Hierarchical Computation and Communication for Distributed Sparse Operations on Supercomputers with Multi-GPU Nodes

Mert Hidayetoğlu

Coordinated Science Laboratory

University of Illinois at Urbana-Champaign

February 26, 2021

Preliminary Exam Committee:

Wen-mei Hwu (Chair)

Weng Cho Chew

Michale Oelze

Andreas Kloeckner

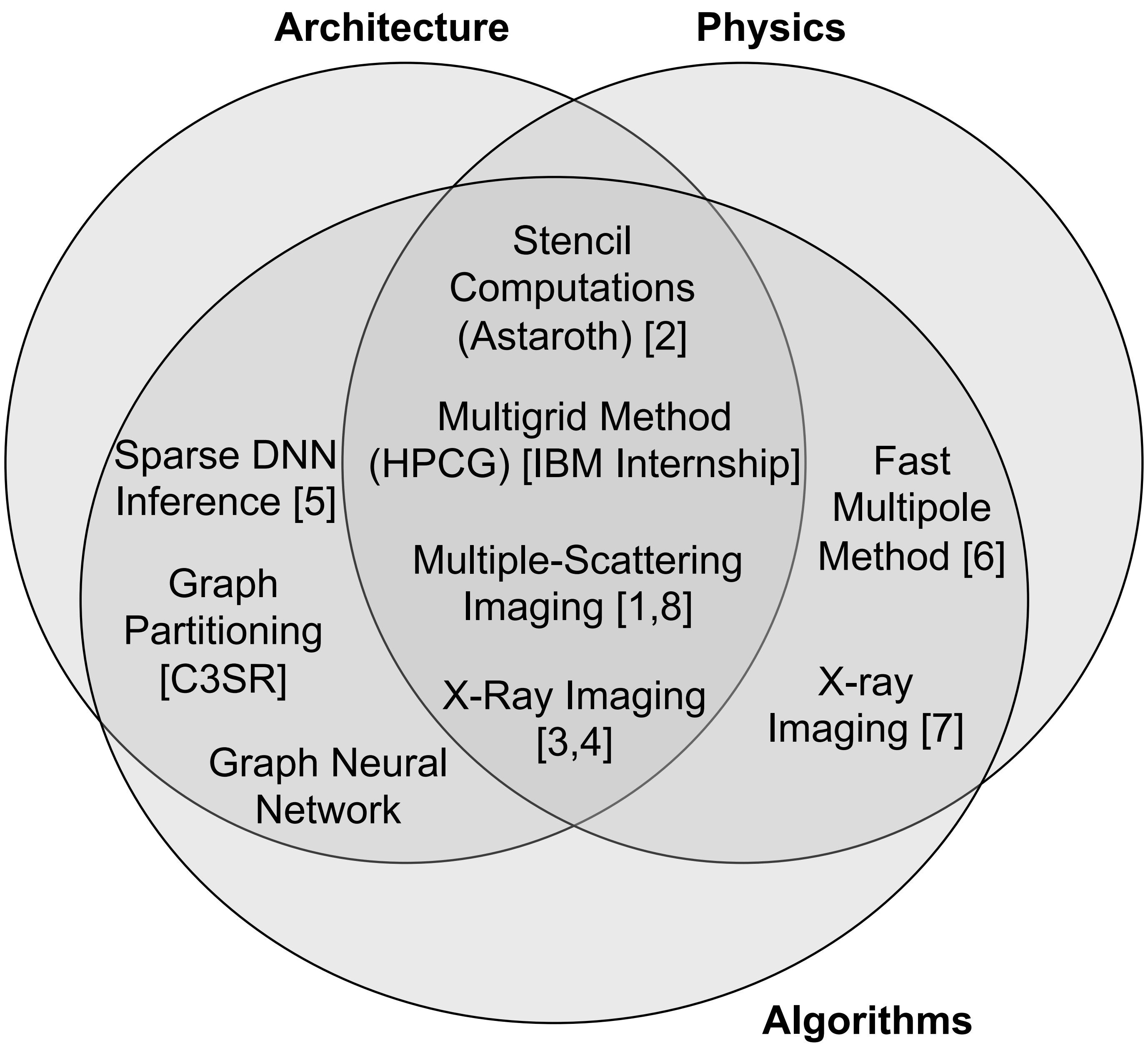
Bill Gropp

Large-Scale Sparse Applications

Conference
[1] IPDPS'18 (Outstanding Presentation)
[2] HPCC'19 (Best Paper)
[3] SC'19 (SCC Benchmark)
[4] SC'20 (Best Paper)
[5] HPEC'20 (Graph Challenge Champion)

Journal
[6] IEEE TAP (2018)
[7] OSA Applied Optics (2018)

Book Chapter
[8] New Trends in Computational Electromagnetics



Performance Bottlenecks

Single GPU

- Memory Capacity → Large-Scale
- Memory Bandwidth → Low Arithmetic Intensity; → On-chip Memory Underutilization
- Memory Latency → On-chip Memory Underutilization

GPU Cluster

- Bisection Bandwidth → Large-Scale
- Latency
- Load Balancing → Left to the Application Developer

Performance Bottlenecks

Single GP

GPU Clust

e

etic Intensity

mory

tion

ale

plication

Developer

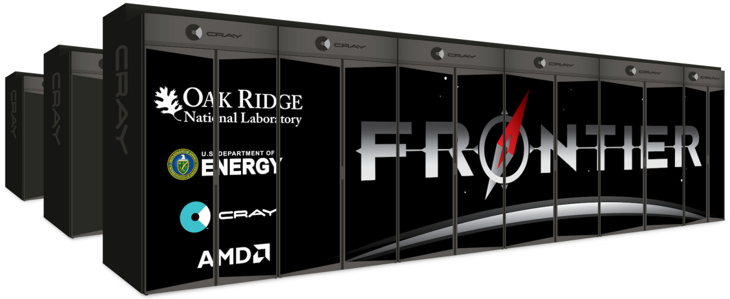
Summit (2018) 6 GPUs per Node

200 PFLOPS 4,608 Nodes

Aurora (2021) 6 GPUs per Node

1 EFLOPS

Frontier (2021) 4 GPUs per Node

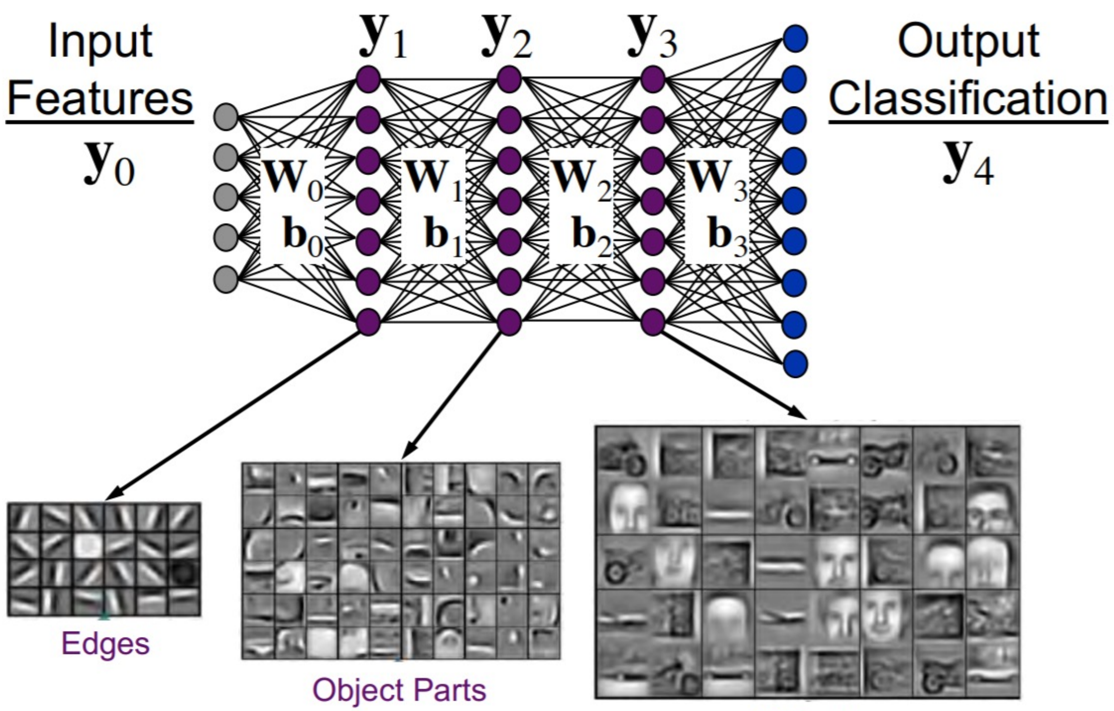


1.5 EFLOPS

El Capitan (2024) 4 GPUs per Node

2 EFLOPS

Application: Sparse Deep Neural Network



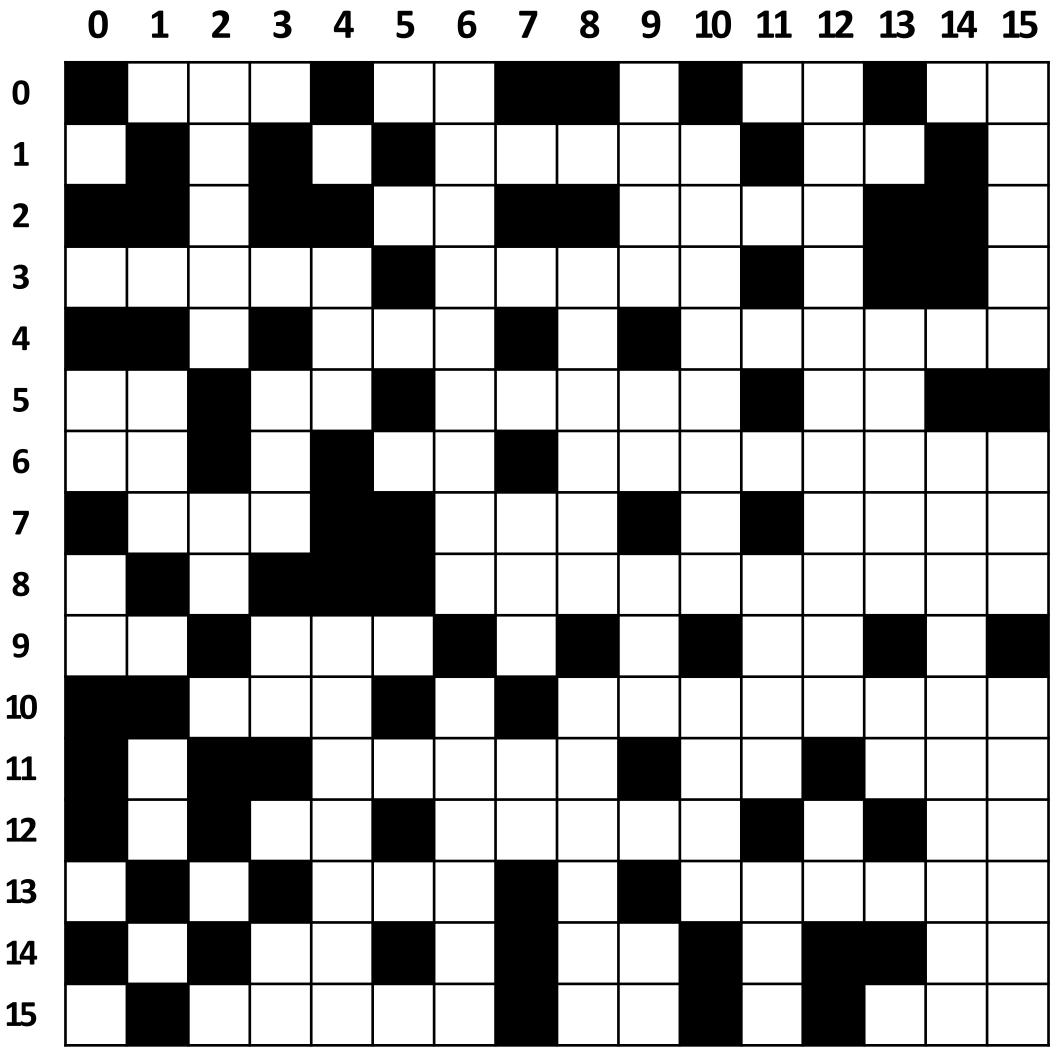
Input Data (Y0)
MNIST corpus of 60,000 handwritten numbers.
- 32x32 → **1,024** neurons
- 64x64 → **4,096** neurons
- 128x128 → **16,384** neurons
- 256x256 → **65,536** neurons

Neurons/Layer

Layers	1024	4096	16384	65536
120	3,932,160	15,728,640	62,914,560	251,658,240
480	15,728,640	62,914,560	251,658,240	1,006,632,960
1920	62,914,560	251,658,240	1,006,632,960	4,026,531,840

Total Edges = 32 x layers x neurons/layer

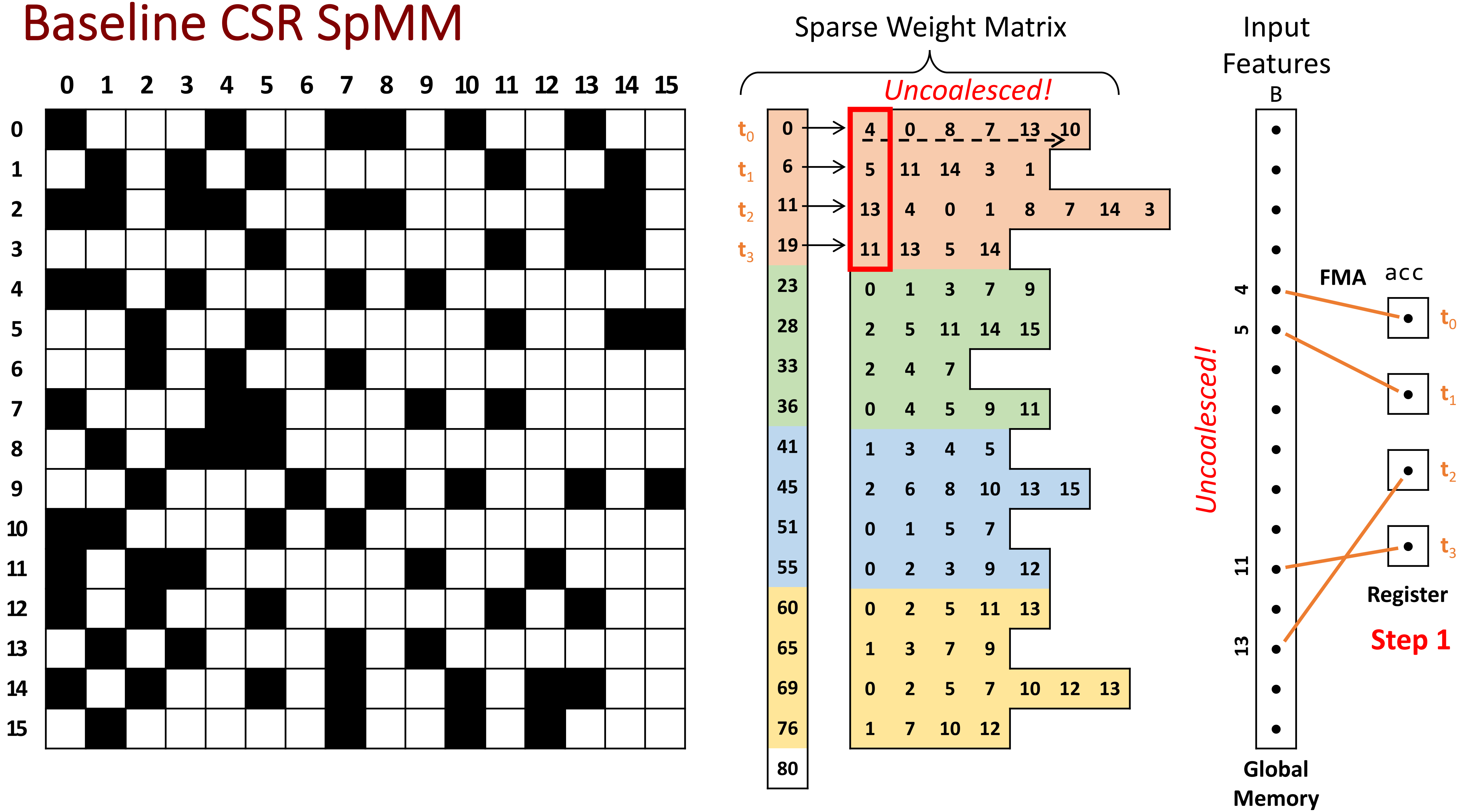
Baseline CSR SpMM



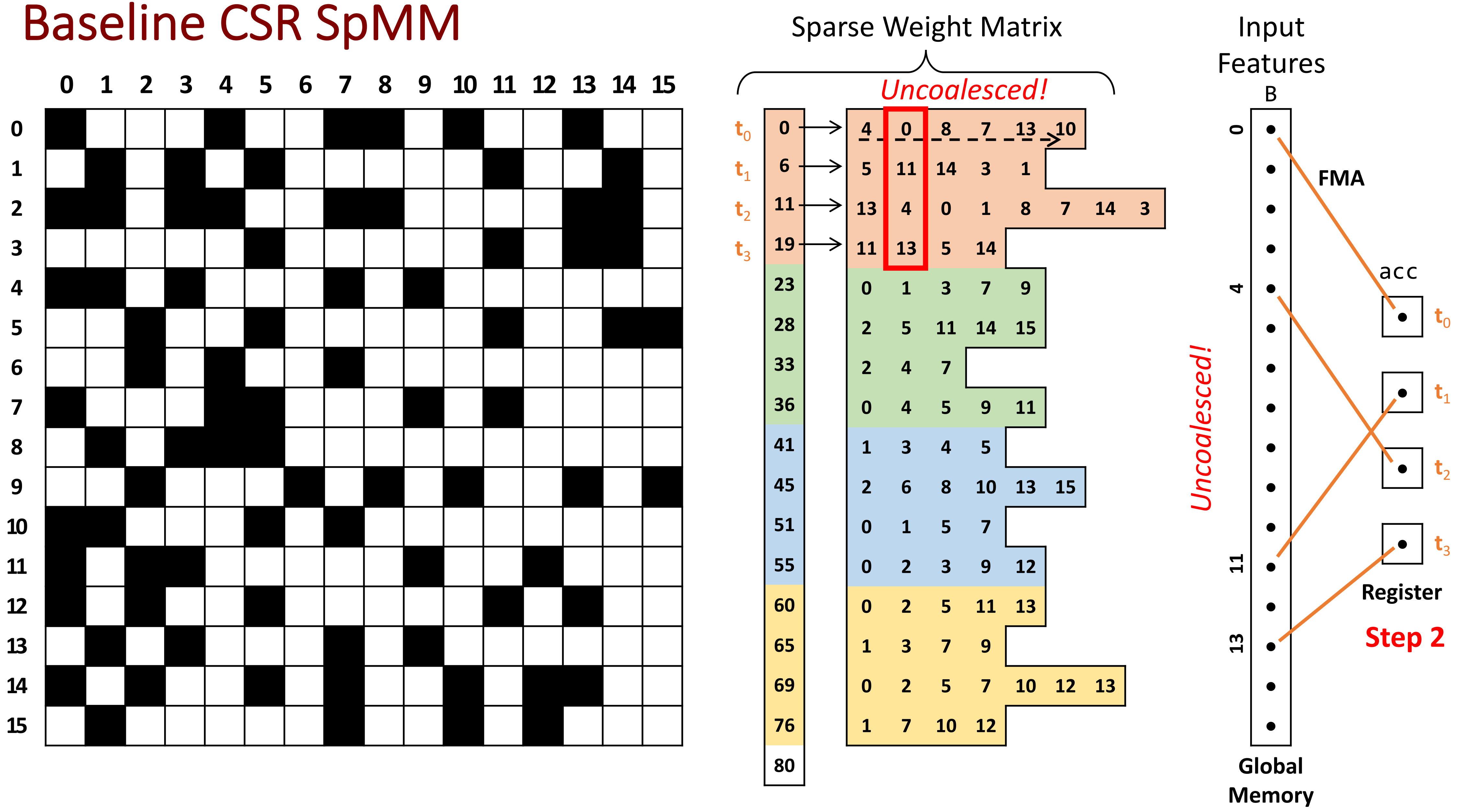
Sparse Weight Matrix

t_0	0	4	0	8	7	13	10		
t_1	6	5	11	14	3	1			
t_2	11	13	4	0	1	8	7	14	3
t_3	19	11	13	5	14				
	23	0	1	3	7	9			
	28	2	5	11	14	15			
	33	2	4	7					
	36	0	4	5	9	11			
	41	1	3	4	5				
	45	2	6	8	10	13	15		
	51	0	1	5	7				
	55	0	2	3	9	12			
	60	0	2	5	11	13			
	65	1	3	7	9				
	69	0	2	5	7	10	12	13	
	76	1	7	10	12				
	80								

