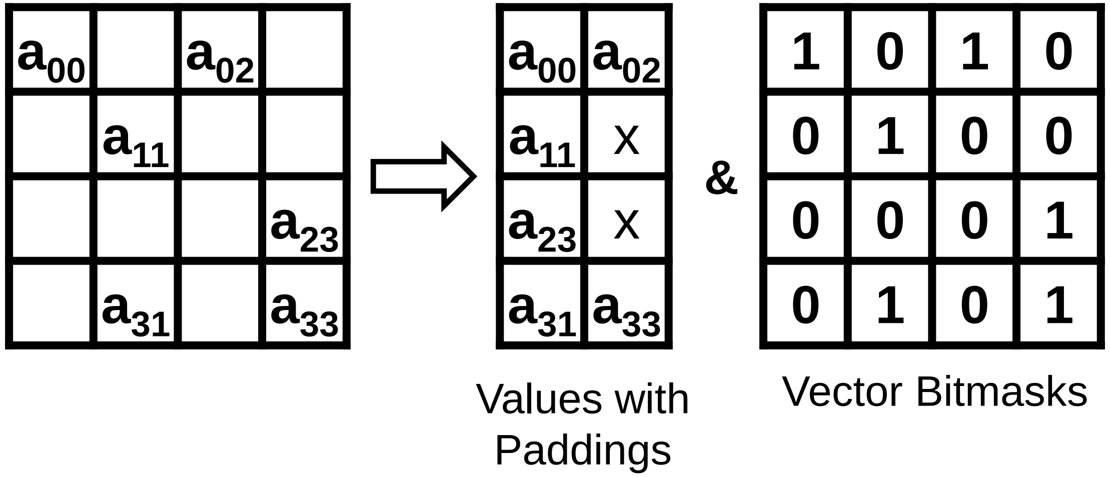
- **Baselines**
  - CPU: Intel(R) Xeon(R) CPU E7-8867
    - SparseBLAS and SPLATT
  - GPU: Titan XP
    - CuSparse, PaRTI
  - Sparse NN Accelerator:
    - Cambricon-X [1]

- **Datasets**
  - FROSTT Tensors, Florida Sparse Matrices, AlexNet, VGG-16

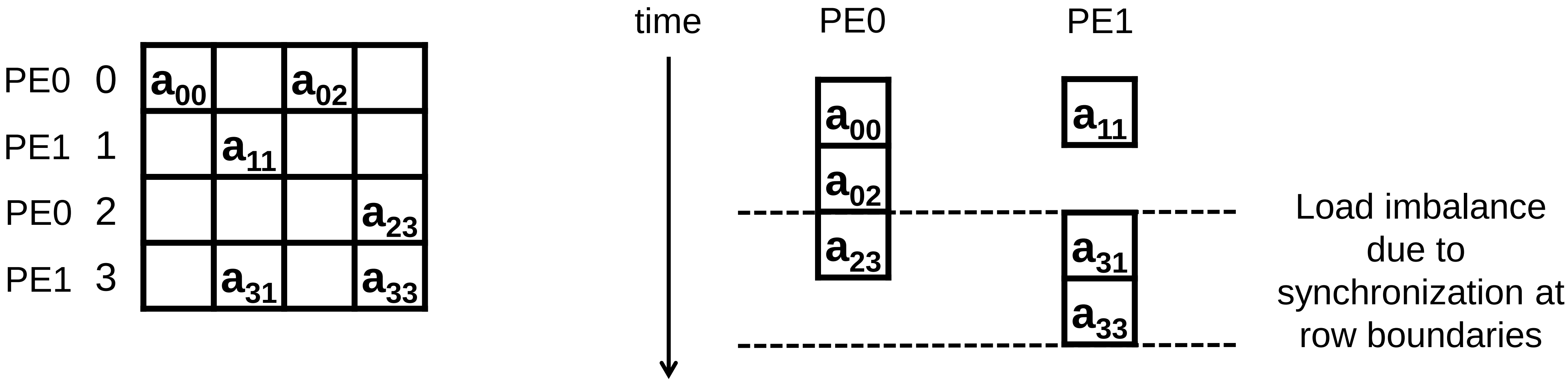
[1] Zhang, Shijin, et al. "Cambricon-x: An accelerator for sparse neural networks.", *Int'l Symp. on Microarchitecture (MICRO)*, 2016.