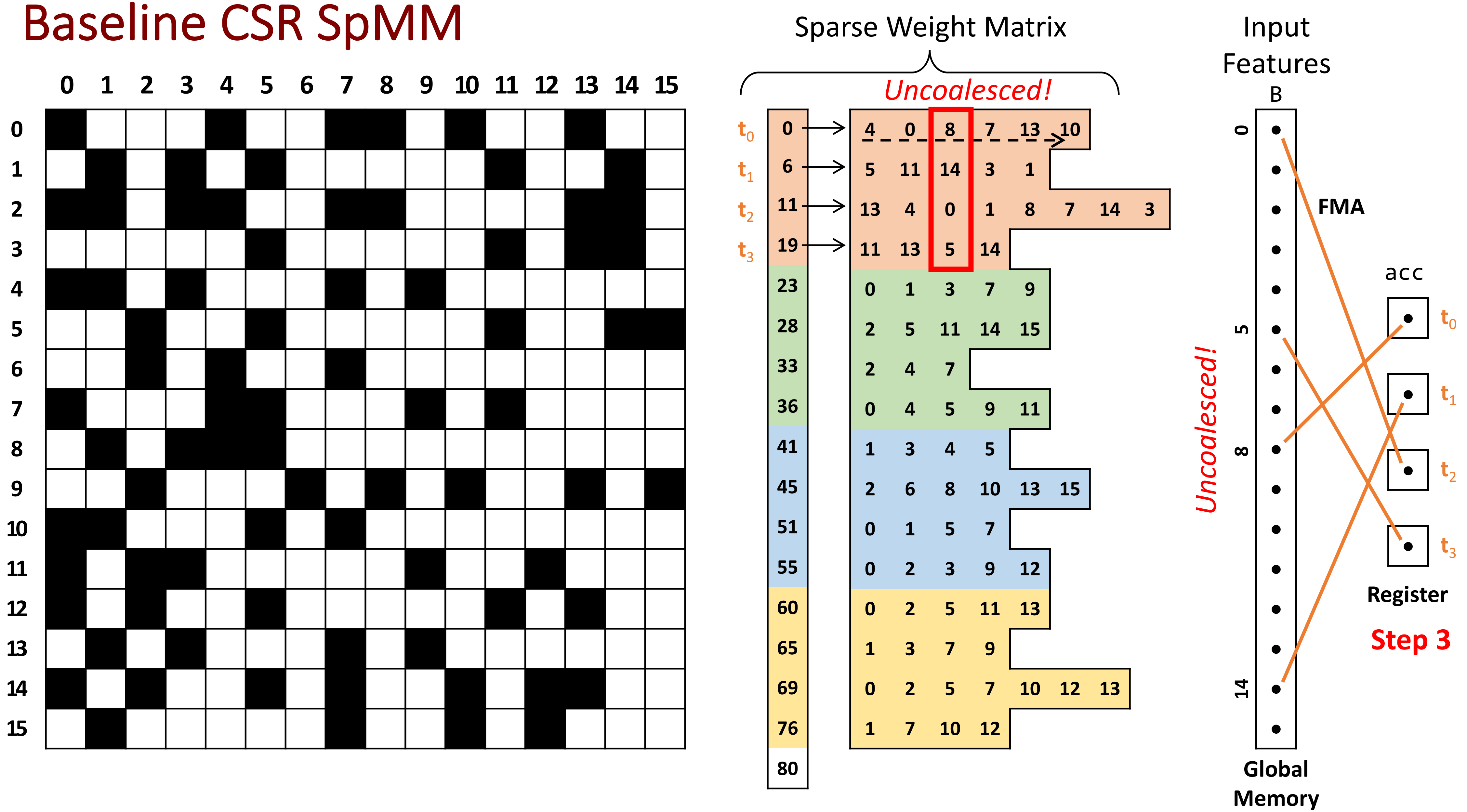
Baseline CSR SpMM



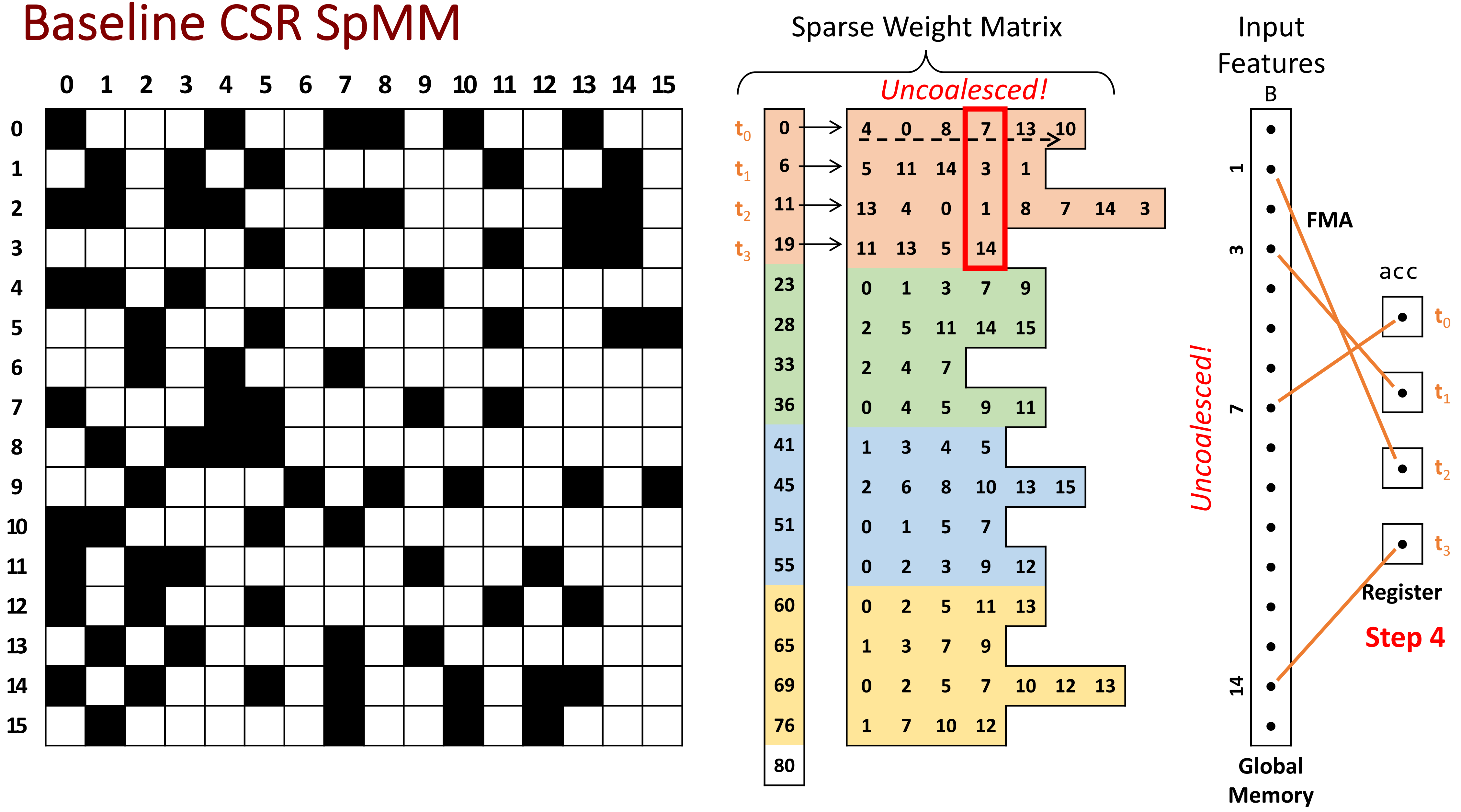
Baseline CSR SpMM



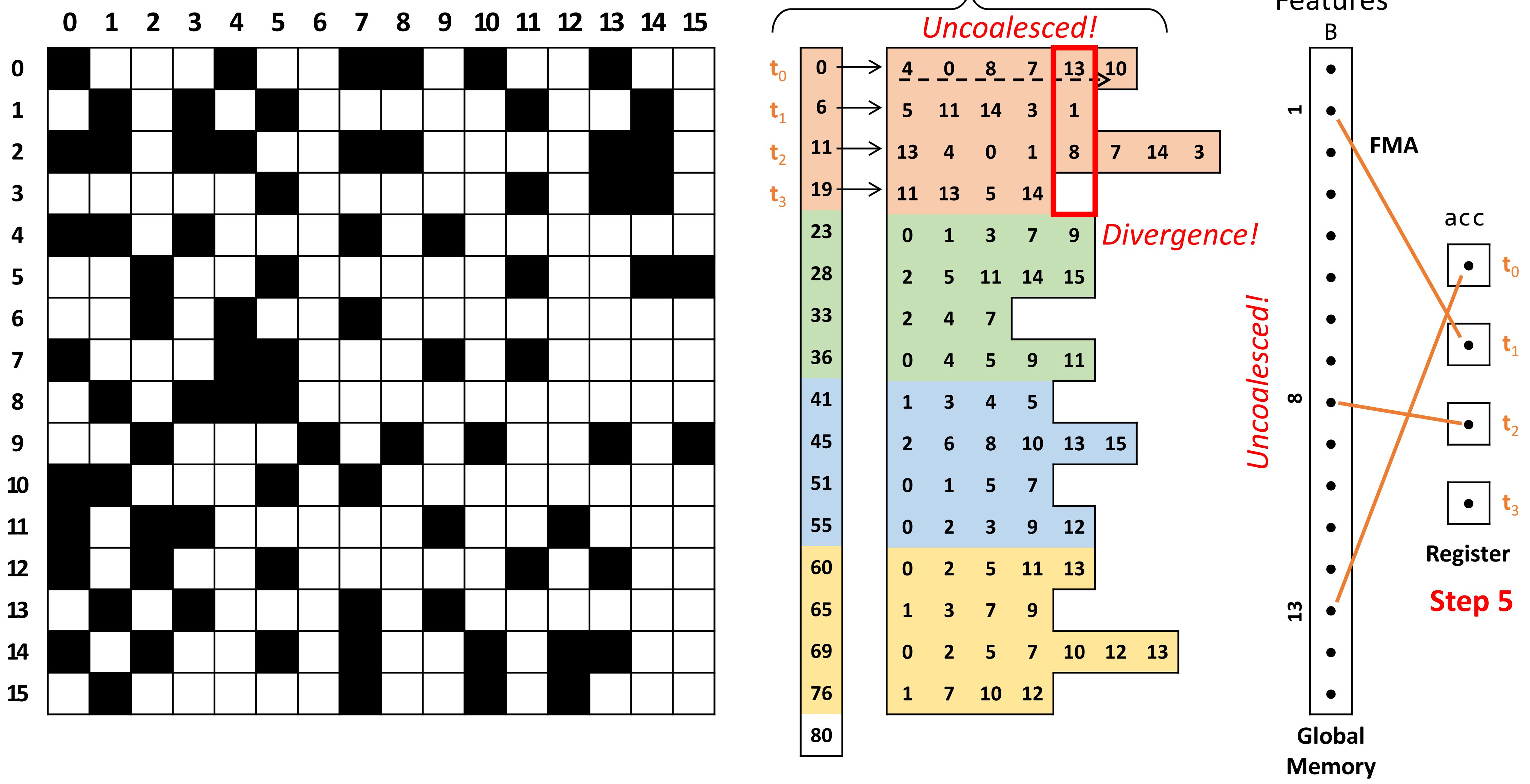
Baseline CSR SpMM



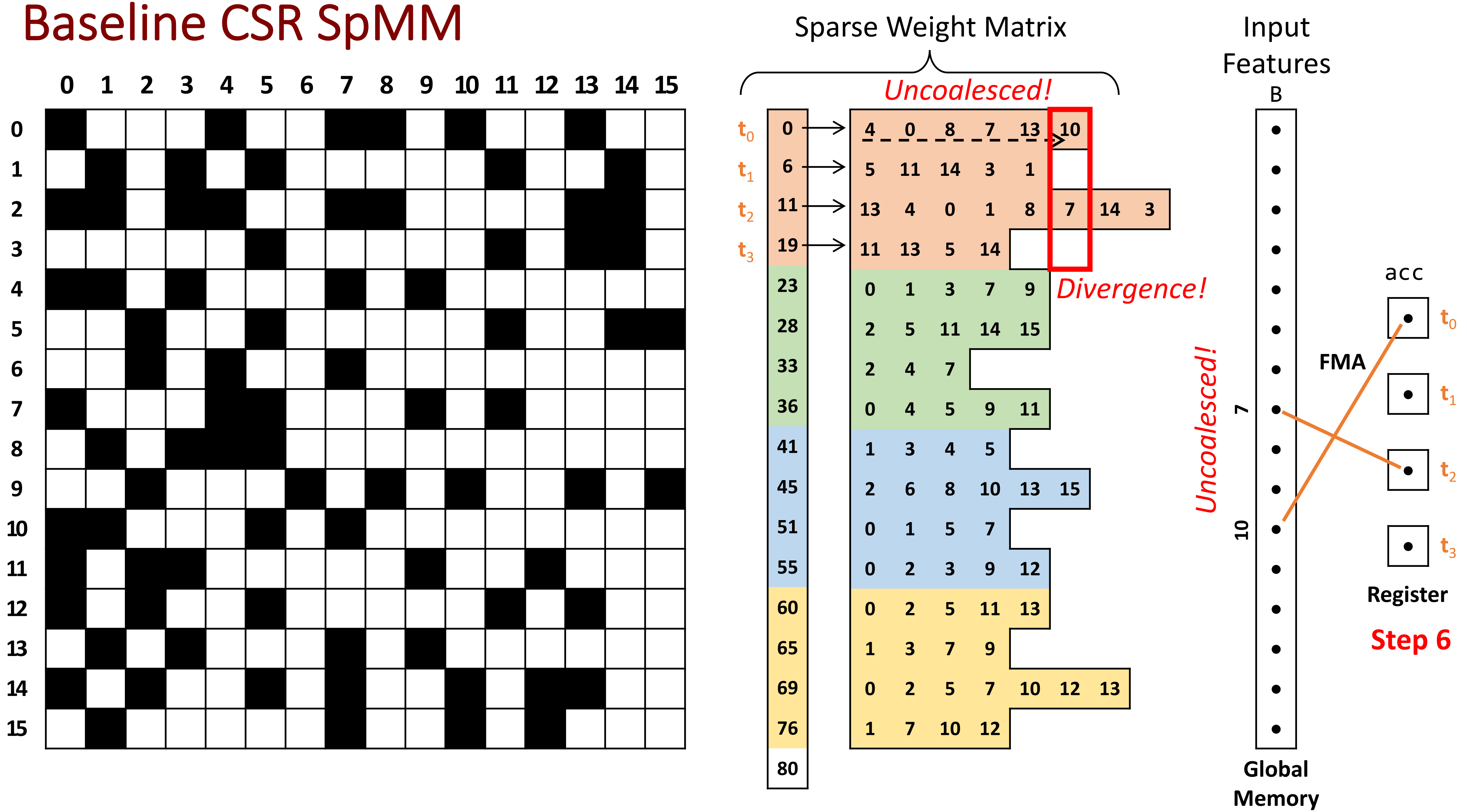
Baseline CSR SpMM



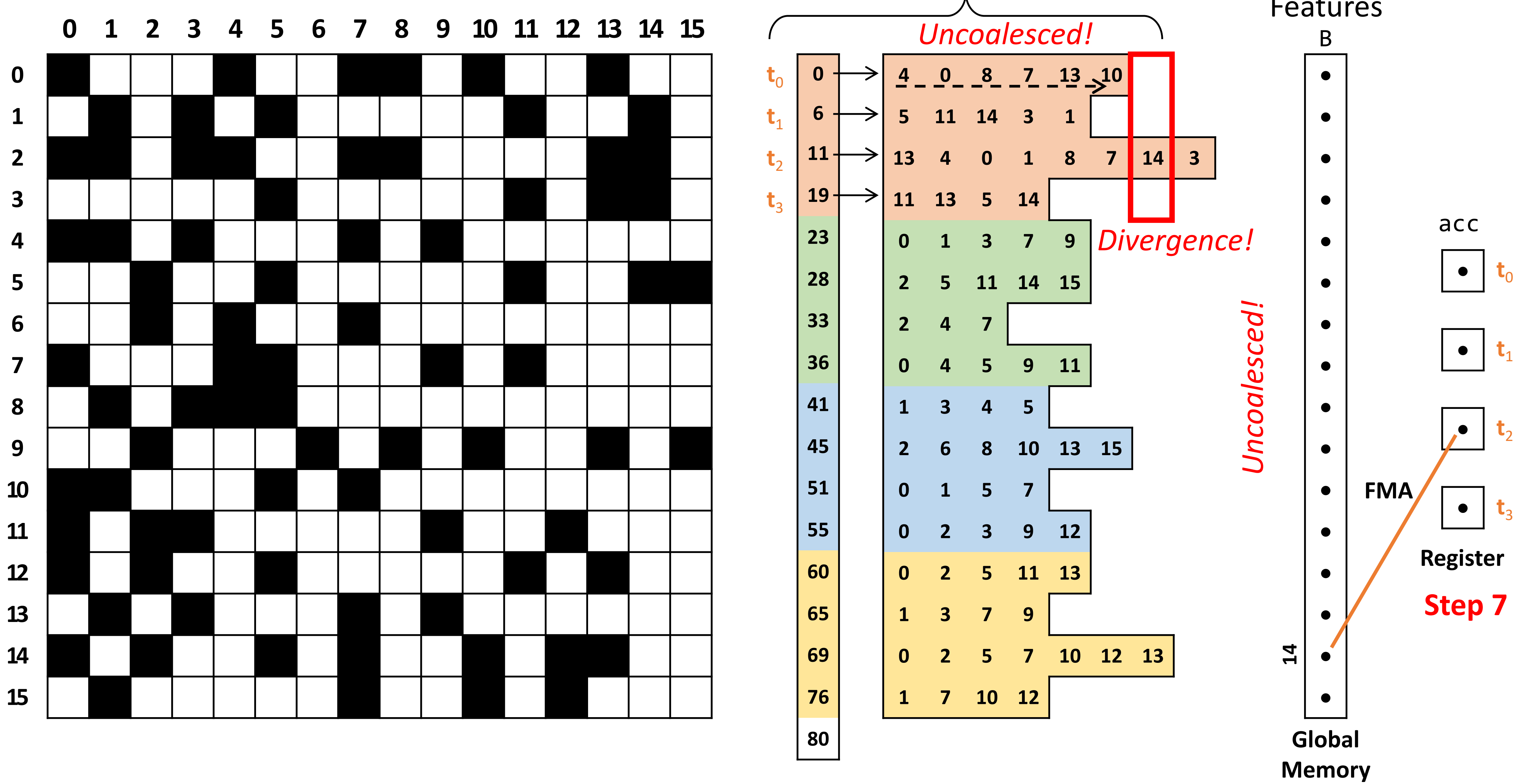
Baseline CSR SpMM



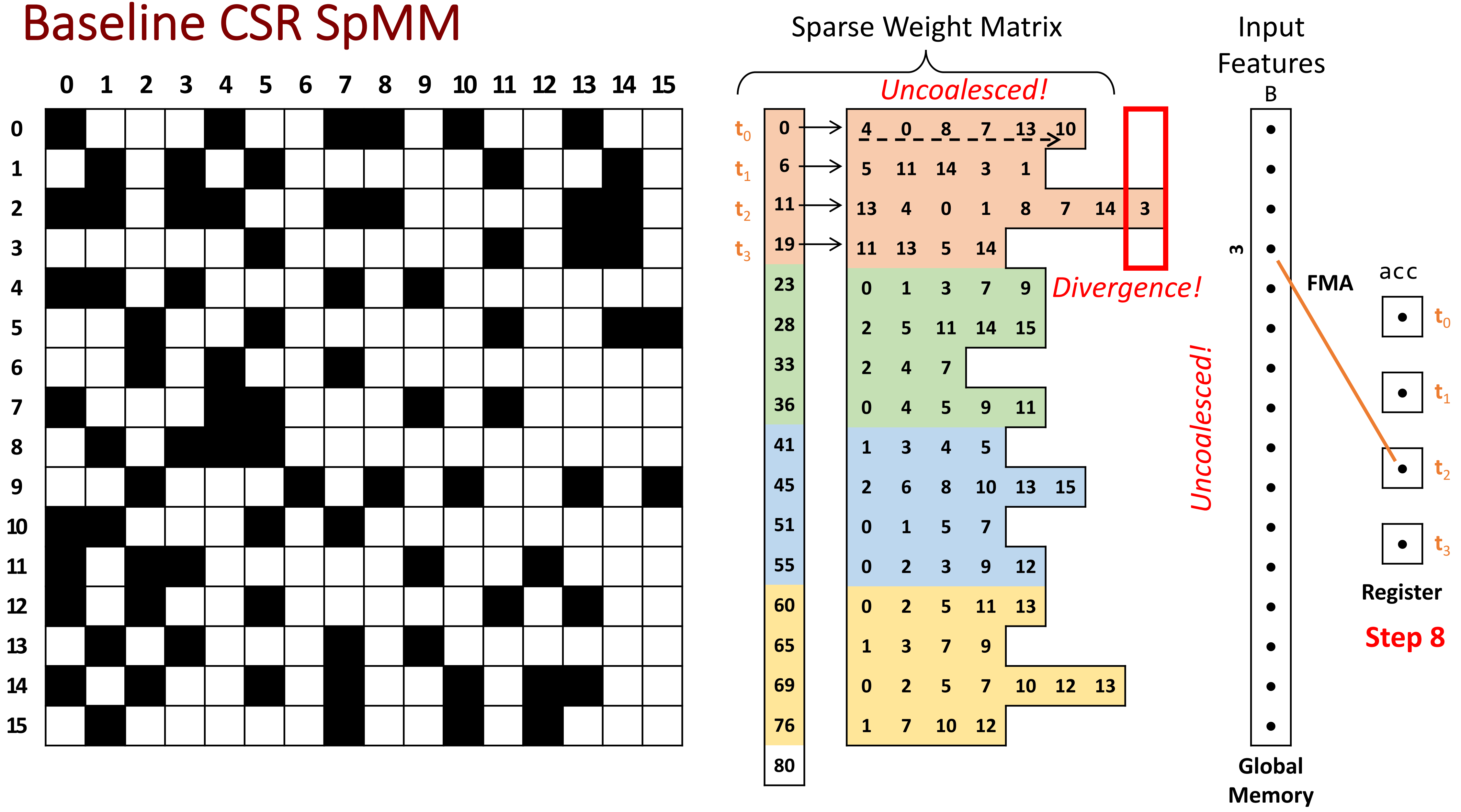
Baseline CSR SpMM



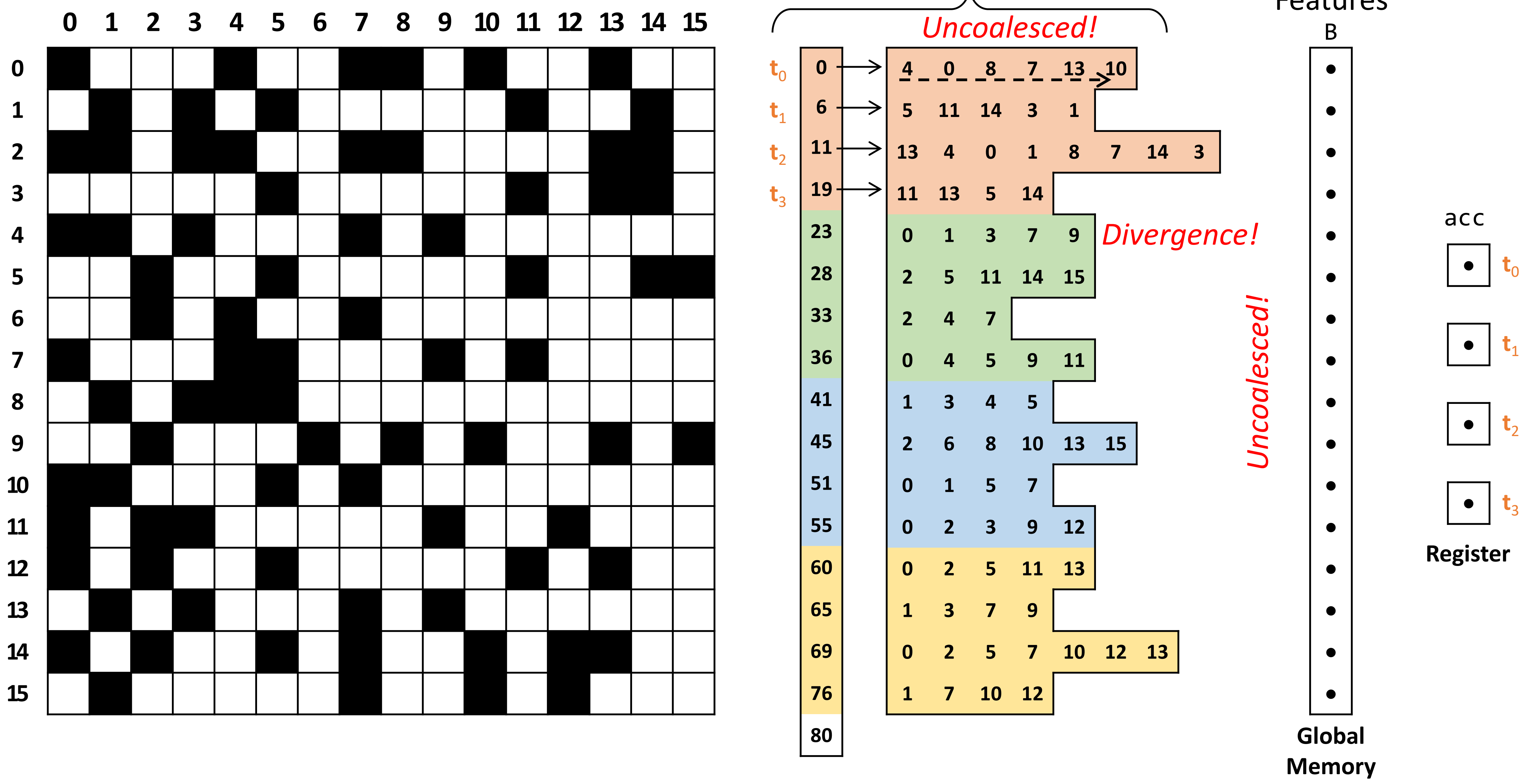
Baseline CSR SpMM



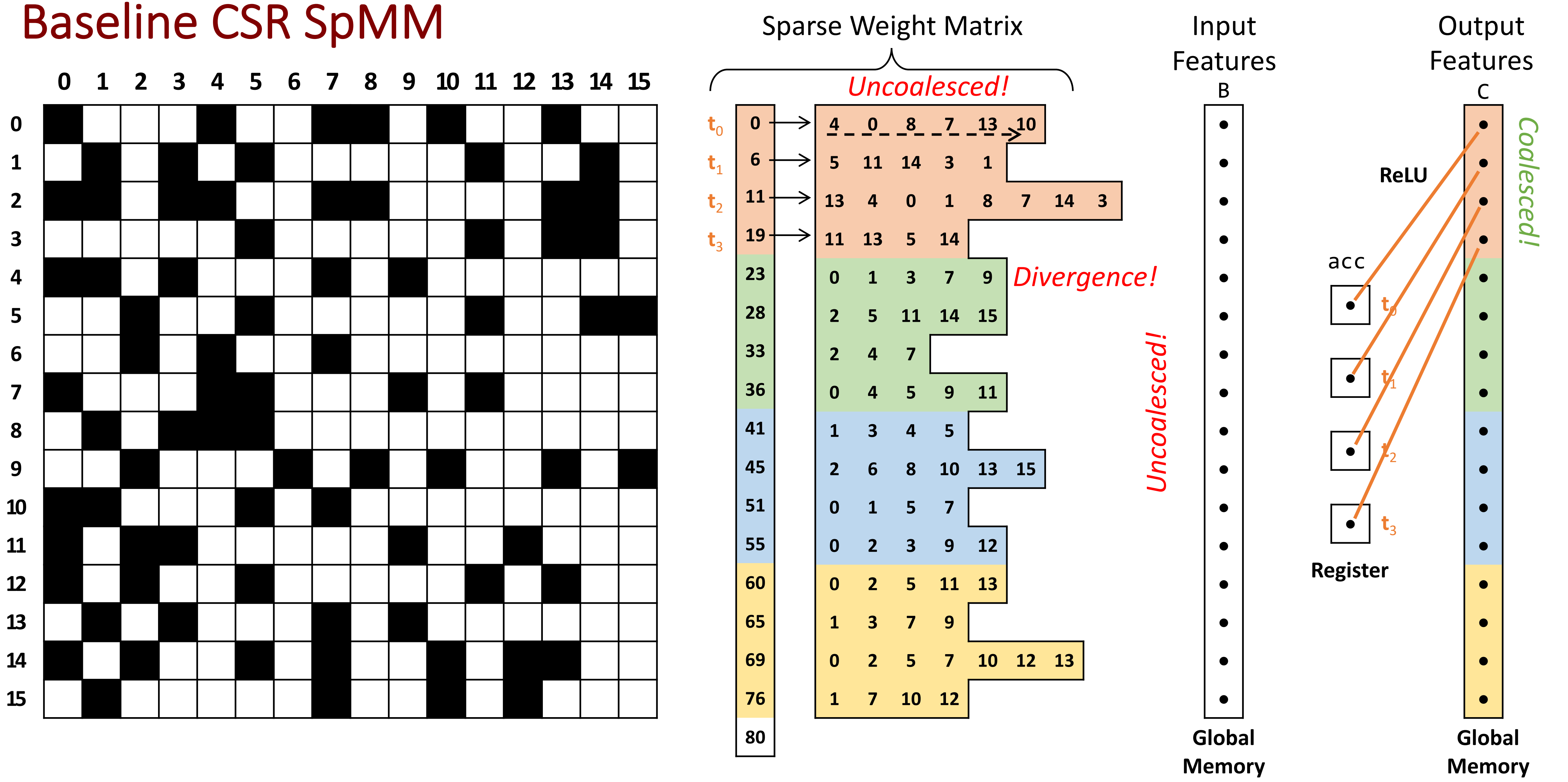
Baseline CSR SpMM



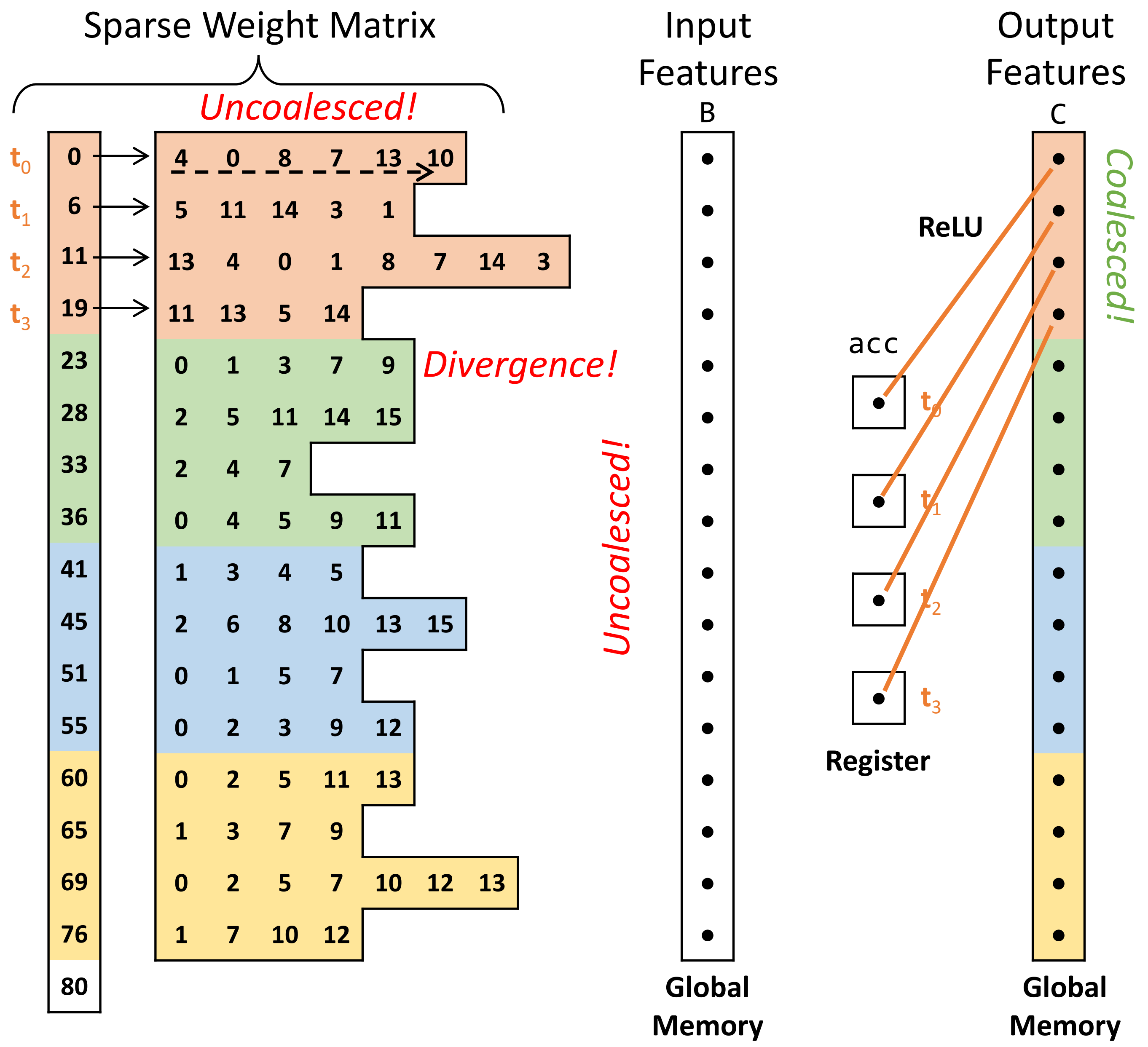
Baseline CSR SpMM



Baseline CSR SpMM



Baseline CSR SpMM



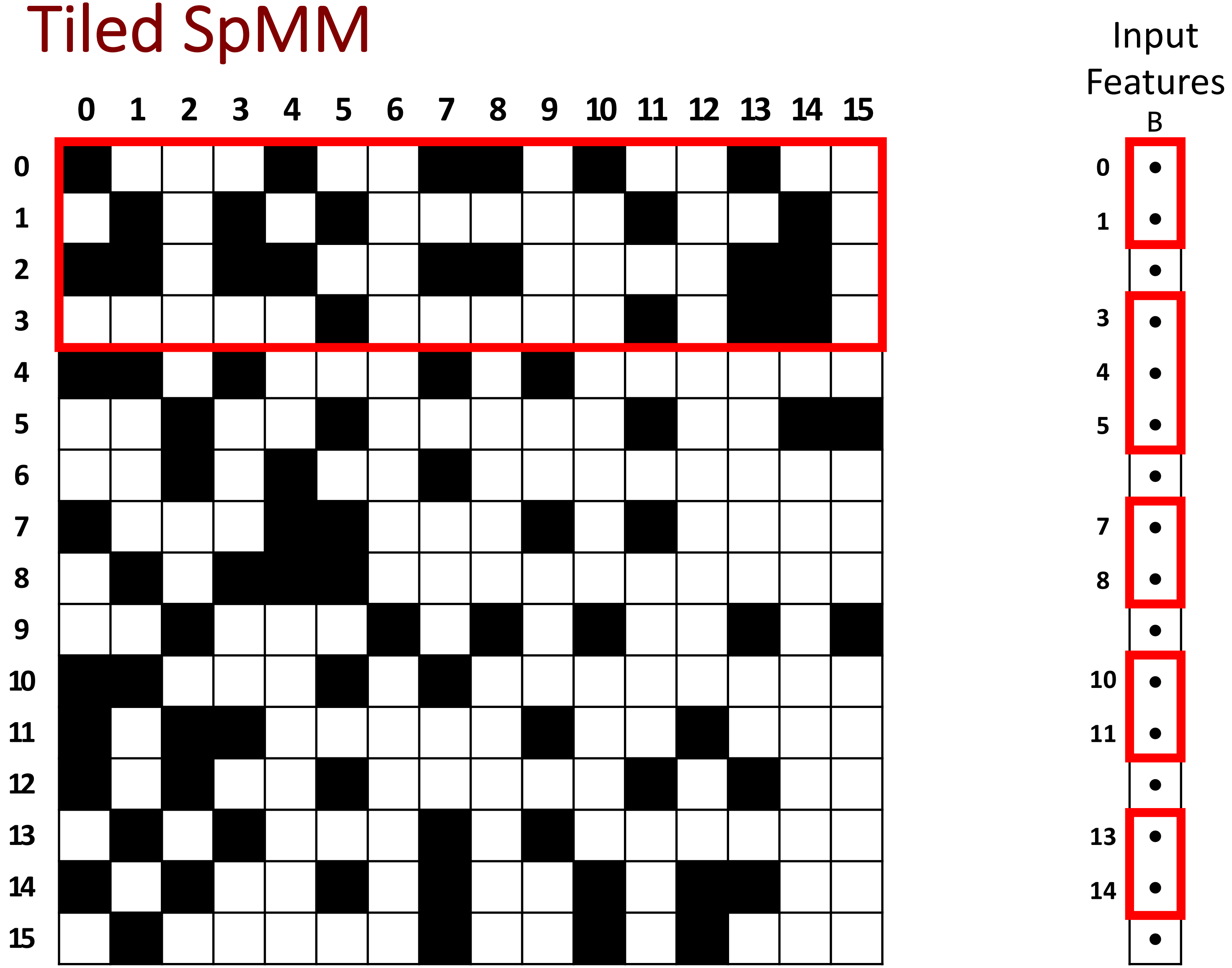
Data Access Redundancies

- Input features by different threads
- Weight matrix by all output features

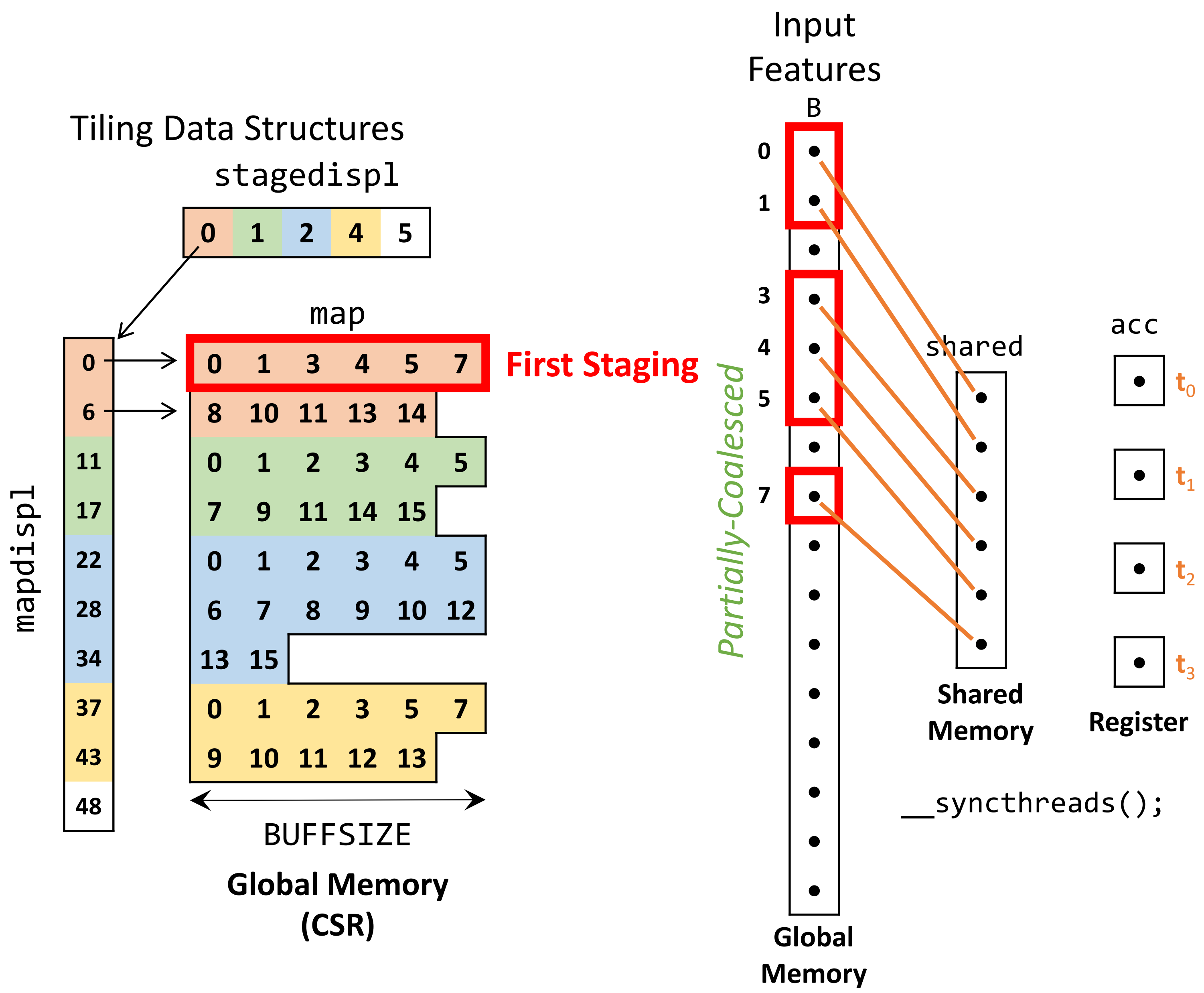
Data Access Latencies

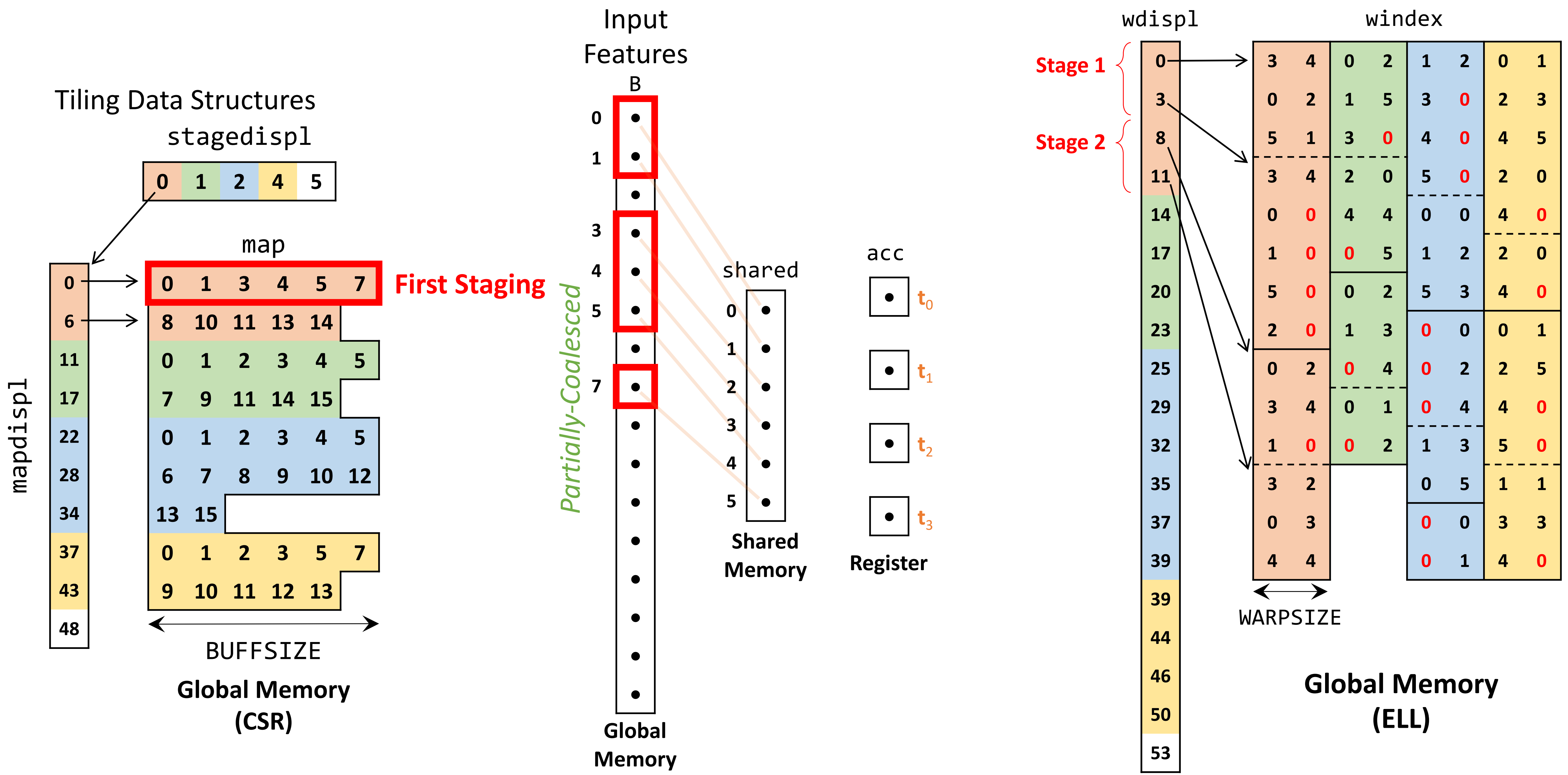
- Irregular access to input features
- Uncoalesced access to weight matrix

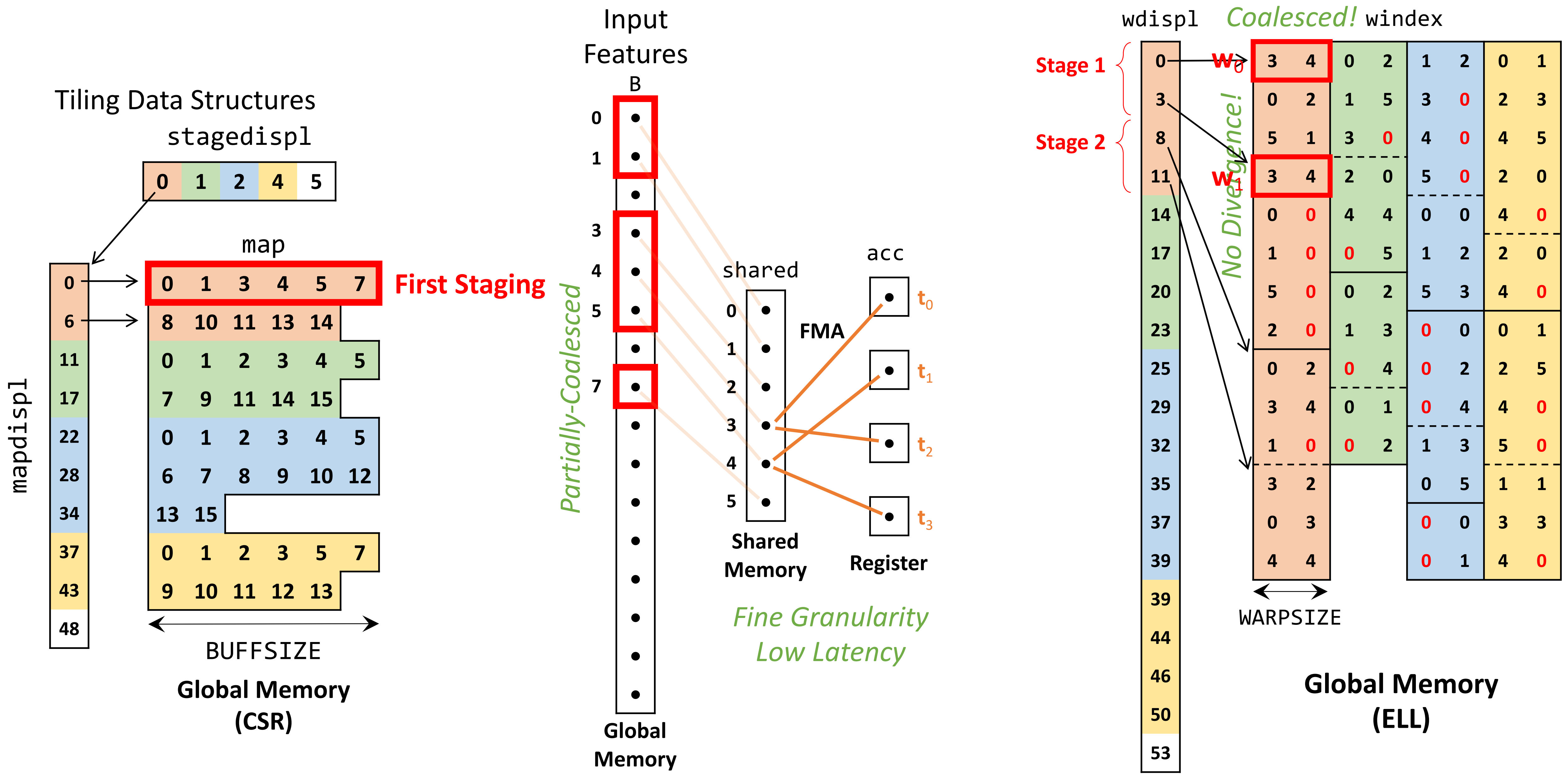
Tiled SpMM

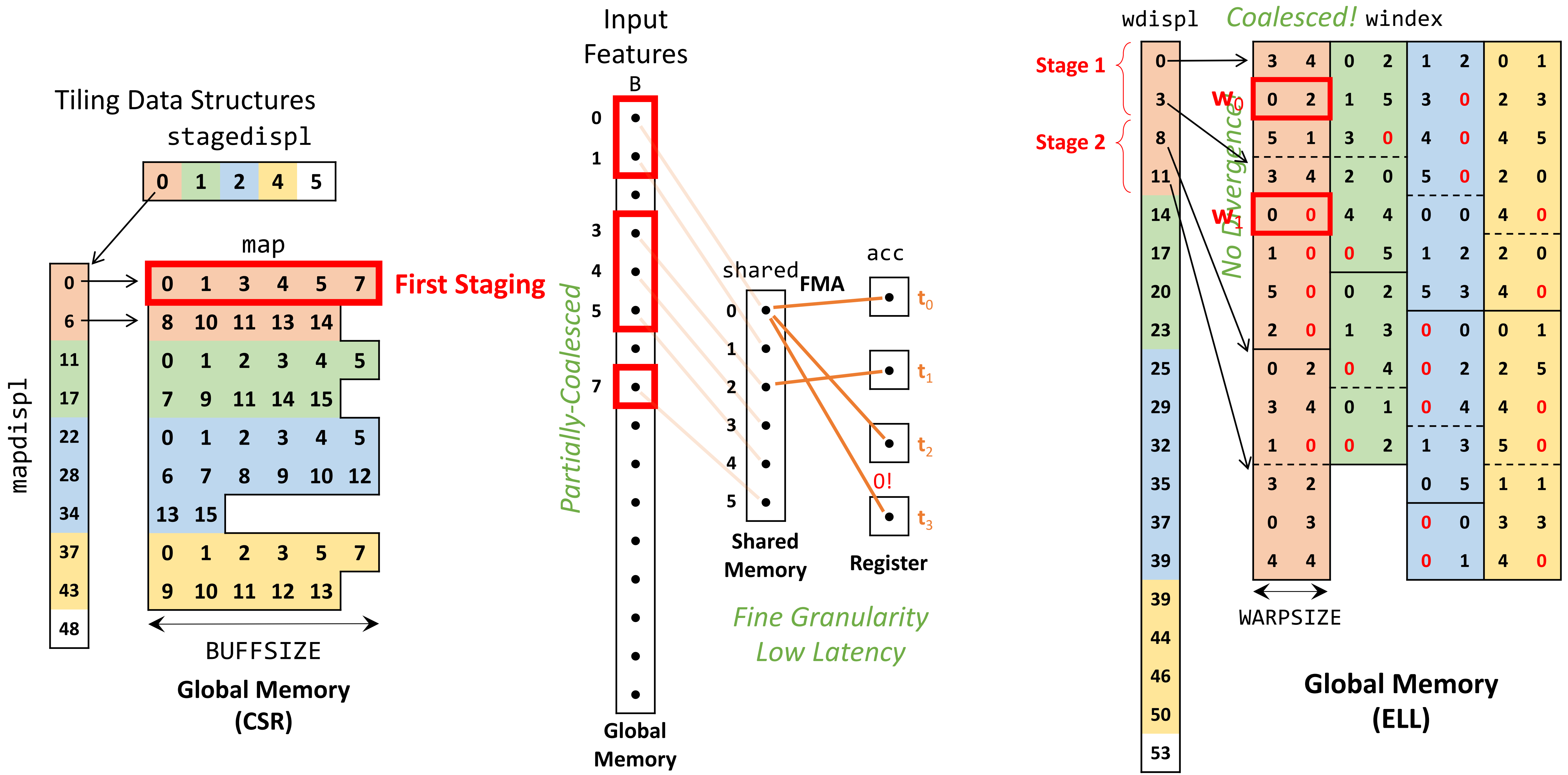


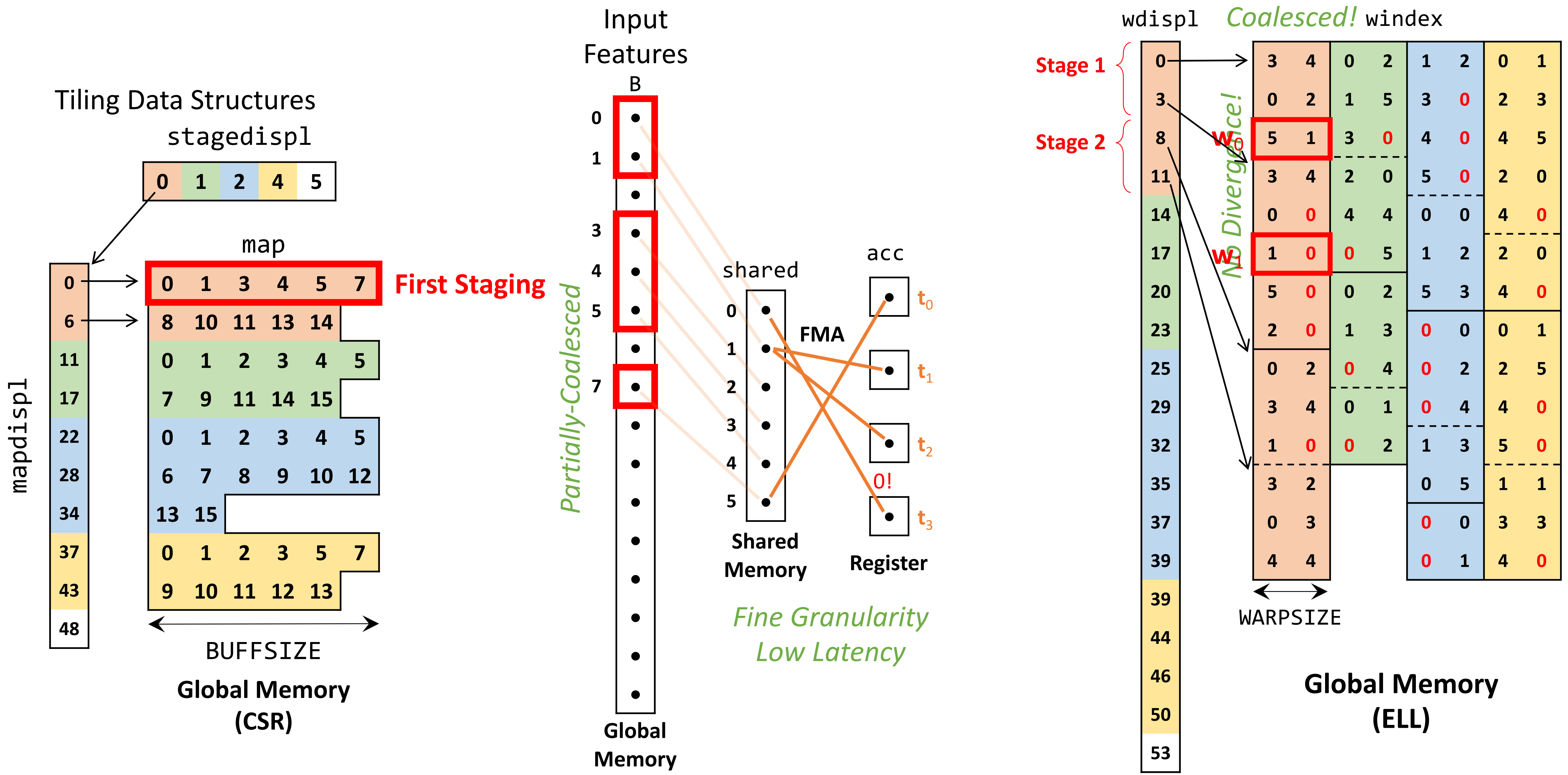
Input
Features
B
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 3 4 5 7 8 10 11 13 14