Cambricon-X Baseline

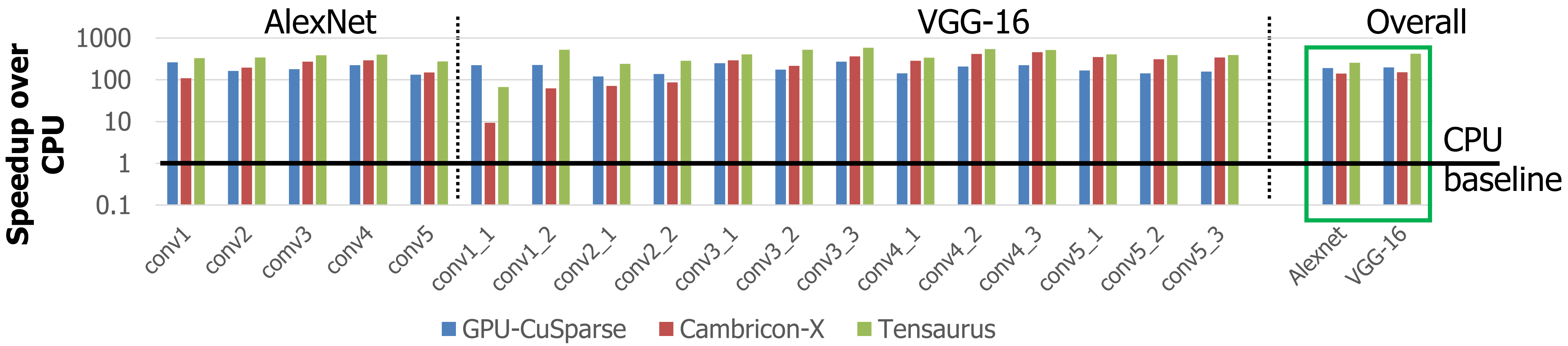
- Cambricon-X uses a CSR-variant
  - Pads empty entries with padding (x)
  - Uses vector bit-masks to indicate non-zero positions
  - Specialized for CNNs with low sparsity

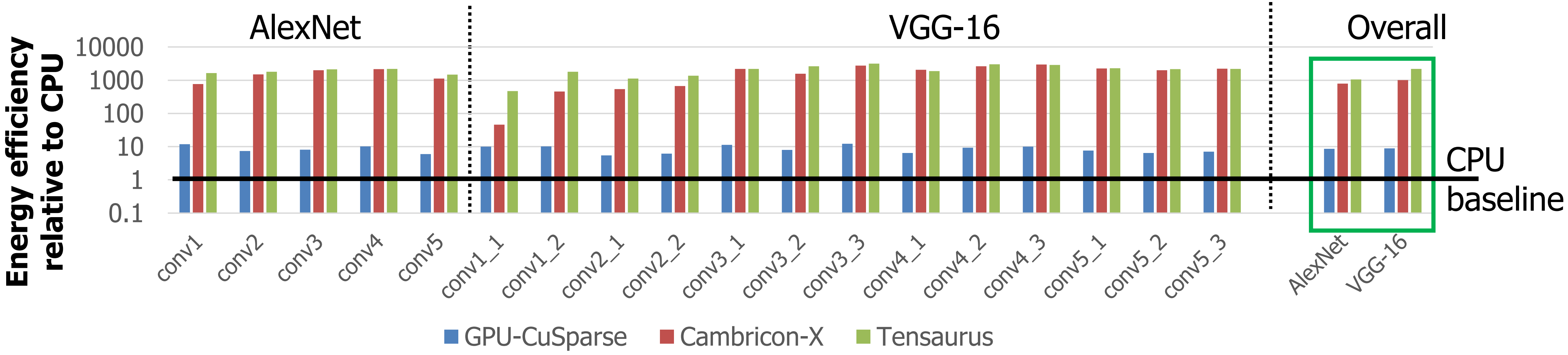


- CSR results in load-imbalanced schedule
  - Cambricon-X has synchronization boundaries across rows