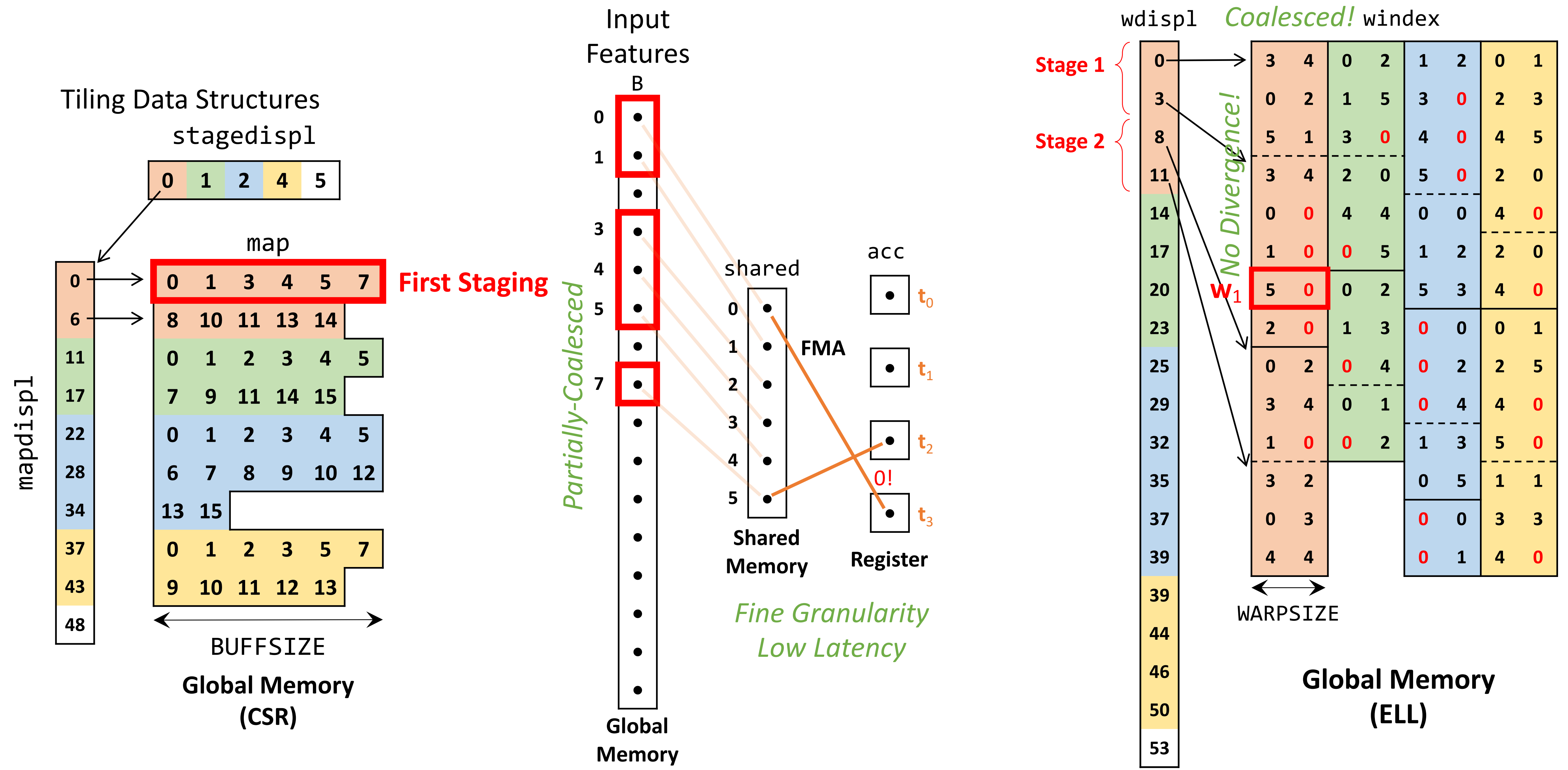

Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0
6
11
17
22
28
34
37
43
48
0 1 3 4 5 7
First Staging
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
0
1
3
4
5
7
Partially-Coalesced
Global
Memory
shared
Shared
Memory
acc
t_0
t_1
t_2
t_3
Register
__syncthreads();


Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0 6 11 17 22 28 34 37 43 48
0 1 3 4 5 7
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
First Staging
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
0 1 3 4 5 7
Partially-Coalesced
Global
Memory
shared
0 1 2 3 4 5
Shared
Memory
acc
t_0
t_1
t_2
t_3
Register
wdispl
windex
Stage 1
Stage 2
0 3 8 11 14 17 20 23 25 29 32 35 37 39 39 44 46 50 53
WARPSIZE
Global Memory
(ELL)

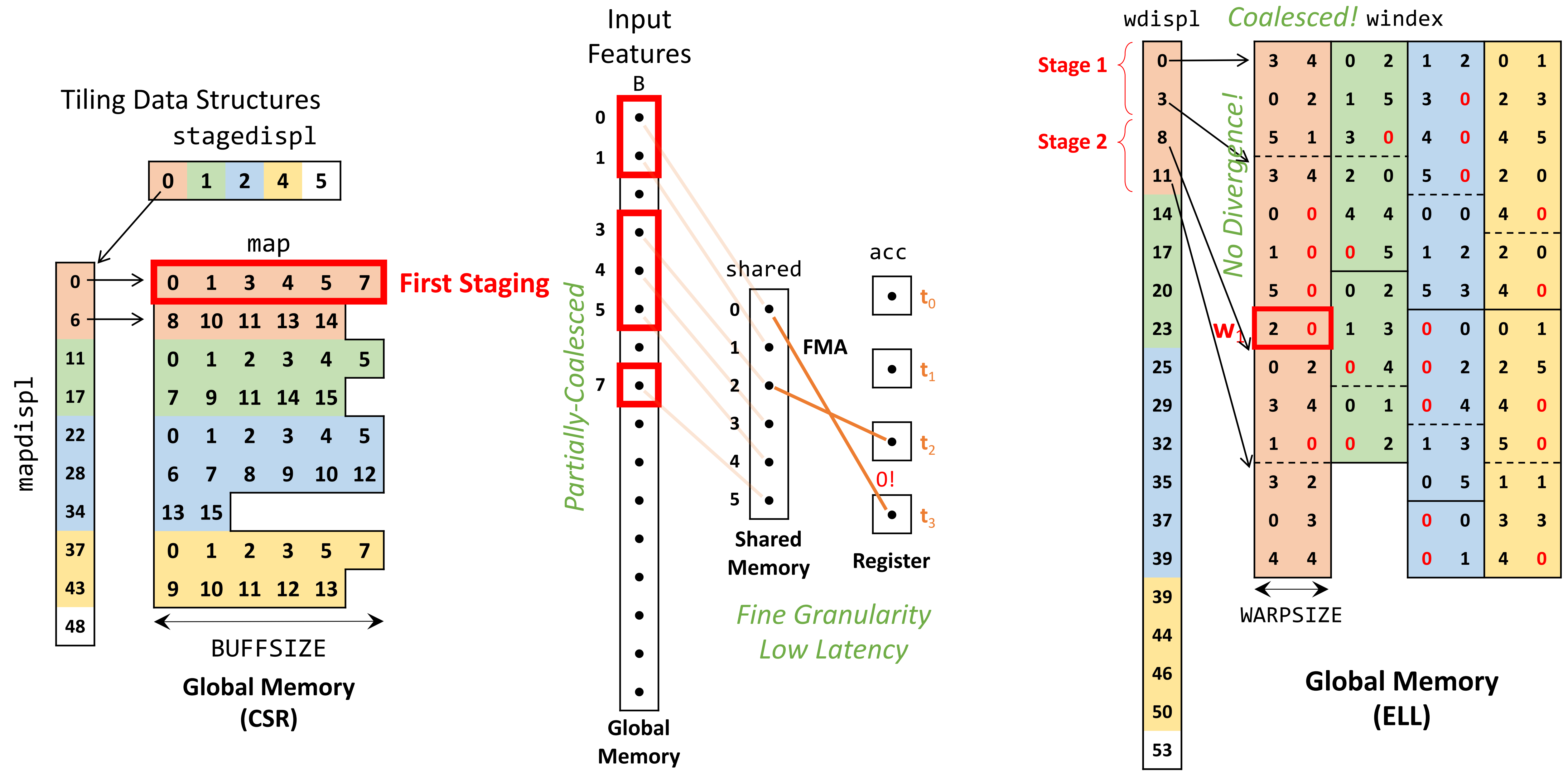

Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0 6 11 17 22 28 34 37 43 48
0 1 3 4 5 7
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
First Staging
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
0
1
3
4
5
7
Partially-Coalesced
Global
Memory
shared
0
1
2
3
4
5
FMA
acc
Shared
Memory
Register
Fine Granularity
Low Latency
wdispl
Coalesced!
windex
Stage 1
Stage 2
No Divergence!
0 3 8 11 14 17 20 23 25 29 32 35 37 39 39 44 46 50 53
WARPSIZE
Global Memory
(ELL)


Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0 6 11 17 22 28 34 37 43 48
0 1 3 4 5 7
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
First Staging
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
Partially-Coalesced
Global
Memory
shared
FMA
acc
Shared
Memory
Register
0!
Fine Granularity
Low Latency
wdispl
Coalesced!
windex
Stage 1
Stage 2
No Divergence
WARPSIZE
Global Memory
(ELL)

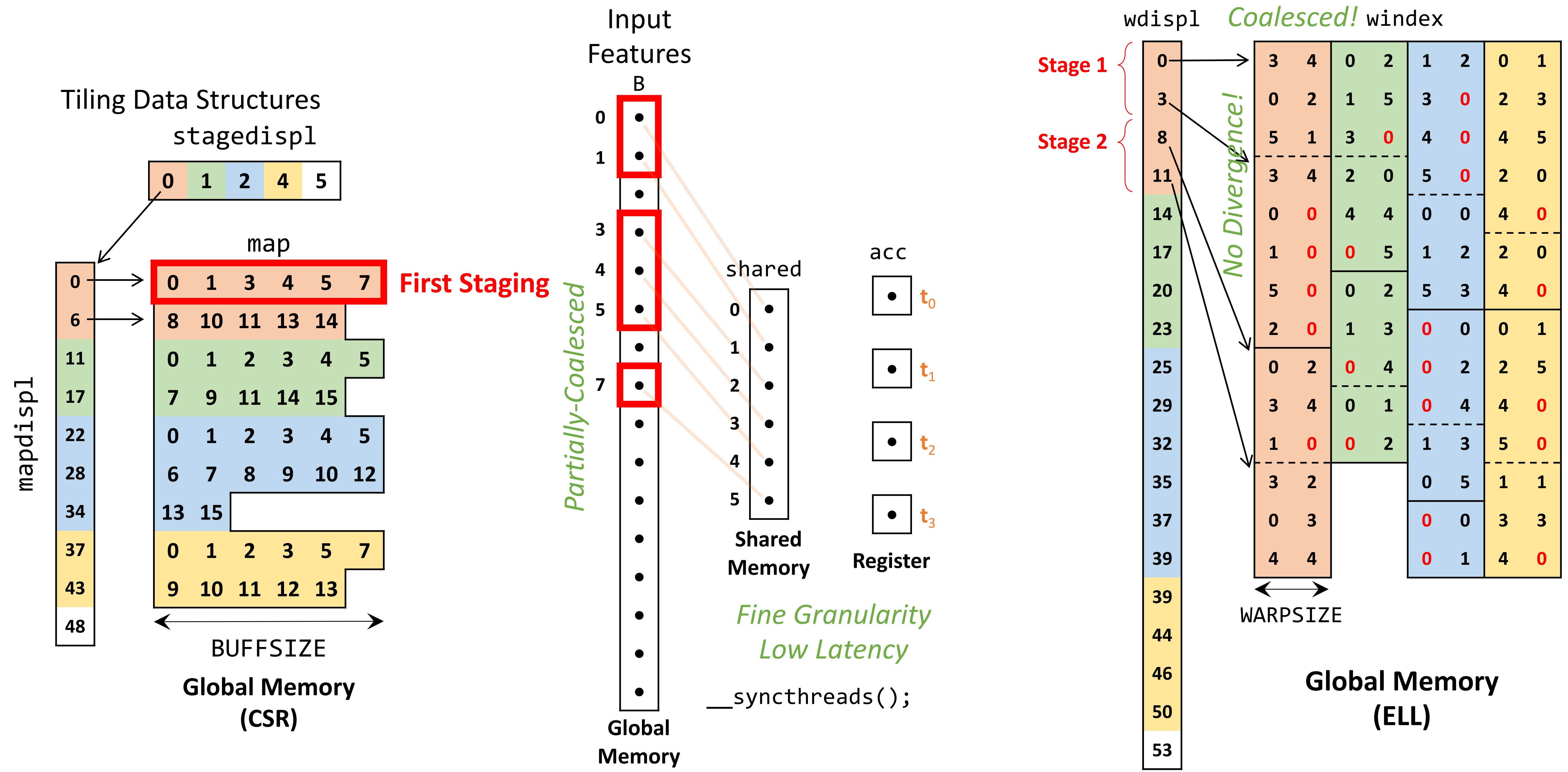

Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
First Staging
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
Partially-Coalesced
Global
Memory
shared
acc
FMA
0!
Shared
Memory
Register
Fine Granularity
Low Latency
wdispl
Coalesced!
windex
Stage 1
Stage 2
No Divergence!
WARPSIZE
Global Memory
(ELL)

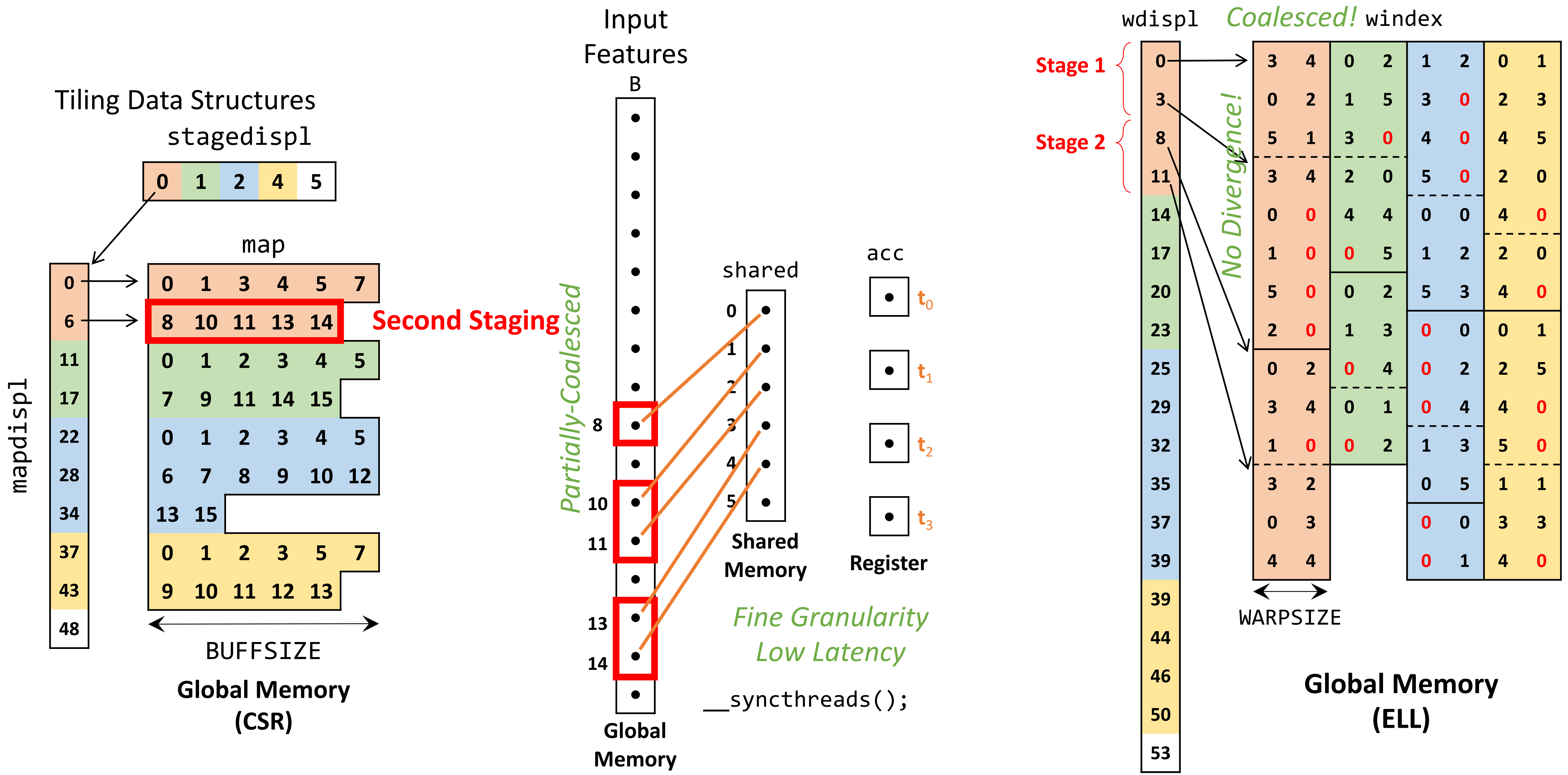


Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0 6 11 17 22 28 34 37 43 48
0 1 3 4 5 7
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
First Staging
BUFFSIZE
Global Memory (CSR)
Input Features
B
Partially-Coalesced
Global Memory
shared
Shared Memory
FMA
acc
t_0
t_1
t_2
t_3
0!
Register
Fine Granularity
Low Latency
wdispl
Coalesced!
windex
Stage 1
Stage 2
No Divergence!
W_1
WARPSIZE
Global Memory (ELL)

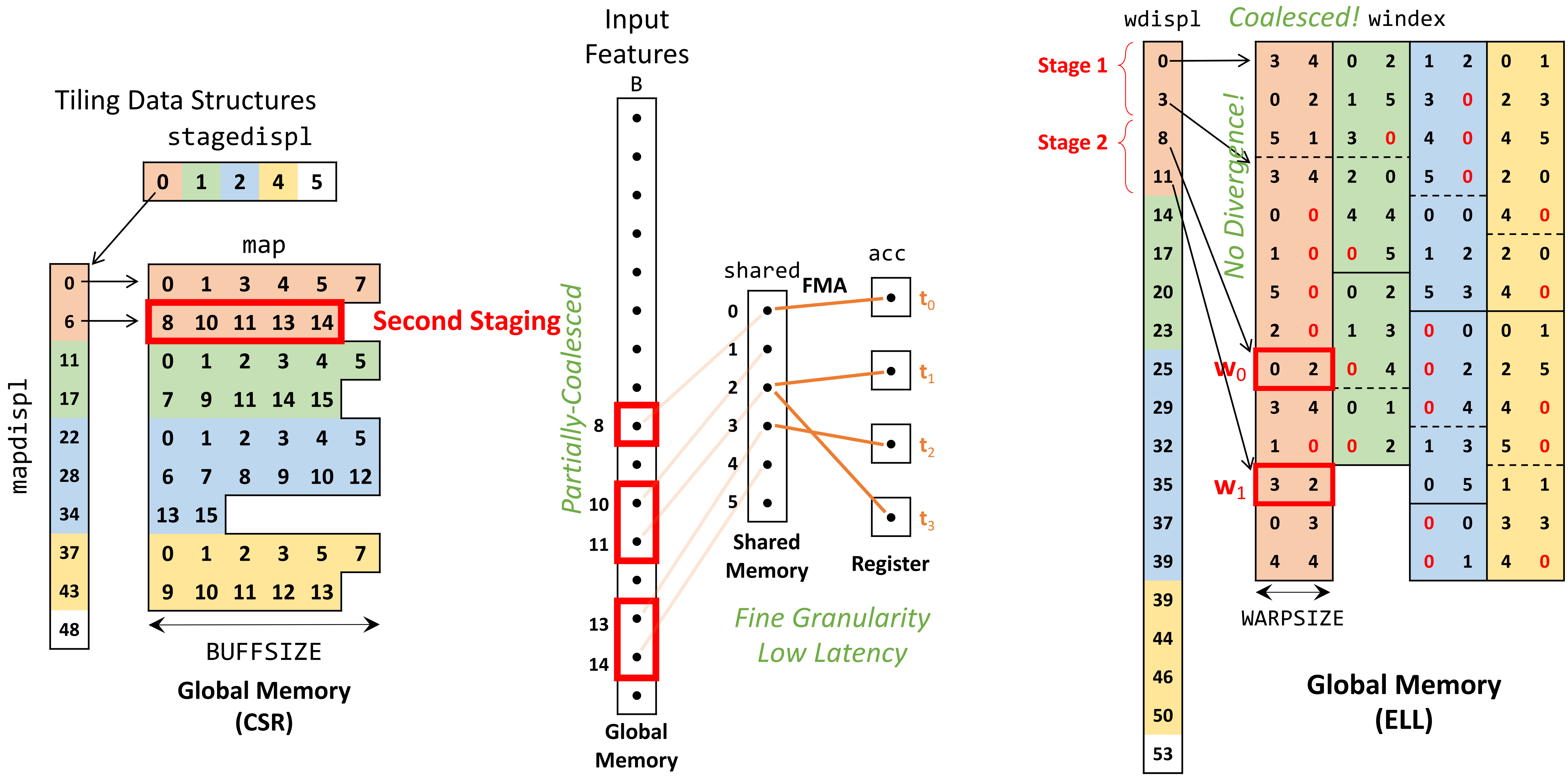


Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0 6 11 17 22 28 34 37 43 48
0 1 3 4 5 7
First Staging
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
BUFFSIZE
Global Memory
(CSR)
Input Features
B
Partially-Coalesced
Global Memory
shared
FMA
acc
t_0
t_1
t_2
t_3
0!
Shared Memory
Register
Fine Granularity
Low Latency
wdispl
Coalesced! windex
Stage 1
Stage 2
No Divergence!
w_1
0 3 8 11 14 17 20 23 25 29 32 35 37 39 39 44 46 50 53
WARPSIZE
Global Memory
(ELL)

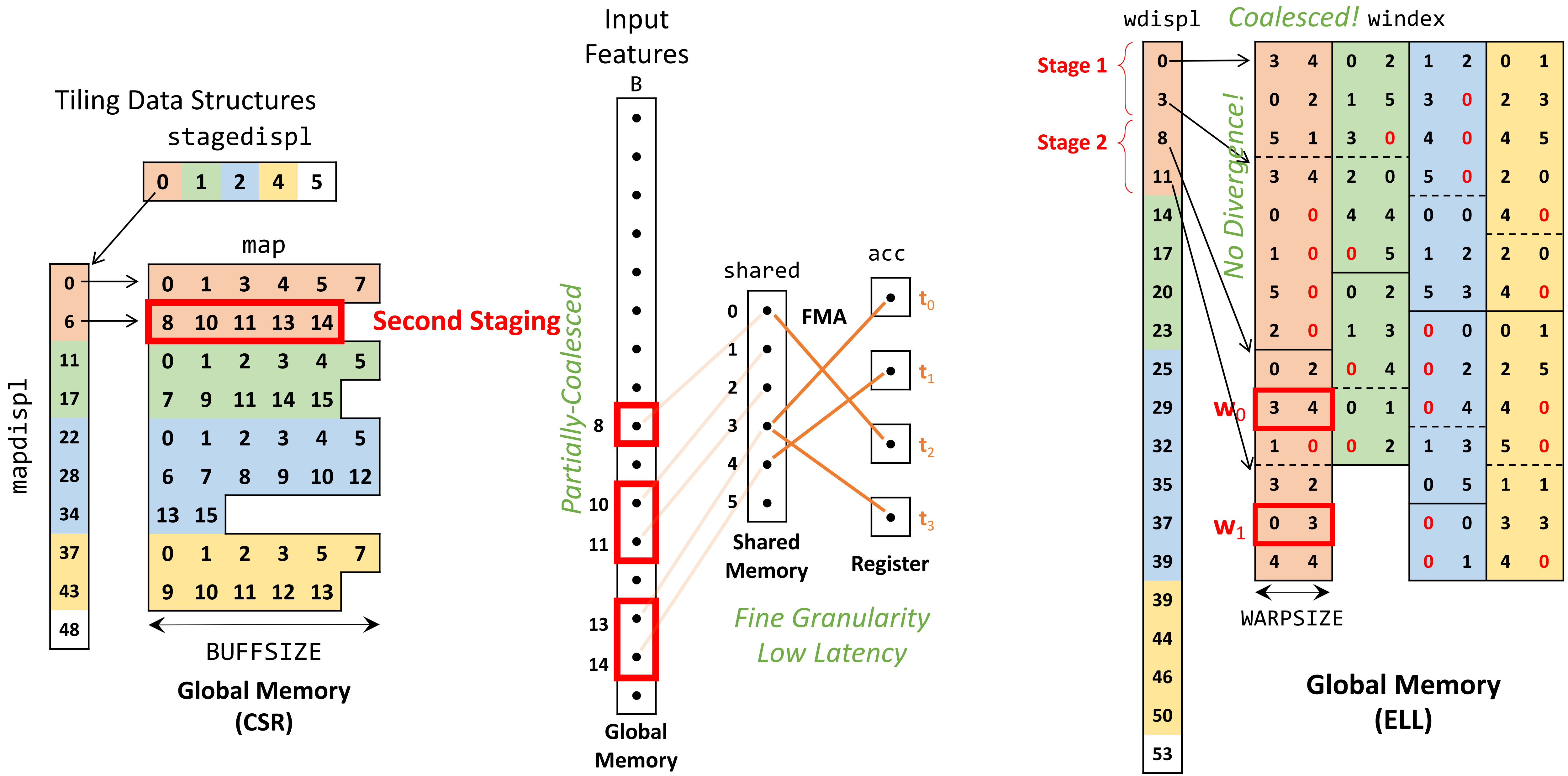

Tiling Data Structures
stagedispl
0 1 2 4 5
mapdispl
0 6 11 17 22 28 34 37 43 48
map
0 1 3 4 5 7
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
First Staging
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
Partially-Coalesced
Global
Memory
shared
Shared
Memory
acc
t_0
t_1
t_2
t_3
Register
Fine Granularity
Low Latency
__syncthreads();
wdispl
Coalesced! windex
Stage 1
Stage 2
No Divergence!
0 3 8 11 14 17 20 23 25 29 32 35 37 39 39 44 46 50 53
WARPSIZE
Global Memory
(ELL)

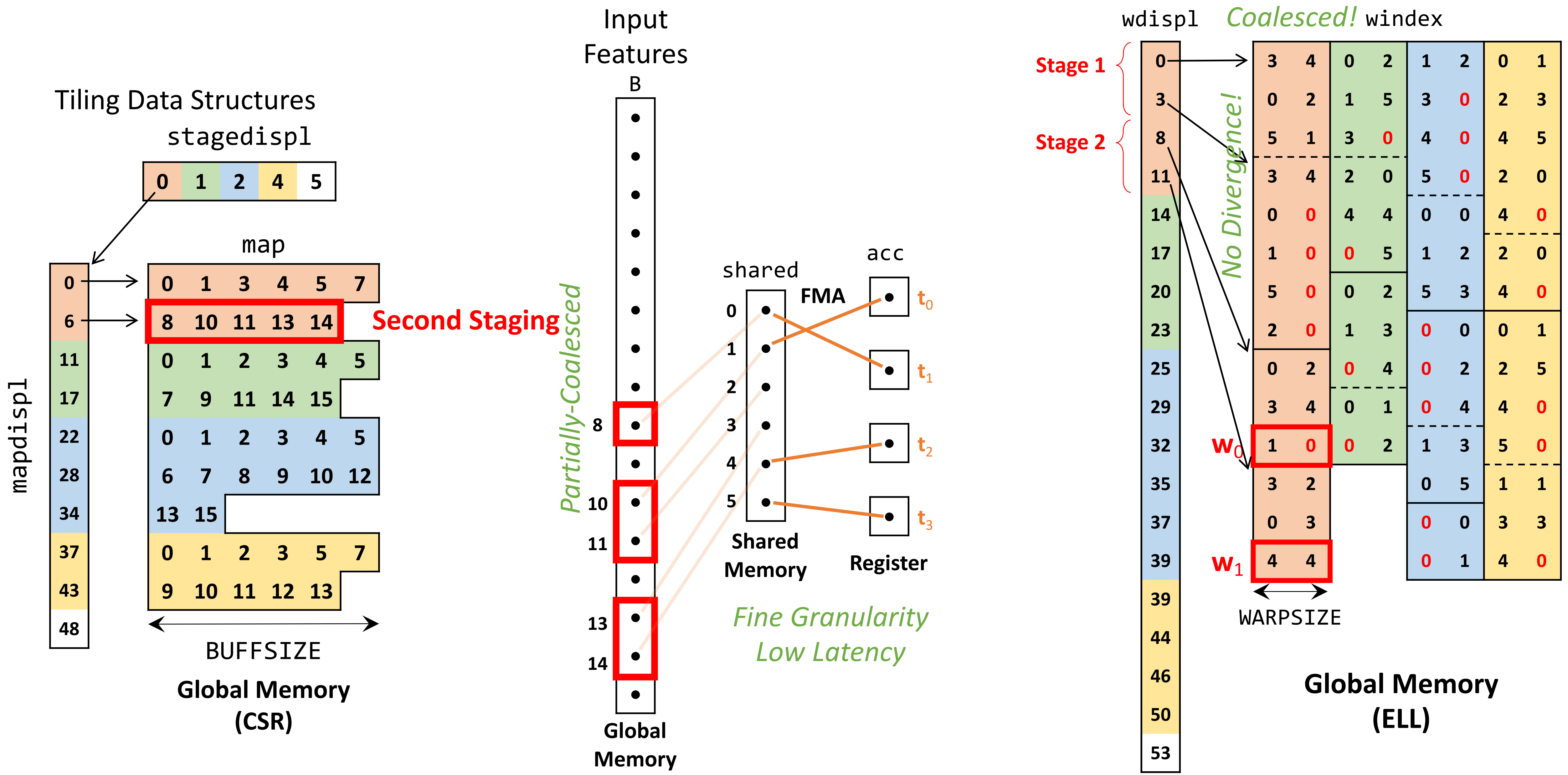


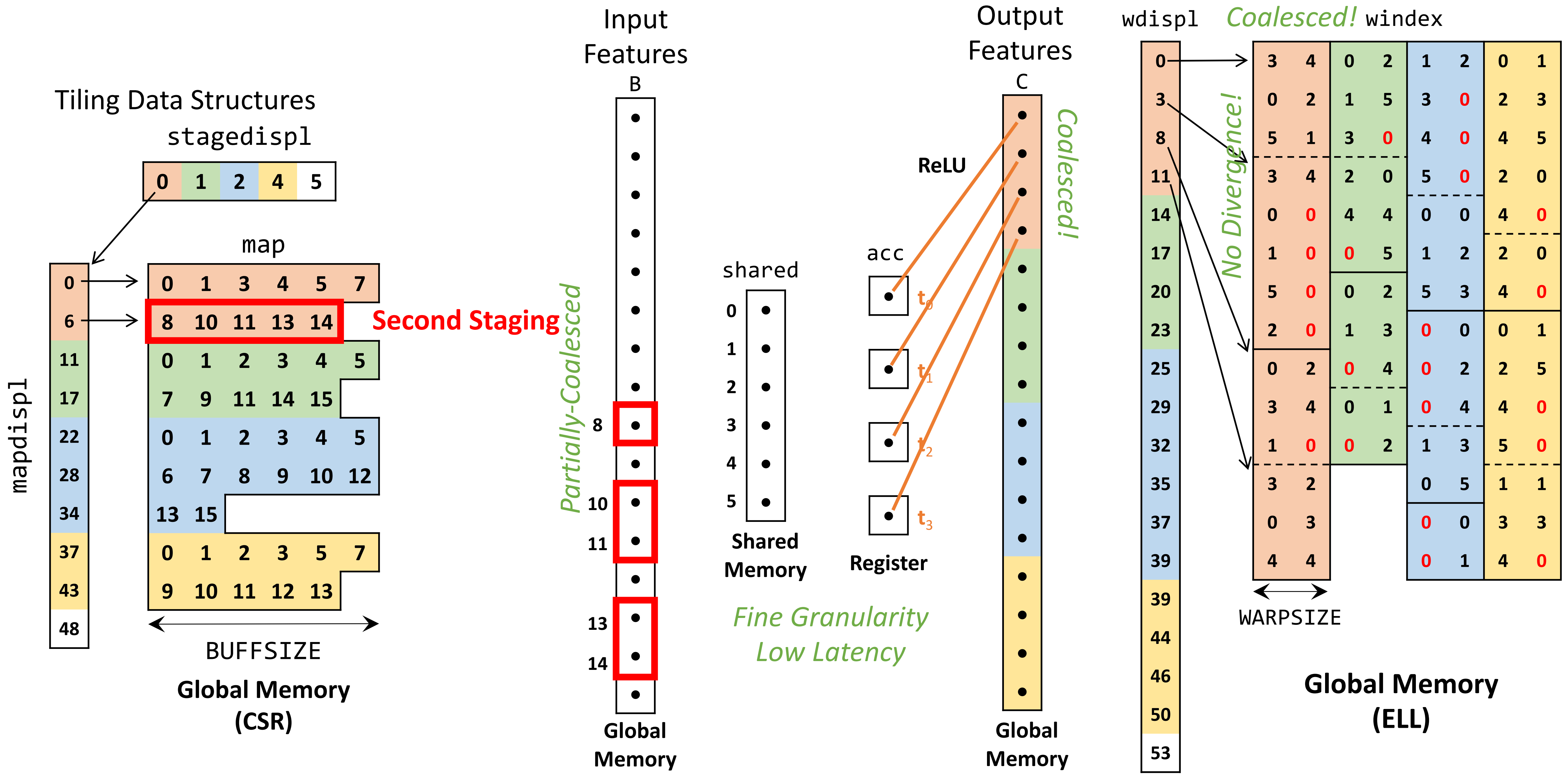
Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0 6 11 17 22 28 34 37 43 48
0 1 3 4 5 7
8 10 11 13 14
Second Staging
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
BUFFSIZE
Global Memory (CSR)
Input Features
B
Partially-Coalesced
8
10
11
13
14
Global Memory
shared
0 1 2 3 4 5
Shared Memory
acc
t_0 t_1 t_2 t_3
Register
Fine Granularity
Low Latency
__syncthreads();
wdispl
Coalesced! windex
Stage 1
Stage 2
No Divergence!
0 3 8 11 14 17 20 23 25 29 32 35 37 39 39 44 46 50 53
WARPSIZE
Global Memory (ELL)



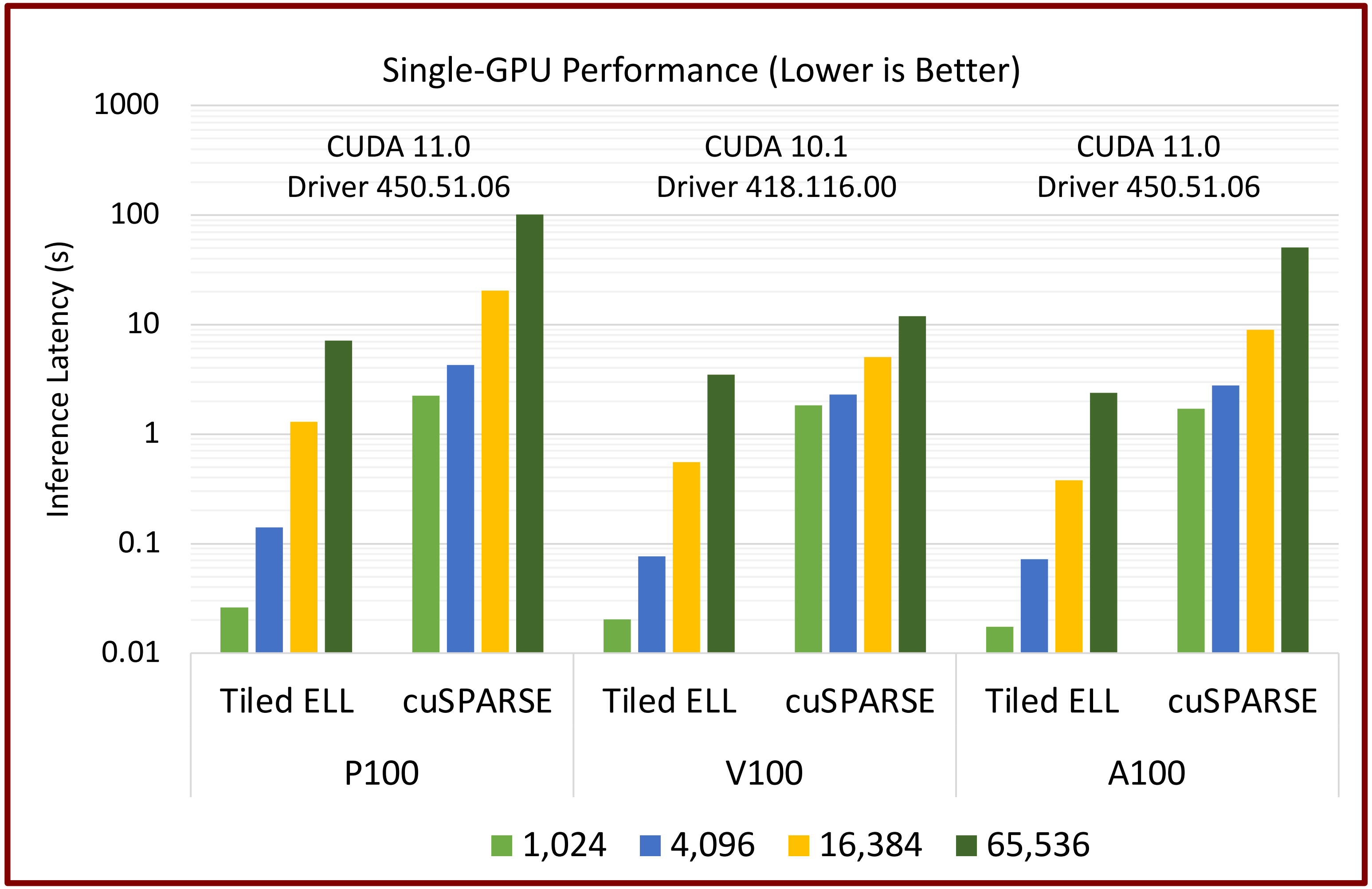
Tiling Data Structures
stagedispl
map
mapdispl
Second Staging
BUFFSIZE
Global Memory (CSR)
Input Features
B
Partially-Coalesced
Global Memory
shared
FMA
acc
Shared Memory
Register
Fine Granularity
Low Latency
wdispl
Coalesced!
windex
Stage 1
Stage 2
No Divergence!
WARPSIZE
Global Memory (ELL)


Tiling Data Structures
stagedispl
0 1 2 4 5
map
mapdispl
0 6 11 17 22 28 34 37 43 48
0 1 3 4 5 7
8 10 11 13 14
Second Staging
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
Partially-Coalesced
8
10
11
13
14
Global
Memory
shared
0 1 2 3 4 5
FMA
acc
t_0
t_1
t_2
t_3
Shared
Memory
Register
Fine Granularity
Low Latency
wdispl
Coalesced!
windex
Stage 1
Stage 2
0 3 8 11 14 17 20 23 25 29 32 35 37 39 39 44 46 50 53
No Divergence!
w_0
w_1
WARPSIZE
Global Memory
(ELL)


Tiling Data Structures
stagedispl
0 1 2 4 5
mapdispl
0 6 11 17 22 28 34 37 43 48
map
0 1 3 4 5 7
8 10 11 13 14
0 1 2 3 4 5
7 9 11 14 15
0 1 2 3 4 5
6 7 8 9 10 12
13 15
0 1 2 3 5 7
9 10 11 12 13
Second Staging
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
Partially-Coalesced
8
10
11
13
14
Global
Memory
shared
0
1
2
3
4
5
FMA
acc
t_0
t_1
t_2
t_3
Shared
Memory
Register
Fine Granularity
Low Latency
wdispl
Coalesced! windex
Stage 1
Stage 2
No Divergence!
0 3 8 11 14 17 20 23 25 29 32 35 37 39 39 44 46 50 53
w_0
w_1
WARPSIZE
Global Memory
(ELL)


Tiling Data Structures
stagedispl
mapdispl
map
Second Staging
BUFFSIZE
Global Memory
(CSR)
Input
Features
B
Partially-Coalesced
Global
Memory
shared
Shared
Memory
acc
Register
ReLU
Fine Granularity
Low Latency
Output
Features
C
Coalesced!
Global
Memory
wdispl
Coalesced! windex
No Divergence!
WARPSIZE
Global Memory
(ELL)

Application: Sparse Deep Neural Network



Application: Sparse Deep Neural Network

Inference Throughput (TeraEdges/Second)

Neurons	Layers	Single V100	Single A100	Number of V100 GPUs (Six per Node)								
				3	6	12	24	48	96	192	384	768
1024	**120**	10.51 (0.225s)	16.74 (1.59×)	18.92	22.46	25.52	28.52	27.77	**29.17**	27.89	29.12	29.13
	480	12.87 (0.073s)	20.99 (1.63×)	21.47	24.34	26.92	28.73	28.43	**29.30**	28.80	29.10	23.06
	1920	14.30 (0.264s)	20.68 (1.45×)	22.26	24.77	27.33	28.70	28.58	28.60	28.73	**28.83**	28.83
4096	**120**	9.45 (0.100s)	14.27 (1.51×)	20.69	31.36	47.82	62.03	70.31	75.81	79.11	81.13	**82.20**
	480	11.74 (0.322s)	18.63 (1.59×)	28.18	40.58	56.54	67.63	73.16	77.27	80.02	79.97	**82.22**
	1920	13.88 (1.08s)	19.86 (1.43×)	30.53	44.48	62.74	72.57	73.72	76.25	79.99	80.67	**82.32**
16384	**120**	6.15 (0.614s)	11.60 (1.89×)	16.31	28.85	50.74	64.33	89.18	111.44	**146.88**	114.87	111.30
	480	7.45 (2.027s)	14.31 (1.92×)	19.82	32.88	50.83	71.45	95.78	112.61	138.62	138.30	**139.44**
	1920	7.84 (7.704s)	15.27 (1.95×)	20.86	33.62	57.08	77.73	104.83	120.63	146.11	146.30	**146.40**
65536	**120**	3.47 (4.352s)	8.15 (2.35×)	10.90	18.77	34.20	51.14	73.67	100.72	162.19	173.25	**179.58**
	480	3.83 (15.769s)	9.08 (2.37×)	12.13	20.39	37.63	56.66	75.29	108.06	166.15	**170.26**	169.30
	1920	3.93 (61.474s)	9.33 (2.37×)	12.47	20.88	38.81	58.08	77.55	112.01	167.43	170.06	**171.37**

Application: Sparse Deep Neural Network

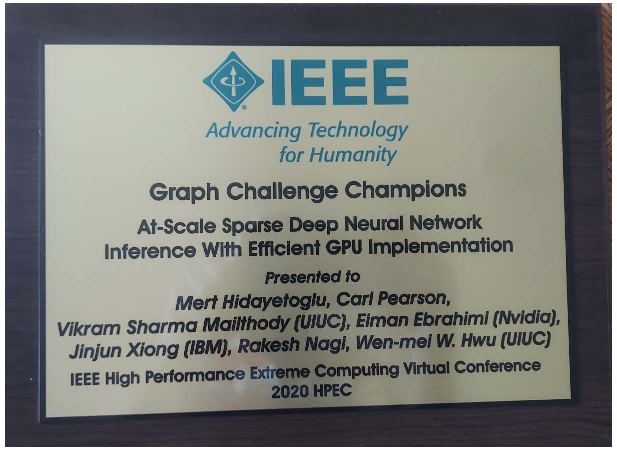
GraphChallenge

center for cognitive computing systems research

Graph Challenge Champions

2020 Champions

- *Scaling Graph Clustering with Distributed Sketches* - Benjamin Priest (LLNL), Alec Dunton (CU Boulder), Geoffrey Sanders (LLNL)
- *At-Scale Sparse Deep Neural Network Inference With Efficient GPU Implementation* - Mert Hidayetoglu, Carl Pearson, Vikram Sharma Mailthody (UIUC), Eiman Ebrahimi (Nvidia), Jinjun Xiong (IBM)), Rakesh Nagi, Wen-mei W. Hwu (UIUC)
- *A Novel Inference Algorithm for Large Sparse Neural Network using Task Graph Parallelism* - Dian-Lun Lin, Tsung-Wei Huang (Univ of Utah)
- *TriC: Distributed-memory Triangle Counting by Exploiting the Graph Structure* - Sayan Ghosh, Mahantesh Halappanavar (PNNL)

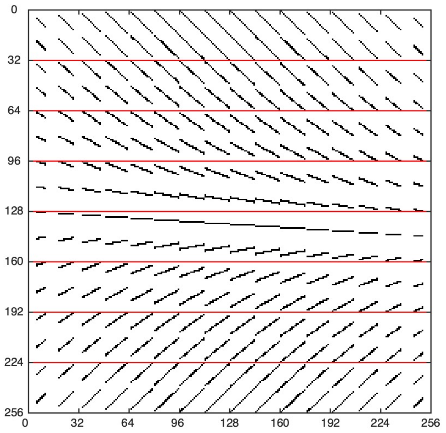


Open Source and Reproducible:
https://github.com/merthidayetoglu/SpDNN_Challenge2020

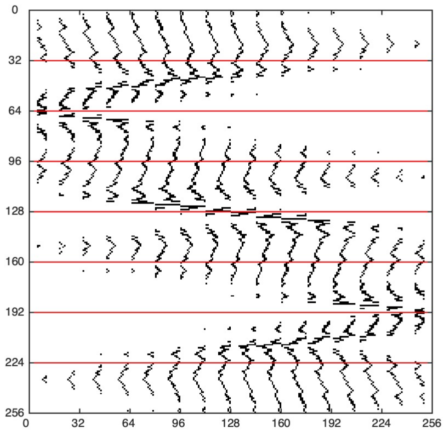
Open Problems: Enhancing Tiled SpMM by Matrix Reordering

Original Matrix

Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7



Row Reordering



Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7

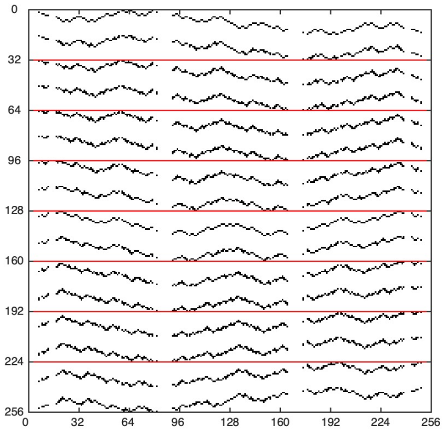
Data Reuse: 2.52

Data Reuse: 4.44

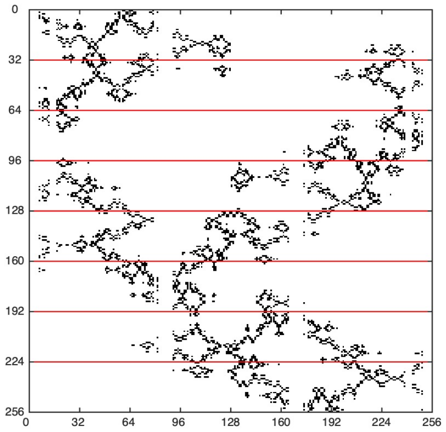
Column Reordering

Column Reordering

Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7



Row Reordering



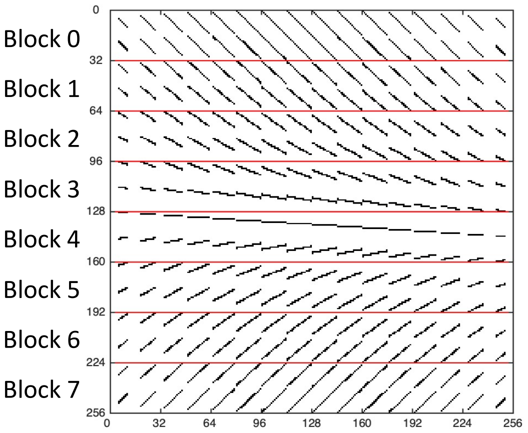
Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7

Data Reuse: 2.52

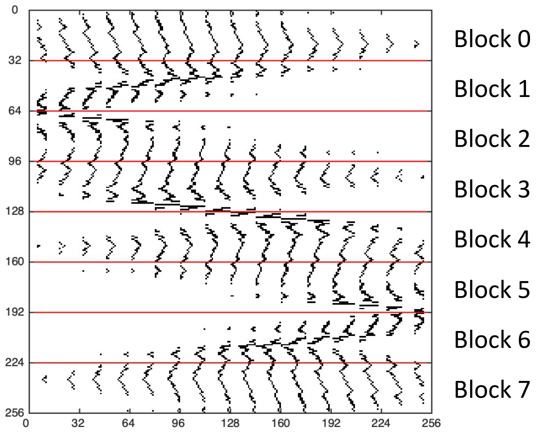
Data Reuse: 4.44

Open Problems: Enhancing Tiled SpMM by Matrix Reordering

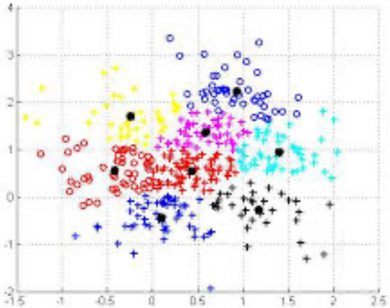
Original Matrix



Row Reordering



K-Means Clustering



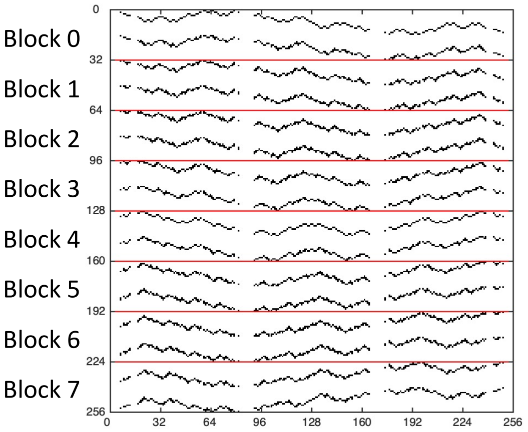
O(kNM)