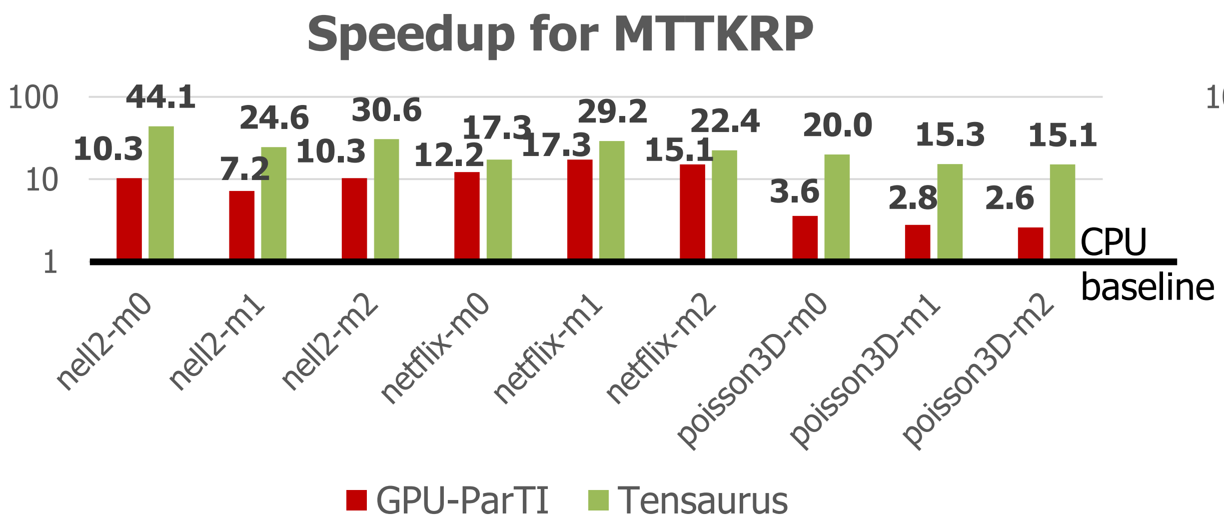
Results on Sparse Neural Nets

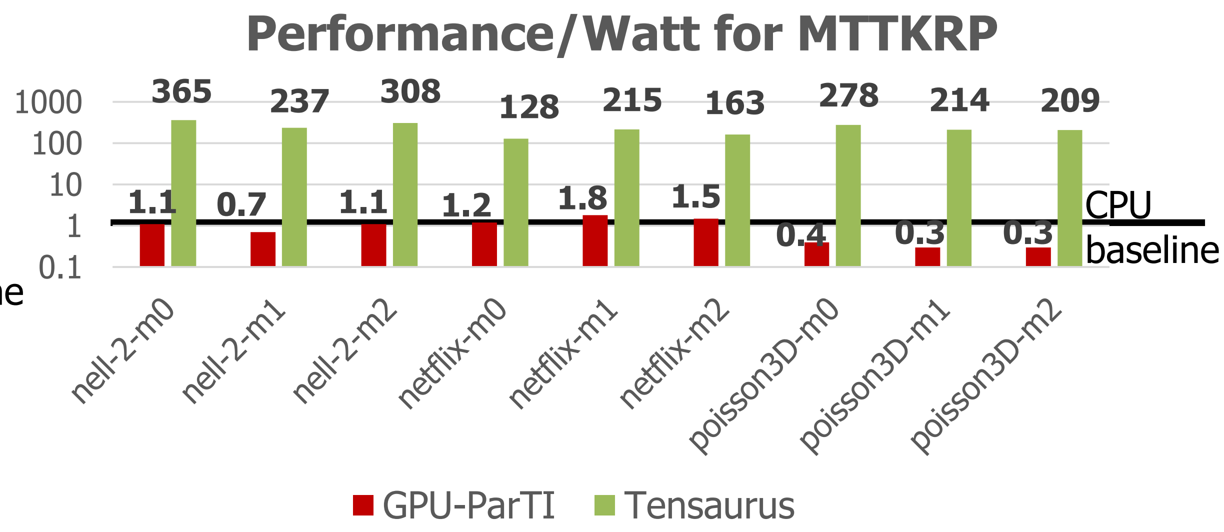




Overall Tensaurus is 1.9x faster and 1.7x more energy-efficient than Cambricon-X even for Sparse Neural Nets

Results on Sparse Tensor Decomposition





Tensaurus is 22.9x & 3.1x faster, and 220x & 290x more energy-efficient than CPU & GPU for MTTKRP

Concluding Remarks

- **Tensaurus: A versatile accelerator for sparse-dense tensor acceleration**
  - First accelerator for sparse tensor decompositions (MTTKRP, TTMc)
  - **Versatile:** NOT limited to tensor decompositions. Also efficient for sparse-dense matrix computations
  - **Adaptable:**
    - Also accelerates dense kernels
    - Easily adapts to different levels of sparsity found in various domains

- **Key Approach:** Co-design sparse format and architecture

- **Key Results:**
  - High bandwidth utilization (> 70% of peak bandwidth)
  - High speedup and energy efficiency compared to CPU, GPU and Cambricon-X

Thank you! Questions?

Tensaurus: A Versatile Accelerator for Mixed Sparse-Dense Tensor Computations

Nitish Srivastava, Hanchen Jin, Shaden Smith[2], Hongbo Rong[3],
David Albonesi, and Zhiru Zhang

Cornell University
[2]Microsoft AI & Research
[3]Intel Parallel Computing Lab