Data Reuse: 2.52

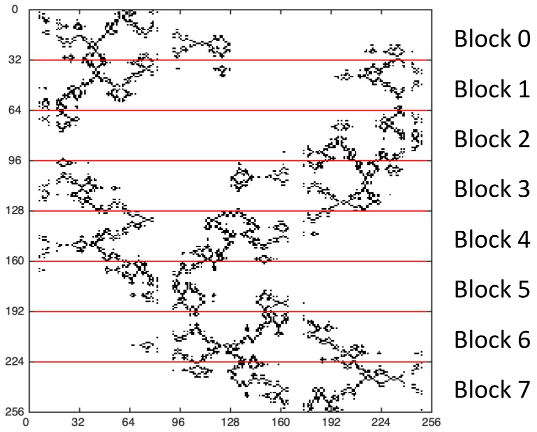
Data Reuse: 4.44

Column Reordering

Column Reordering



Row Reordering



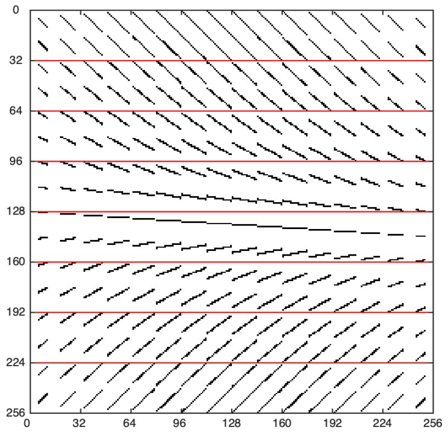
Data Reuse: 2.52

Data Reuse: 4.44

Open Problems: Enhancing Tiled SpMM by Matrix Reordering

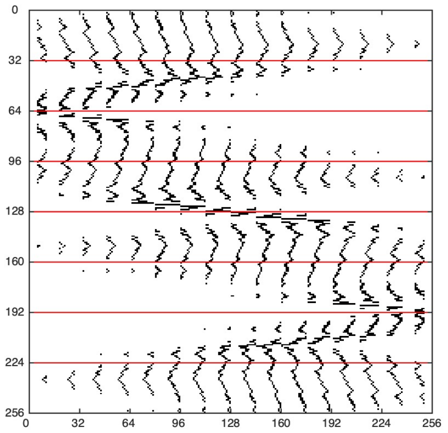
Original Matrix

Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7



Data Reuse: 2.52

Row Reordering



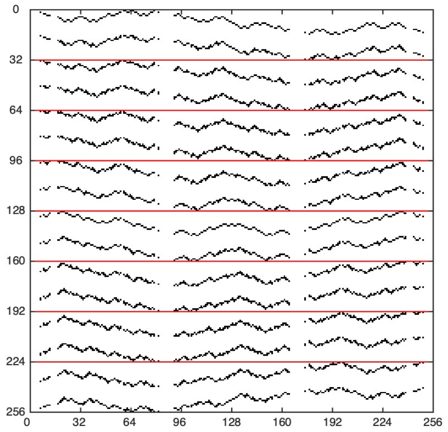
Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7

Data Reuse: 4.44

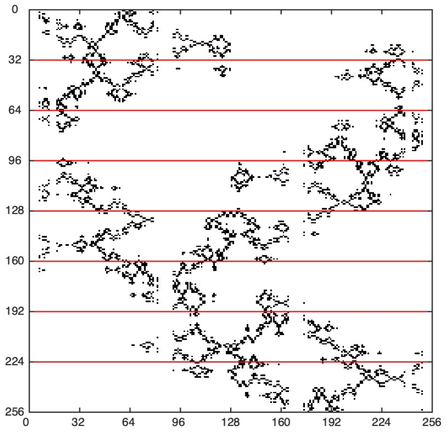
Column Reordering

Column Reordering

Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7



Row Reordering

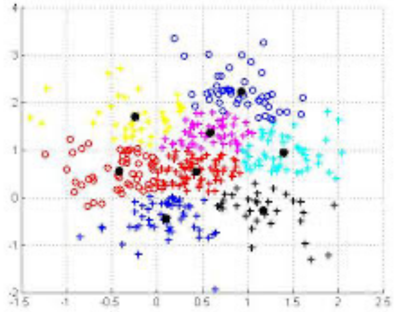


Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7

Data Reuse: 2.52

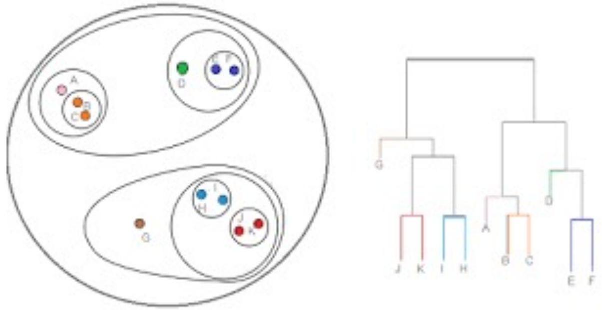
Data Reuse: 4.44

K-Means Clustering



O(kNM)

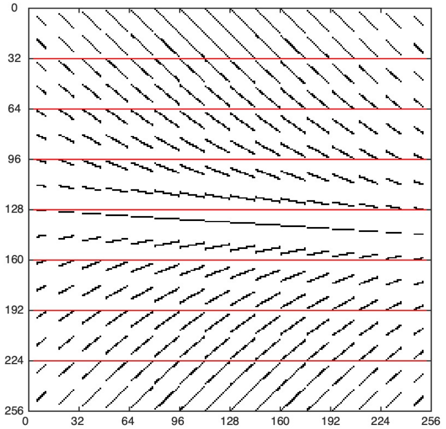
Hierarchical Clustering



Open Problems: Enhancing Tiled SpMM by Matrix Reordering

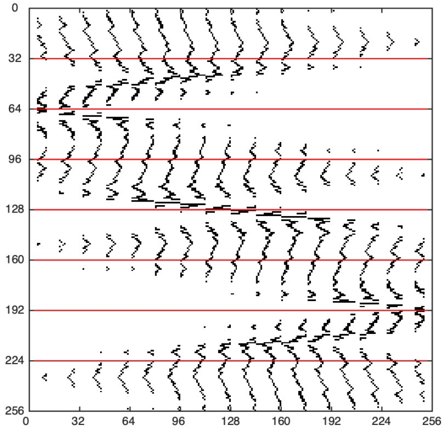
Original Matrix

Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7



Data Reuse: 2.52

Row Reordering



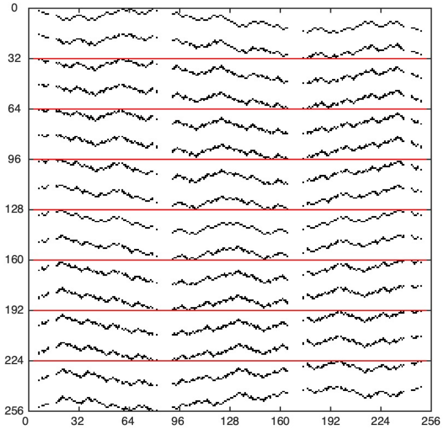
Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7

Data Reuse: 4.44

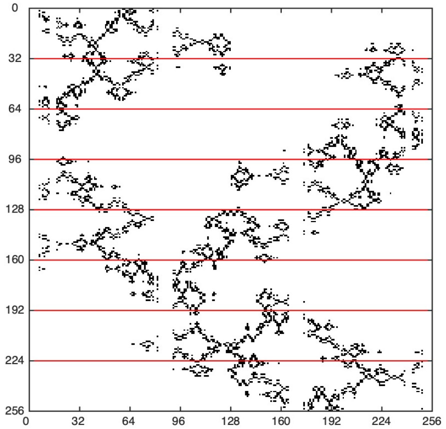
Column Reordering

Column Reordering

Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7



Row Reordering

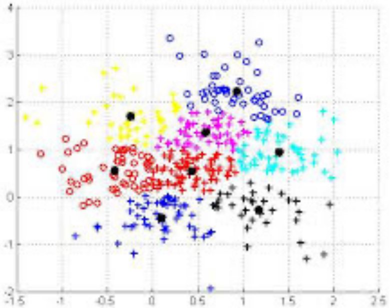


Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7

Data Reuse: 2.52

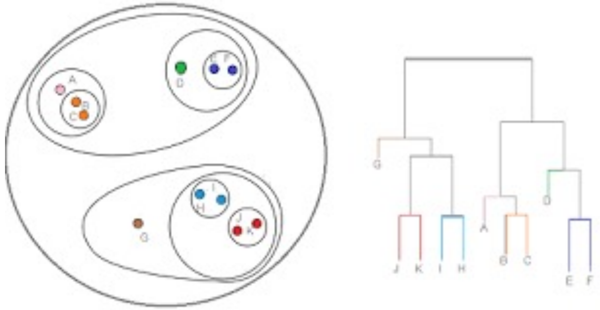
Data Reuse: 4.44

K-Means Clustering

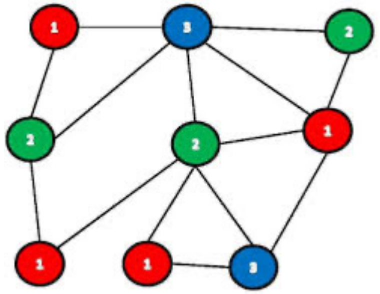


O(kNM)

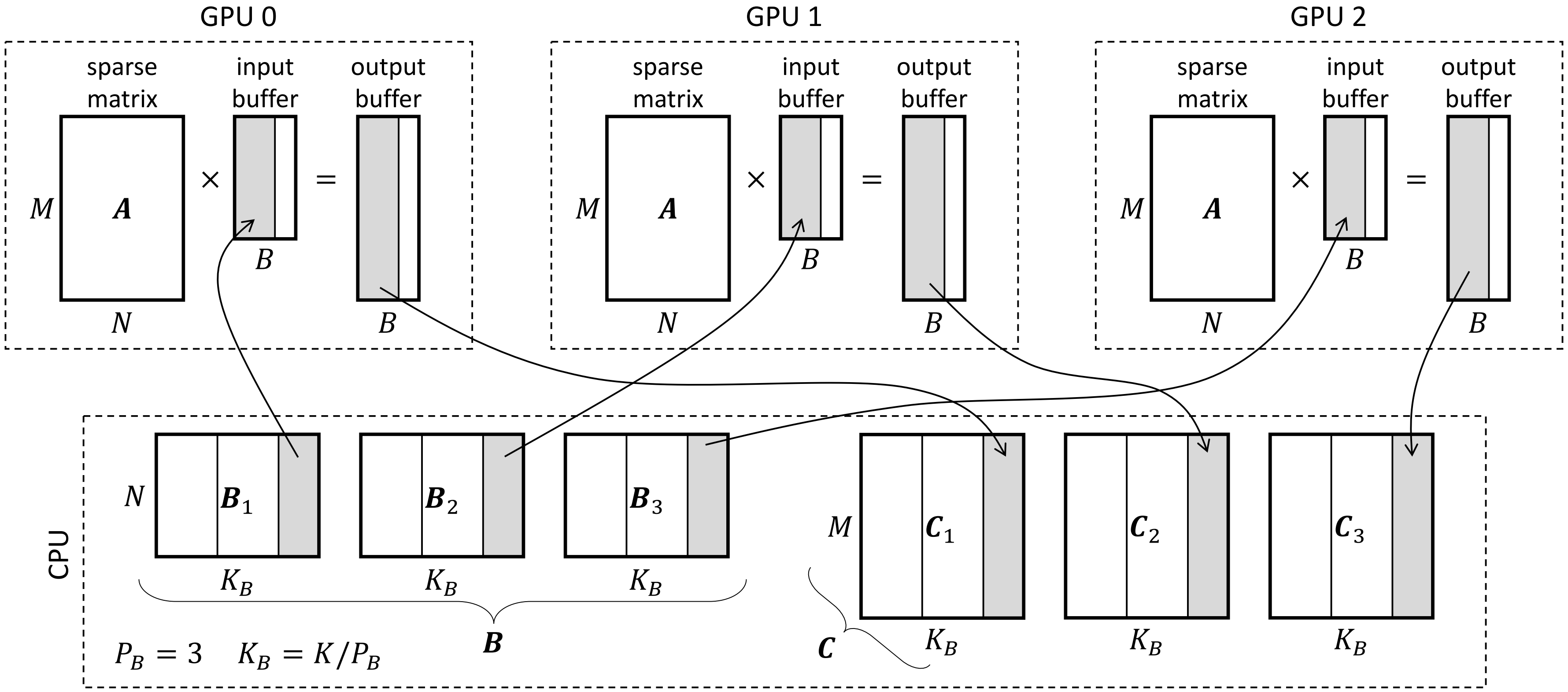
Hierarchical Clustering



Graph Coloring



Partitioning SpMM: Batch Parallelization & Streaming

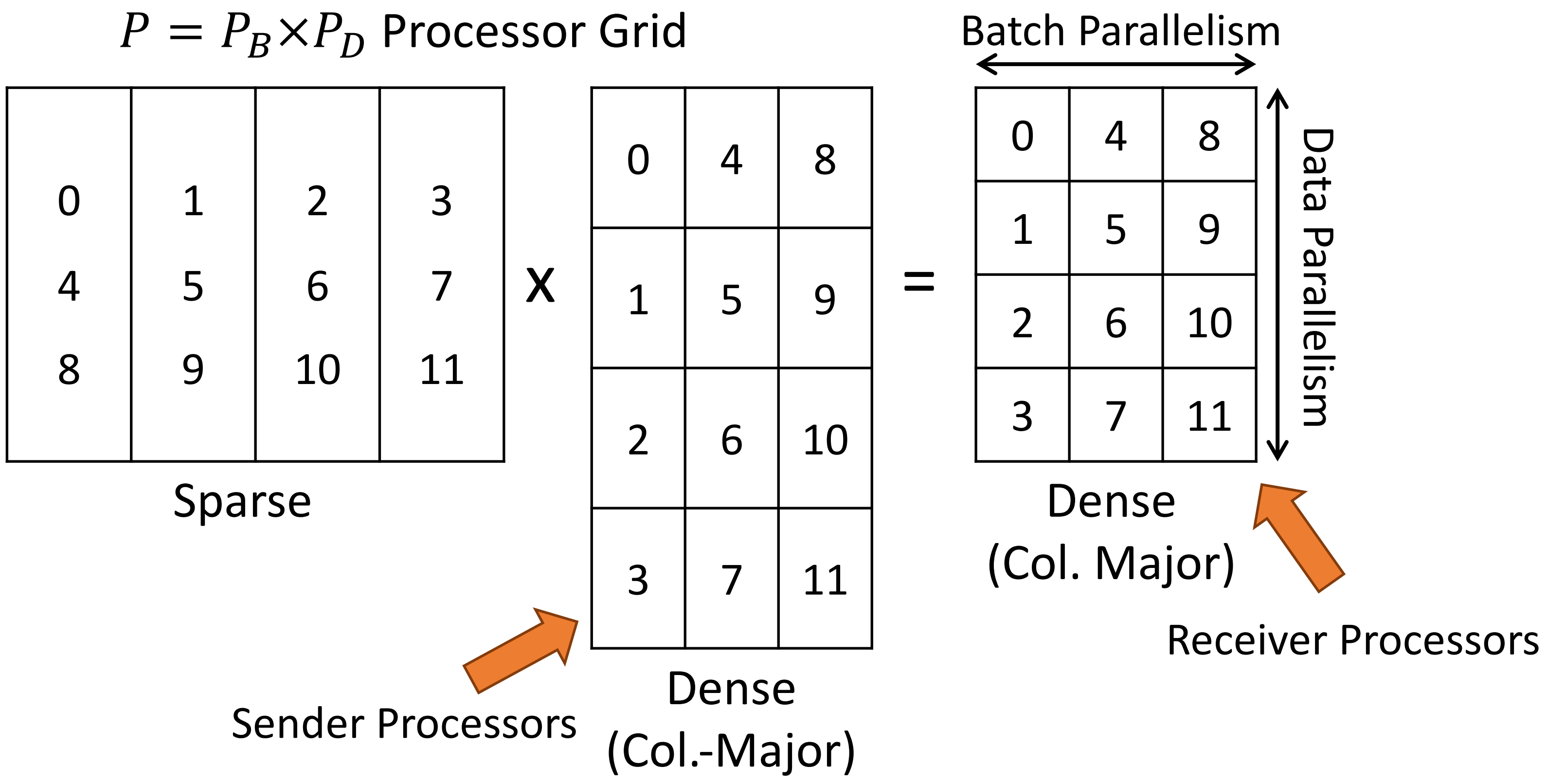


SpMM: $\boldsymbol{A} \times \boldsymbol{B} = \boldsymbol{C}$
$\boldsymbol{A}$: $M \times N$ Sparse
$\boldsymbol{B}$: $N \times K$ Dense (Col.-Major)
$\boldsymbol{C}$: $M \times K$ Dense (Col.-Major)

- **Batch parallelism duplicates the partitions**

Per-process memory — Total Memory ↑ Communications —

Partitioning SpMM: Data Parallelization



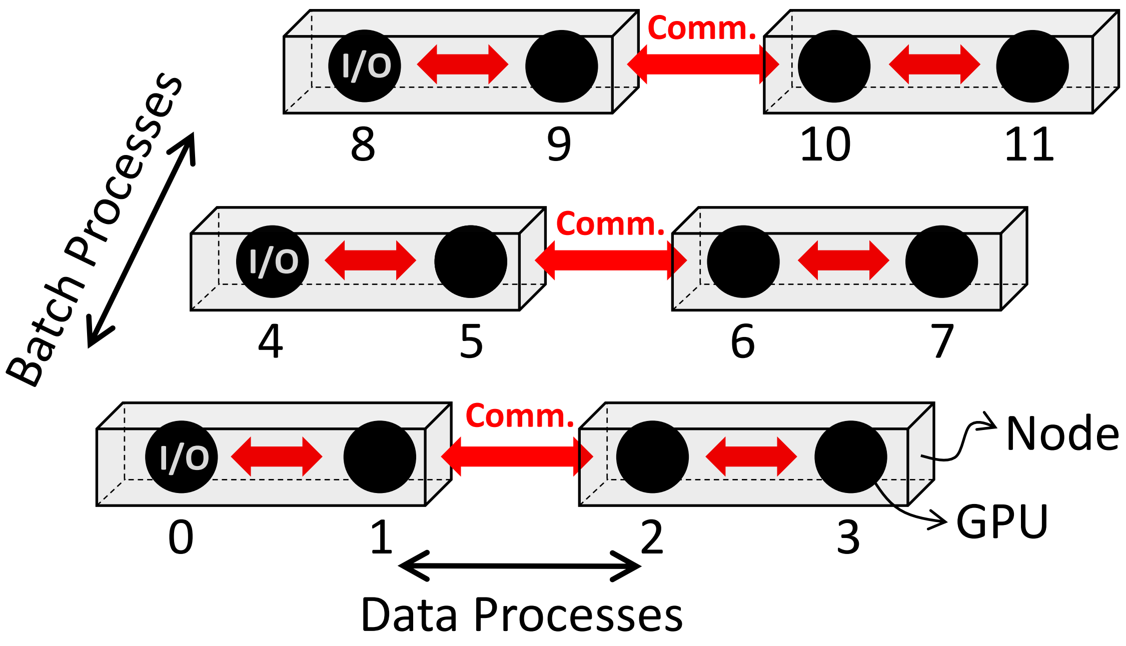
- **Batch parallelism duplicates the partitions**

  Per-process memory　　Total Memory　　Communications
- **Data parallelism partitions the matrix**

  Per-process memory　　Total Memory　　Communications

Partitioning SpMM: Topology-Aware Rank Placement

3x4 Processor Grid



Defining Orhtogonal Communicators

```
MPI_Comm_split( MPI_Comm comm, int
color, int key, MPI_Comm* newcomm)
```

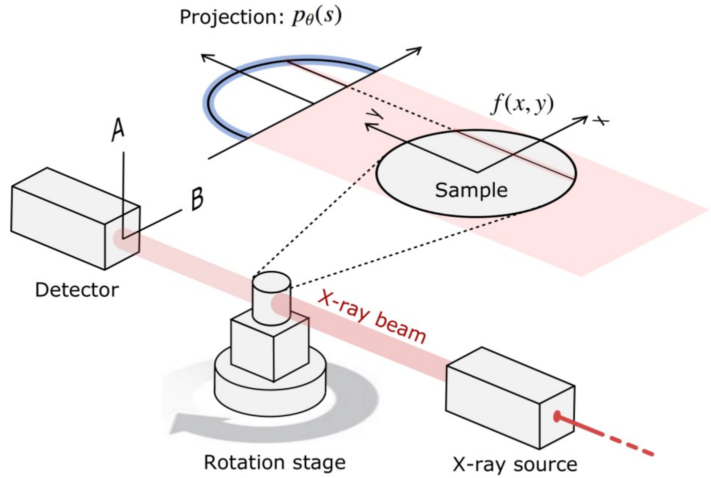
MPI_COMM_WORLD
{0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11}

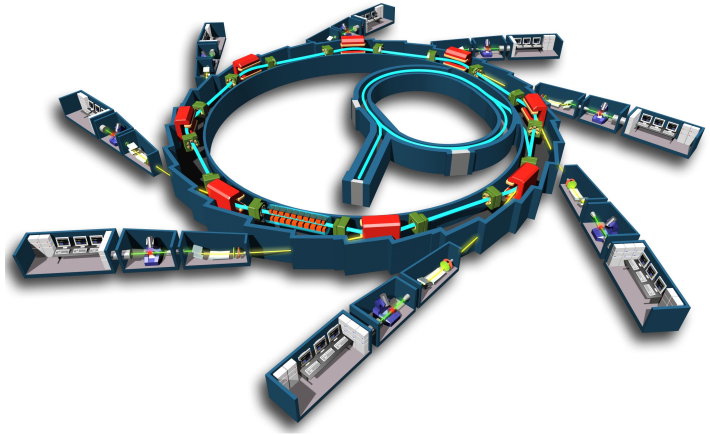
MPI_COMM_BATCH
{0, 4, 8} {2, 6, 10}
{1, 5, 9} {3, 7, 11}
{2, 6, 10}

MPI_COMM_DATA
{0, 1, 2, 3}
{4, 5, 6, 7}
{8, 9, 10, 11}

- **Batch parallelism duplicates the partitions**
  Per-process memory — Total Memory ↑ Communications —
- **Data parallelism partitions the matrix**
  Per-process memory ↓ Total Memory — Communications ↑

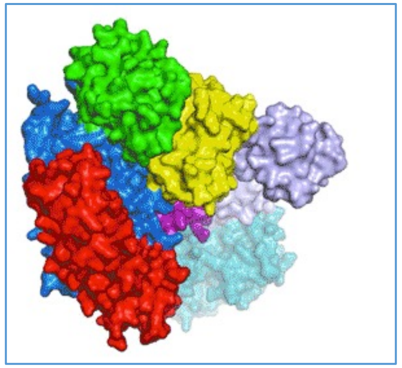
Application: X-Ray Imaging



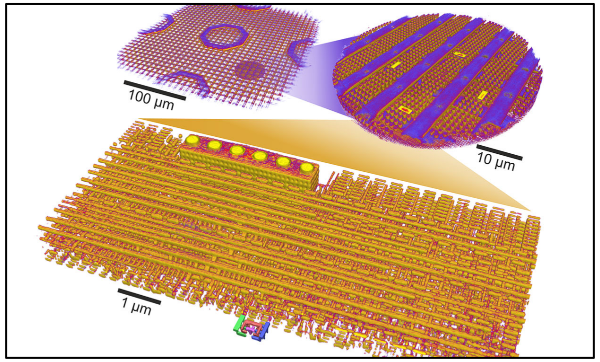


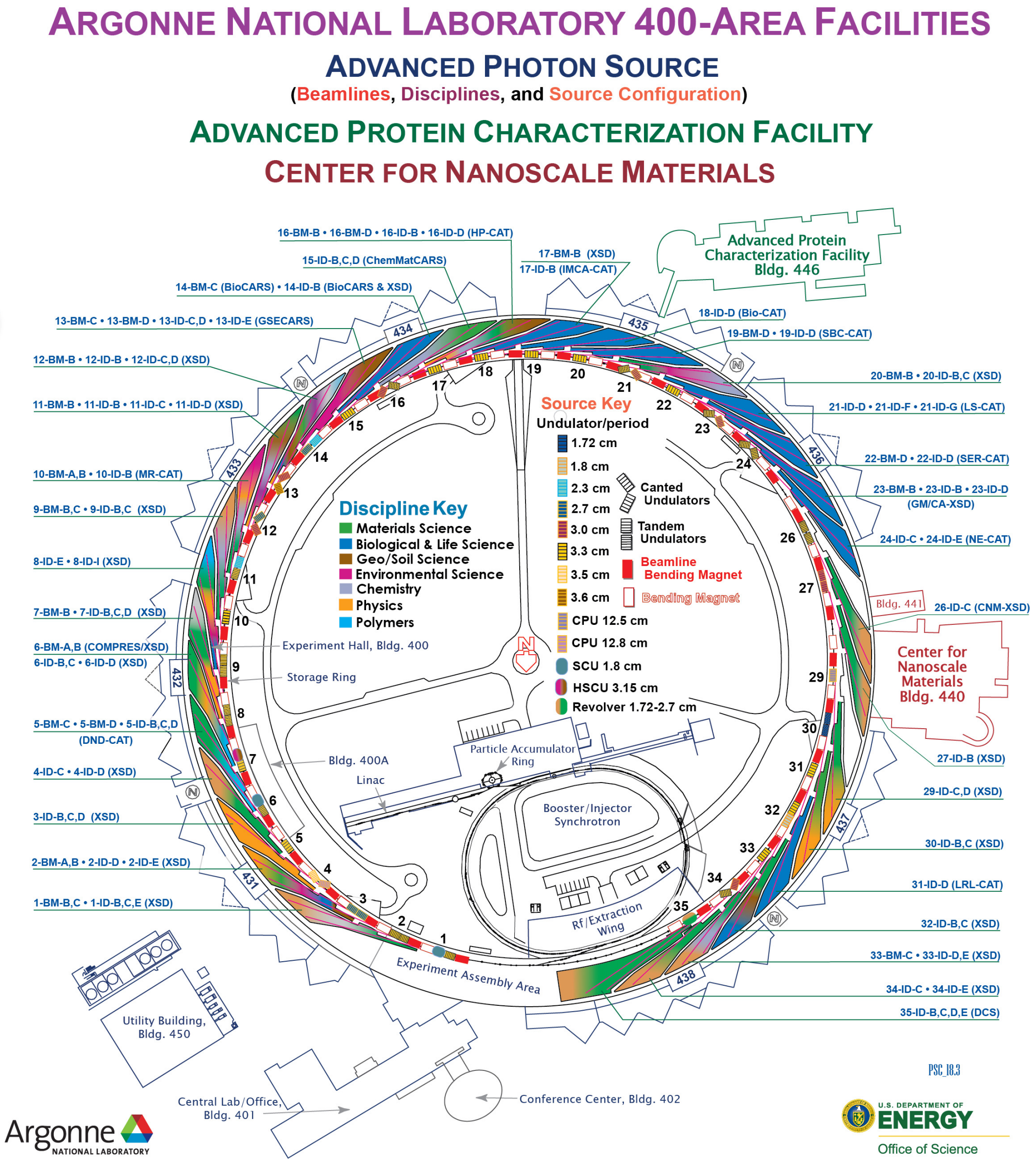
Soleil Light Source Beamlines
Paris, France

Molecular Reactions



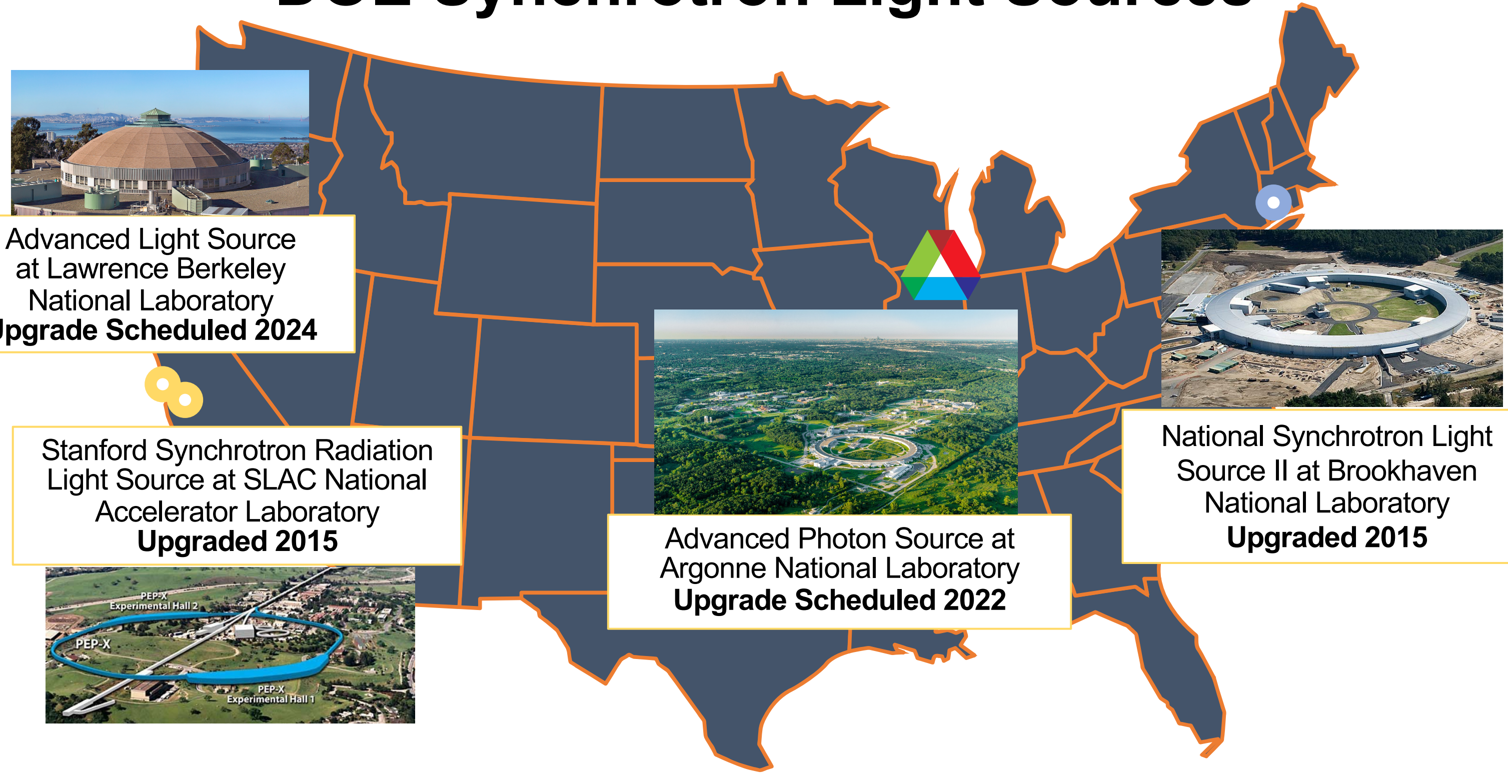
Chip Imaging





Application Performance: X-Ray Imaging

DOE Synchrotron Light Sources



Advanced Light Source
at Lawrence Berkeley
National Laboratory
Upgrade Scheduled 2024

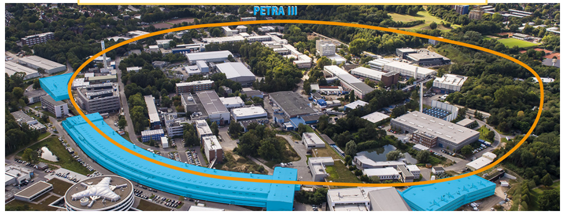
Stanford Synchrotron Radiation
Light Source at SLAC National
Accelerator Laboratory
Upgraded 2015

Advanced Photon Source at
Argonne National Laboratory
Upgrade Scheduled 2022

National Synchrotron Light
Source II at Brookhaven
National Laboratory
Upgraded 2015

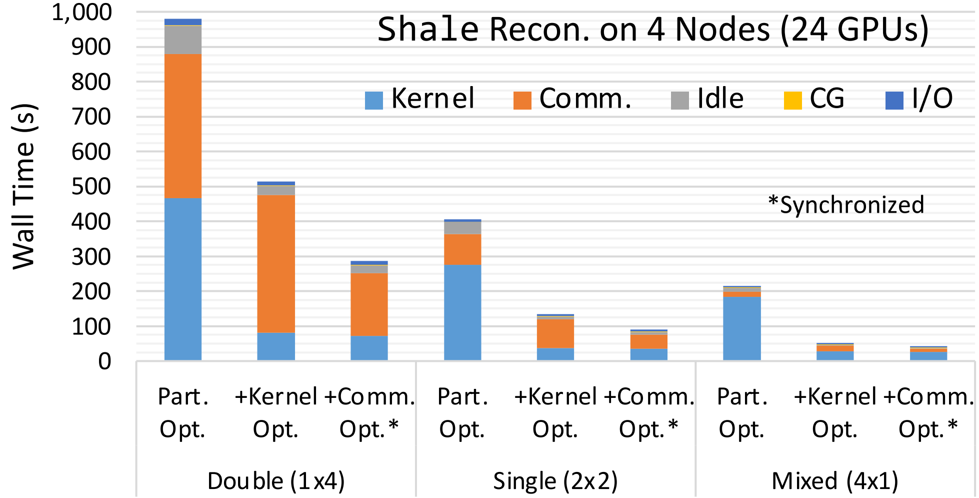
Diamond in Oxford, UK
Upgrade Scheduled 2024

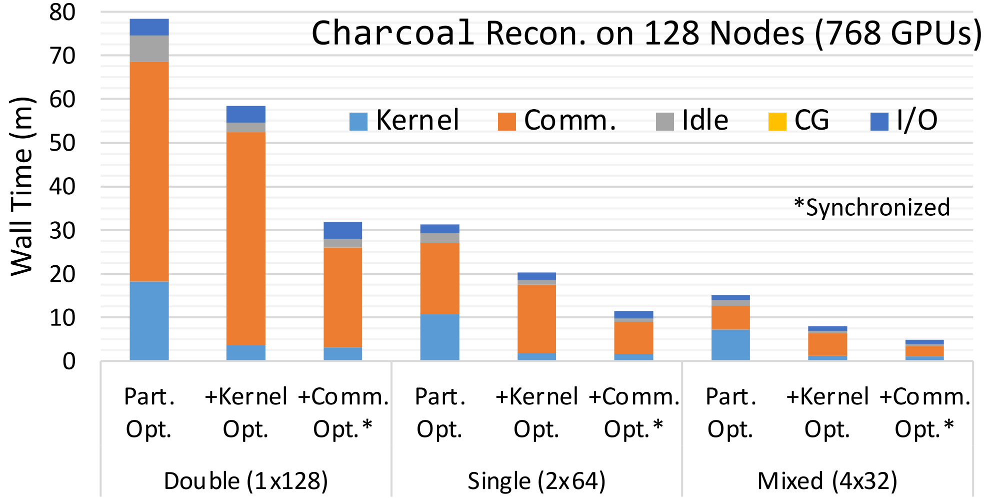
DESY in Hanofer, Germany
Upgrade Scheduled 2025



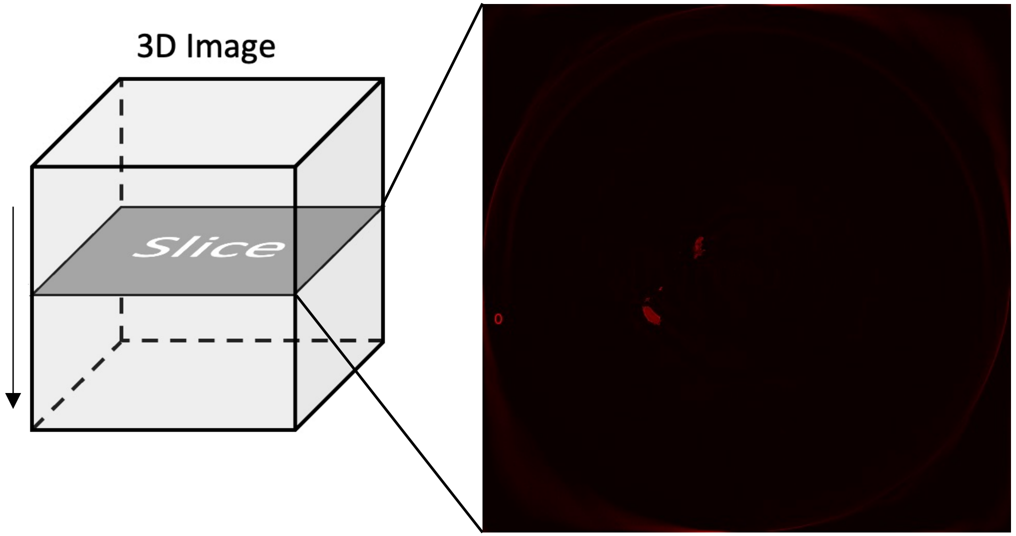
SPRING-8 in Sayo, Japan
Upgrade Scheduled TBD

Communication Bottleneck

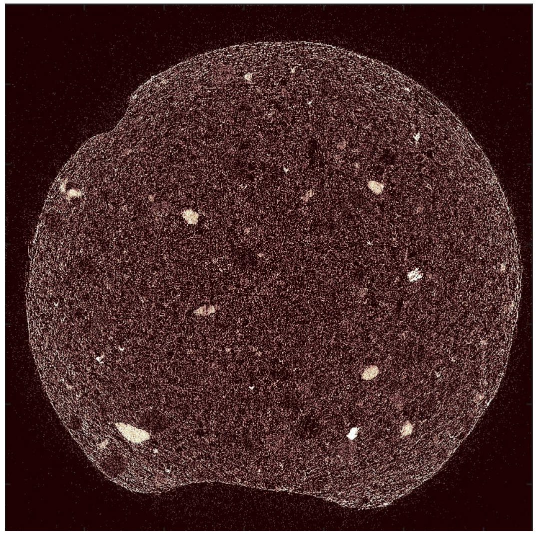




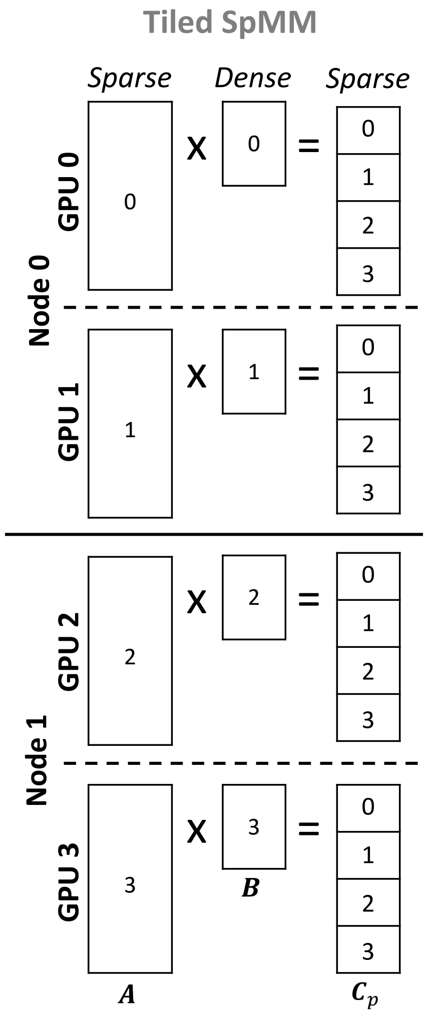
1792 x 2048 x 2048



4198 x 6613 x 6613

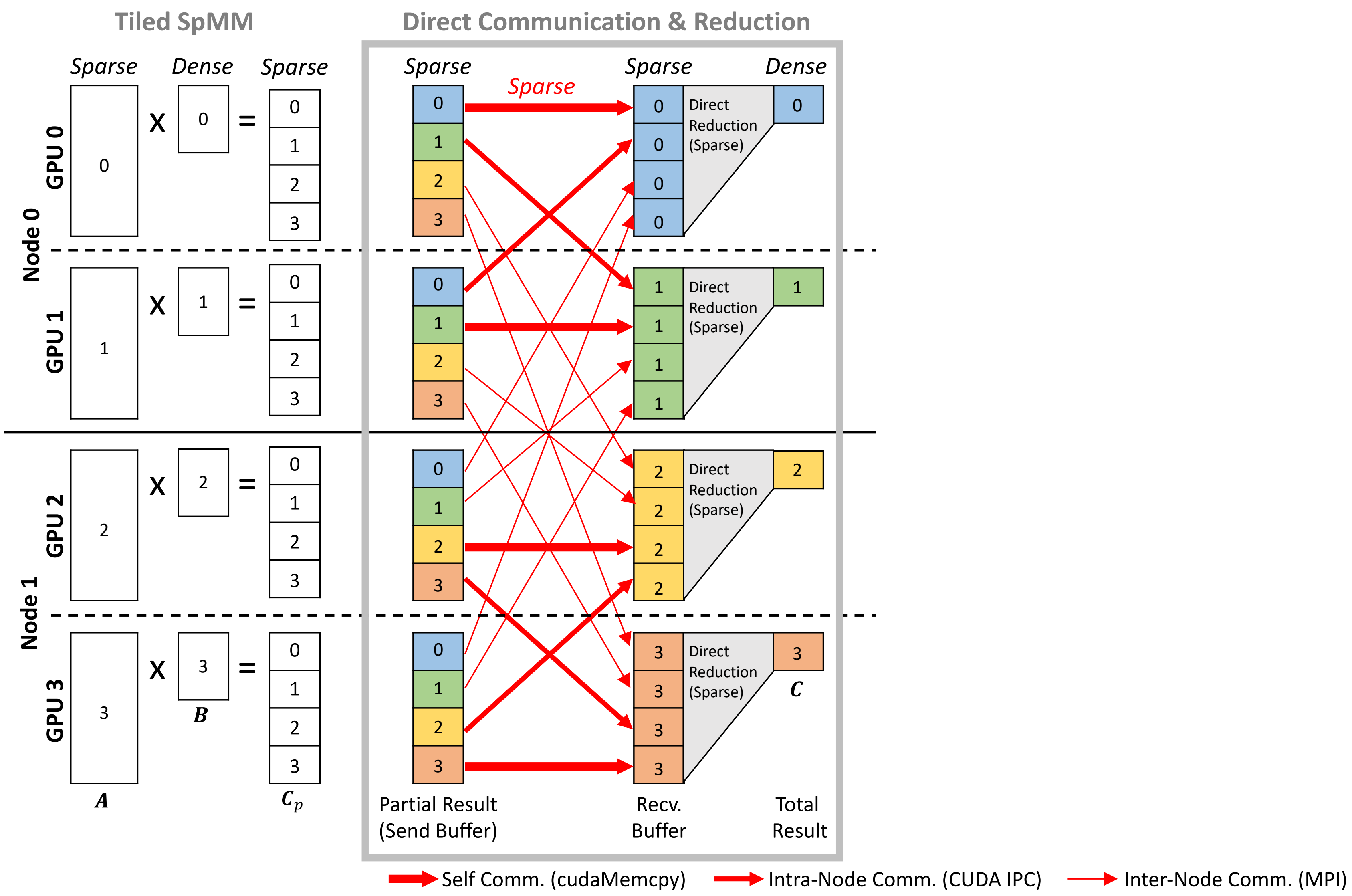


Hierarchical Communications: Overview

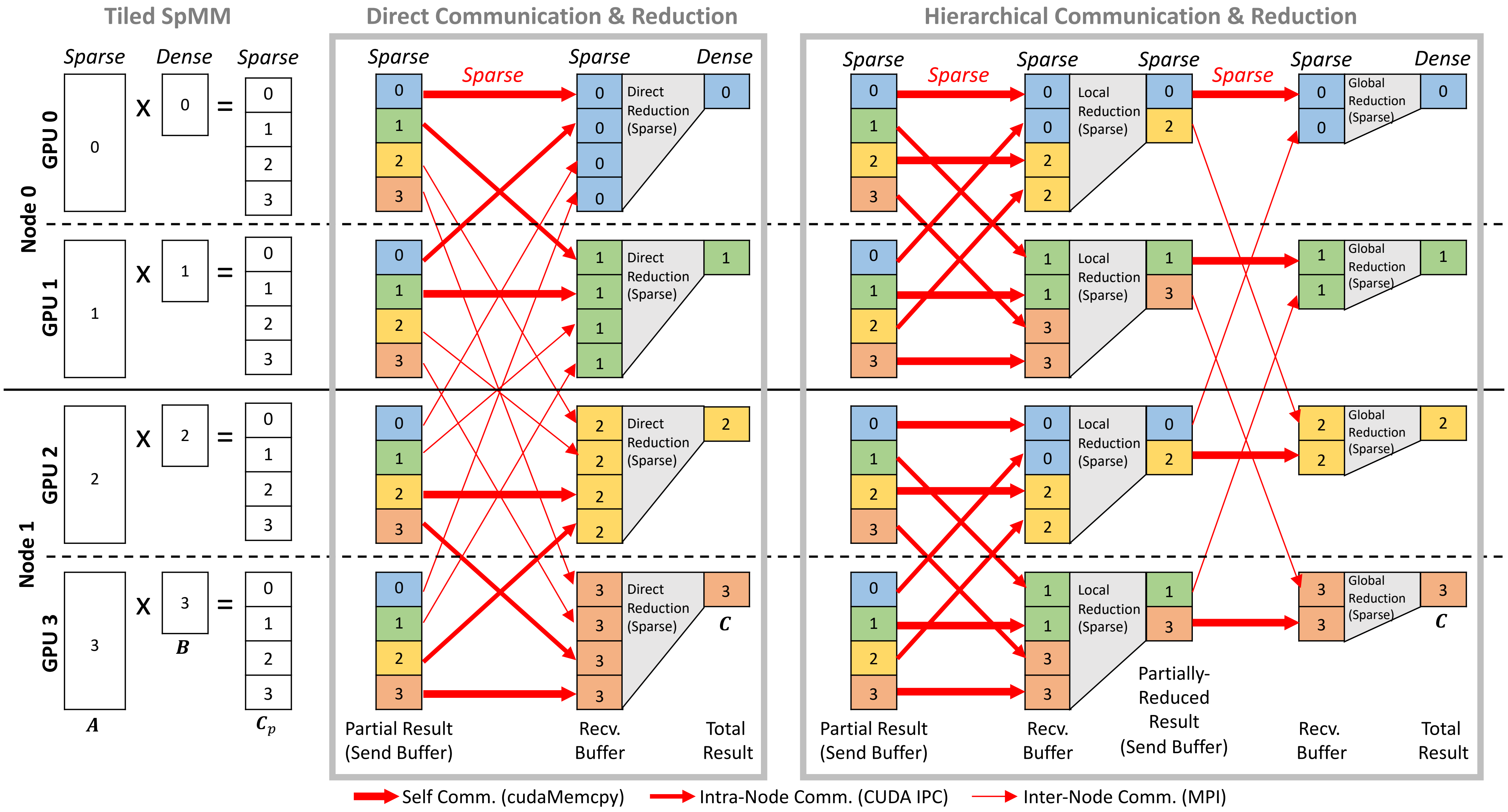


`MPI_Reduce_scatter`
(but sparse)

Hierarchical Communications: Overview



Hierarchical Communications: Overview



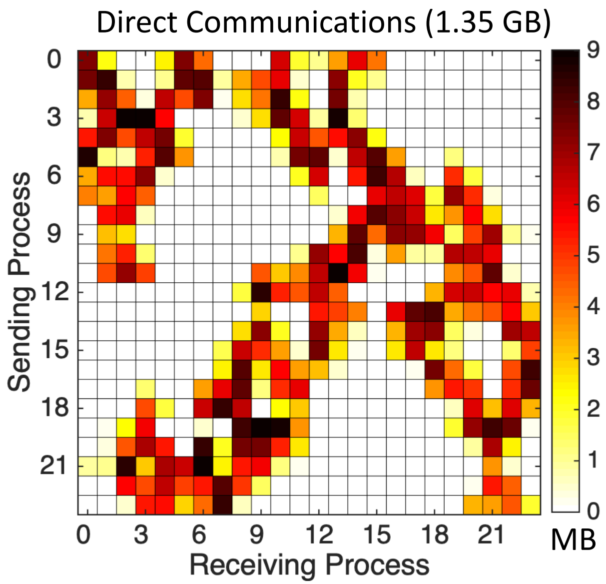
Hierarchical Communications: Three-Level Hierarchy

`MPI_COMM_BATCH`

`MPI_COMM_WORLD`

`MPI_COMM_DATA`

{0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11,
12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23}



Hierarchical Communications: Three-Level Hierarchy

MPI_COMM_WORLD

MPI_COMM_BATCH

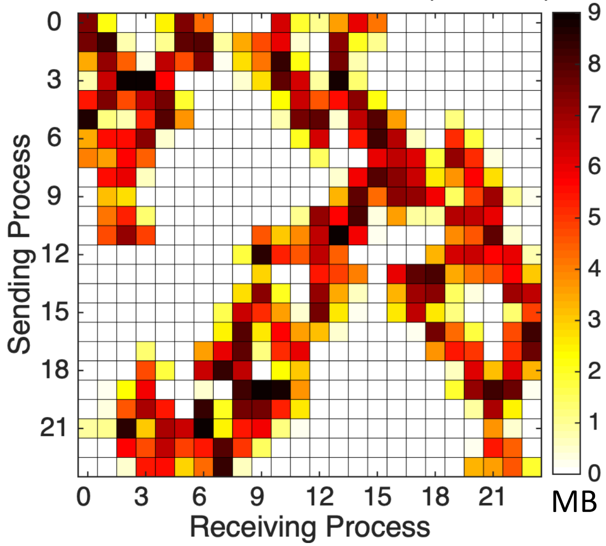
MPI_COMM_DATA

{0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11,
12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23}

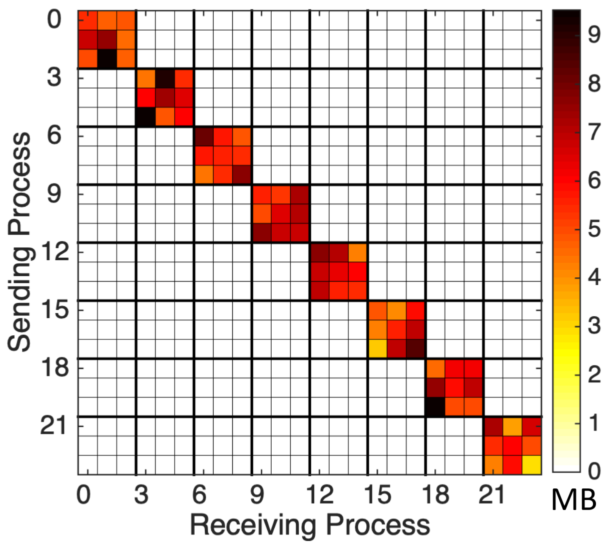
MPI_COMM_SOCKET

{0, 1, 2} {3, 4, 5}
{6, 7, 8} {9, 10, 11}
{12, 13, 14} {15, 16, 17}
{18, 19, 20} {21, 22, 23}

Direct Communications (1.35 GB)



Local Comm. within Sockets (1.35 GB)



Hierarchical Communications: Three-Level Hierarchy

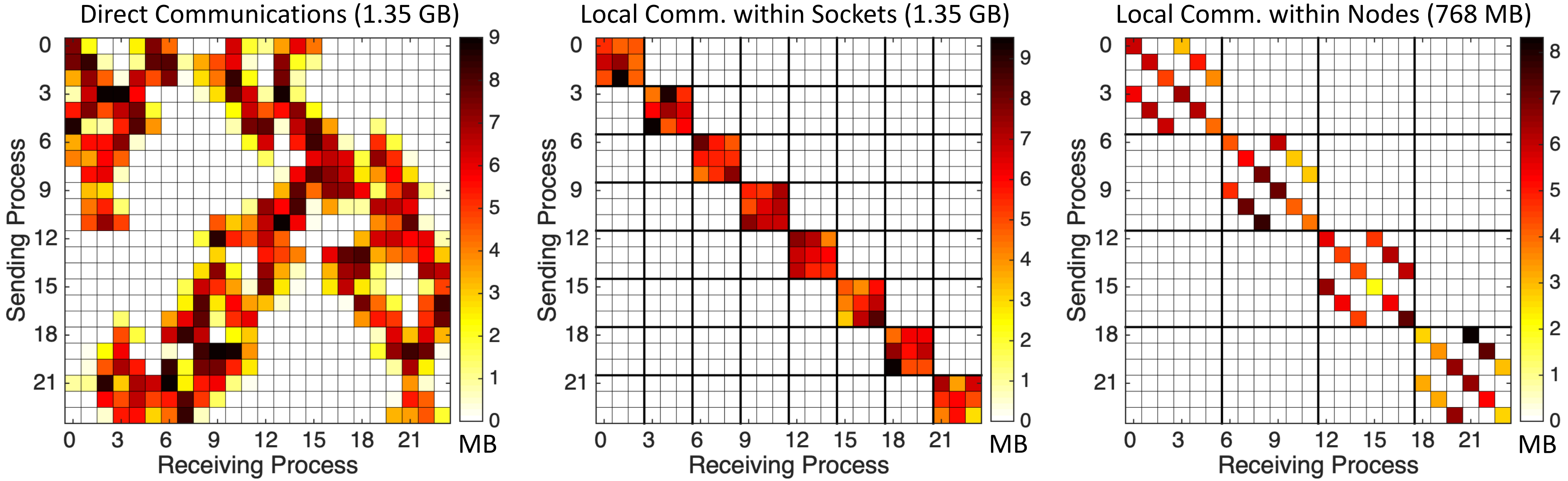
MPI_COMM_WORLD → MPI_COMM_BATCH

MPI_COMM_WORLD → MPI_COMM_DATA

MPI_COMM_DATA
{0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11,
12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23}

MPI_COMM_DATA → MPI_COMM_SOCKET
{0, 1, 2} {3, 4, 5}
{6, 7, 8} {9, 10, 11}
{12, 13, 14} {15, 16, 17}
{18, 19, 20} {21, 22, 23}

MPI_COMM_DATA → MPI_COMM_NODE
{0, 1, 2, 3, 4, 5}
{6, 7, 8, 9, 10, 11}
{12, 13, 14, 15, 16, 17}
{18, 19, 20, 21, 22, 23}



Hierarchical Communications: Three-Level Hierarchy

MPI_COMM_WORLD → MPI_COMM_BATCH

MPI_COMM_WORLD → MPI_COMM_DATA

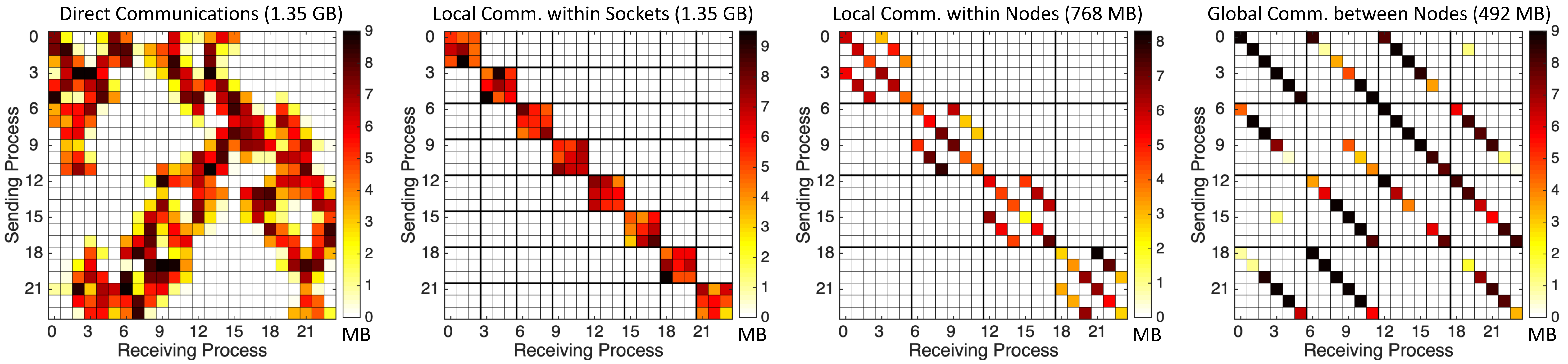
MPI_COMM_DATA
{0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11,
12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23}

MPI_COMM_DATA → MPI_COMM_SOCKET

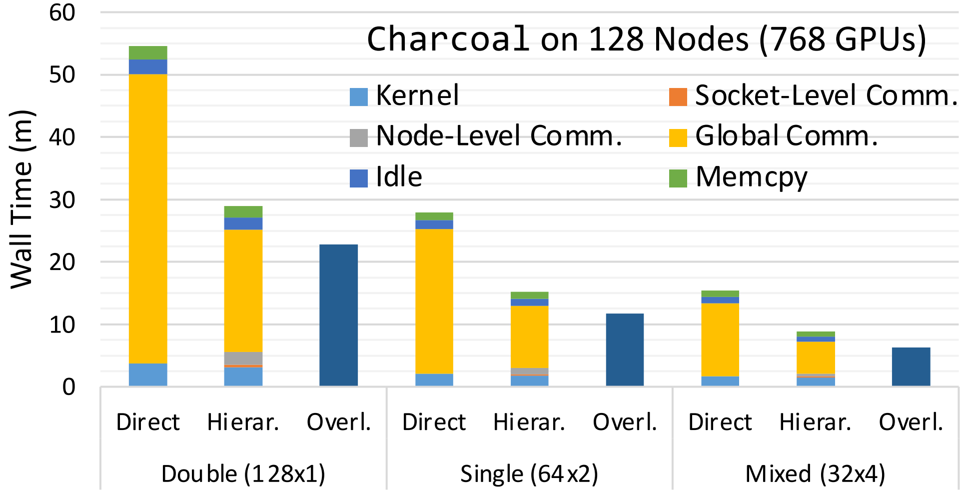
MPI_COMM_SOCKET
{0, 1, 2} {3, 4, 5}
{6, 7, 8} {9, 10, 11}
{12, 13, 14} {15, 16, 17}
{18, 19, 20} {21, 22, 23}

MPI_COMM_DATA → MPI_COMM_NODE

MPI_COMM_NODE
{0, 1, 2, 3, 4, 5}
{6, 7, 8, 9, 10, 11}
{12, 13, 14, 15, 16, 17}
{18, 19, 20, 21, 22, 23}



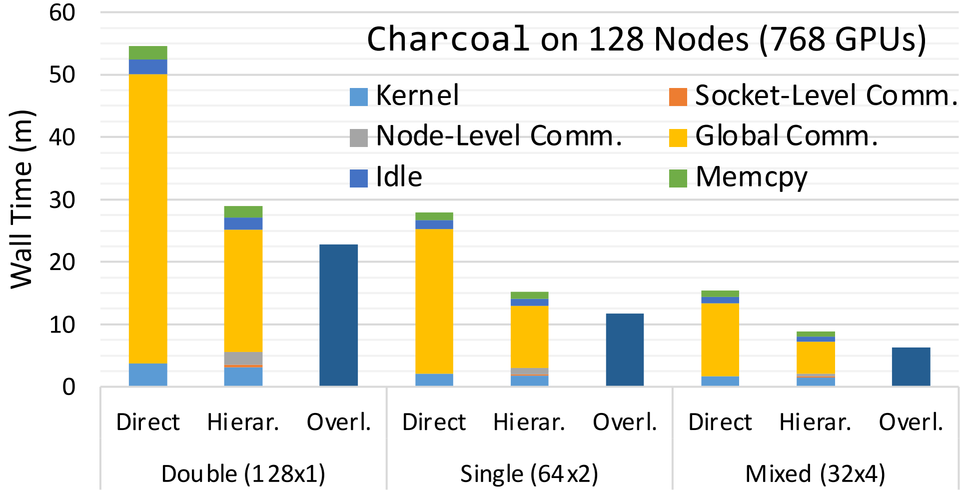
Hierarchical Communications: Benchmarking on Summit



Communication Volume and Effective Bandwidth

	Prec.	Socket-Level Comm. Data	Socket-Level Comm. B/W	Node-Level Comm. Data	Node-Level Comm. B/W	Global Comm. Data	Global Comm. B/W	Memcpy B/W
Direct	Double	N/A		N/A		36.6 TB	1.61 TB/s	35.2 TB/s
	Single					18.3 TB	1.61 TB/s	34.9 TB/s
	Mixed					9.16 TB	1.59 TB/s	34.6 TB/s
Hierar.	Double	36.6 TB	174 TB/s	21.4 TB	21.3 TB/s	15.2 TB	1.58 TB/s	34.9 TB/s
	Single	18.3 TB	170 TB/s	10.7 TB	22.8 TB/s	7.58 TB	1.55 TB/s	34.5 TB/s
	Mixed	9.16 TB	164 TB/s	5.35 TB	23.5 TB/s	3.79 TB	1.49 TB/s	33.6 TB/s

Hierarchical Communications: Benchmarking on Summit

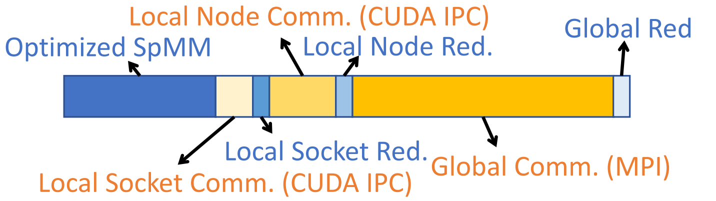


Communication Volume and Effective Bandwidth

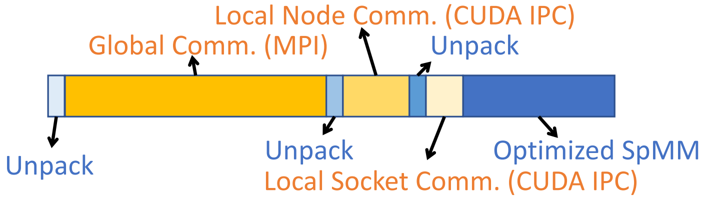
	Prec.	Socket-Level Comm. Data	Socket-Level Comm. B/W	Node-Level Comm. Data	Node-Level Comm. B/W	Global Comm. Data	Global Comm. B/W	Memcpy B/W
Direct	Double	N/A		N/A		36.6 TB	1.61 TB/s	35.2 TB/s
	Single					18.3 TB	1.61 TB/s	34.9 TB/s
	Mixed					9.16 TB	1.59 TB/s	34.6 TB/s
Hierar.	Double	36.6 TB	174 TB/s	21.4 TB	21.3 TB/s	15.2 TB	1.58 TB/s	34.9 TB/s
	Single	18.3 TB	170 TB/s	10.7 TB	22.8 TB/s	7.58 TB	1.55 TB/s	34.5 TB/s
	Mixed	9.16 TB	164 TB/s	5.35 TB	23.5 TB/s	3.79 TB	1.49 TB/s	33.6 TB/s

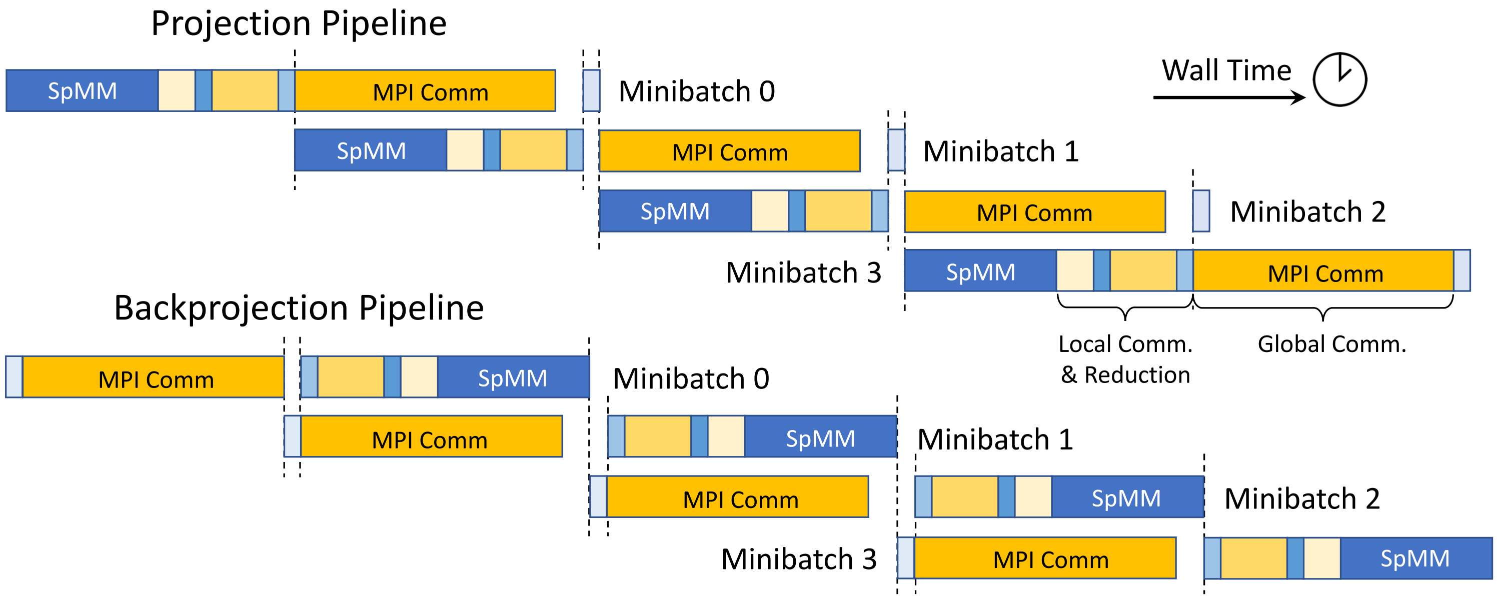
Pipelining Batch Processing

Projection Minibatch

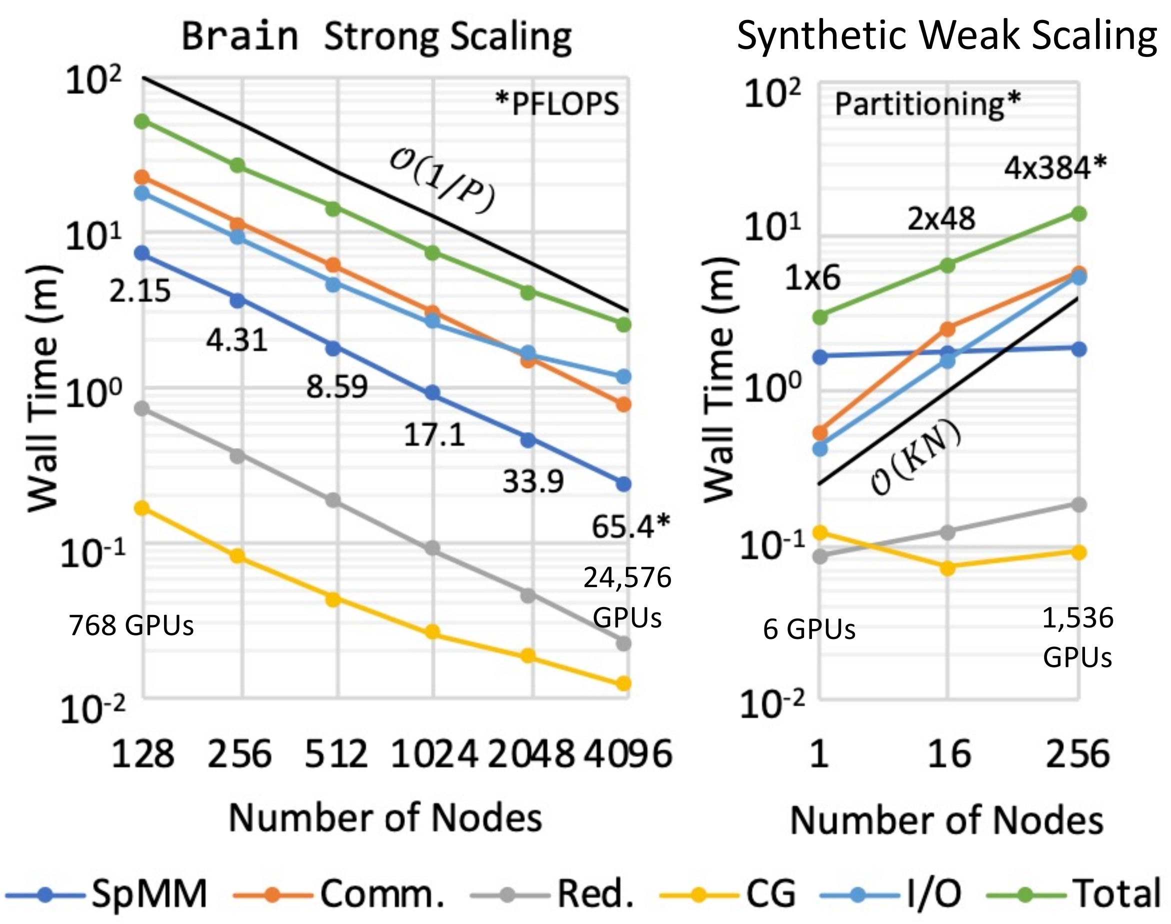


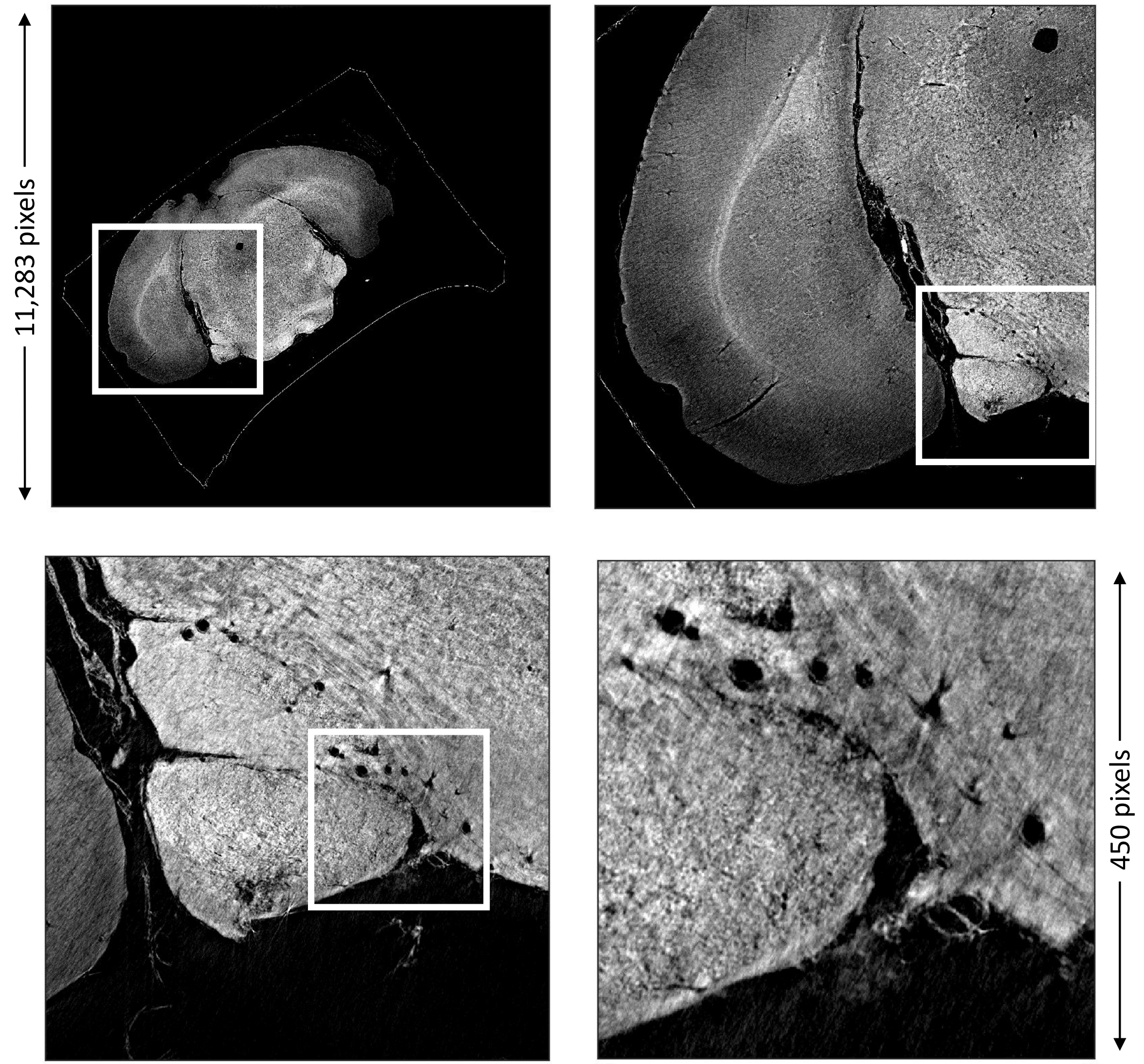
Backprojection Minibatch





Hierarchical Communications: Scaling on Summit





We thank Bobby Kasthuri of UChicago/Argonne, Rafael Vescovi and Ming Du of Argonne for sharing the mouse brain dataset.

Recognition: Best Paper @ SC20

SC20 Student Cluster Reproducibility Committee Chooses Benchmark Wisely

April 15, 2020

https://github.com/merthidayetoglu/MemXCT-GPU

We are excited to announce that the SC20 Reproducibility Committee has selected the SC19 paper *"MemXCT: Memory-Centric X-ray CT Reconstruction with Massive Parallelization"*, by Mert Hidayetoğlu, Tekin Biçer, Simon Garcia de Gonzalo, Bin Ren, Doğa Gürsoy, Rajkumar Kettimuthu, Ian T. Foster, and Wen-mei W. Hwu, to serve as the Student Cluster Competition (SCC) benchmark for the Reproducibility Challenge this year. A team of reviewers selected the paper from 45 finalists, based on the paper's Artifact Descriptor (AD) and its suitability to the SCC. The authors and the Reproducibility Committee have been working to create a reproducible benchmark that builds on the paper's results. At SC20, the sixteen SCC teams will be asked to run the benchmark, replicating the findings from the original paper under different settings and with different datasets.

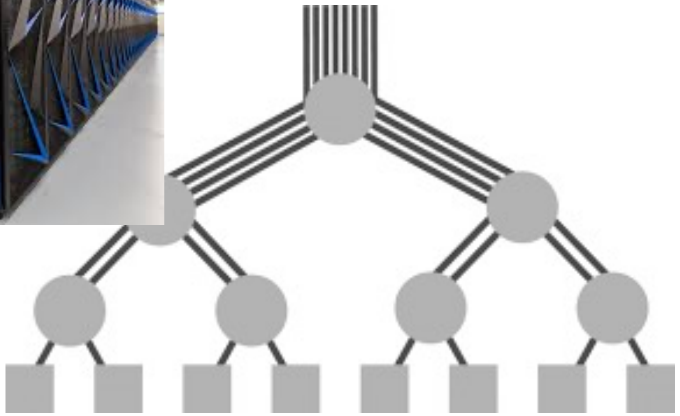
STUDENT RESEARCH COMPETITION

Association for Computing Machinery
Advancing Computing as a Science & Profession

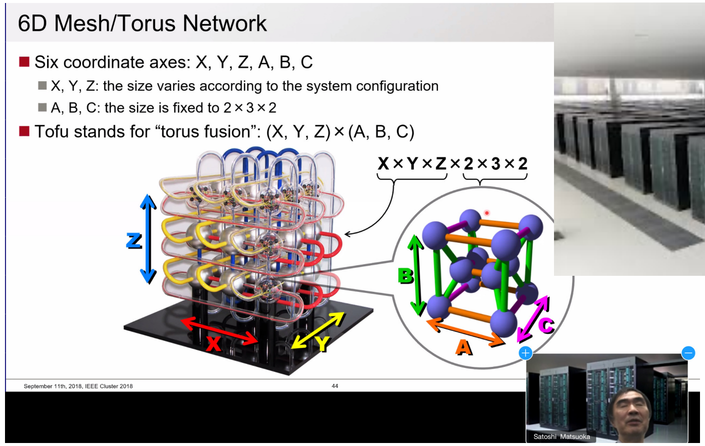
SPONSORED BY Microsoft

Open Problems: Extending Comm. Hierarchy for Inter-Node Communication

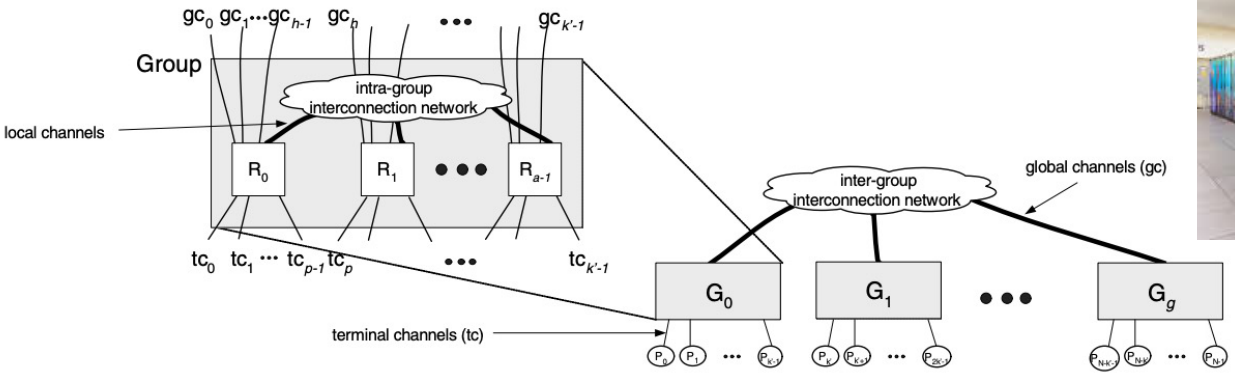
Fat-Tree: OLCF Summit



Tofu: RIKEN Fugaku



Dragonfly: ALCF Theta



Conclusions

Large-Scale Sparse Applications

Conclusions

Large-Scale Sparse Applications

Multi-GPU Node Architecture → Single GPU: Tiled SpMM

Multi-GPU Node Architecture → GPU Cluster: Hierarchical Comm.

Conclusions

Large-Scale Sparse Applications

Multi-GPU Node Architecture

Single GPU: Tiled SpMM

GPU Cluster: Hierarchical Comm.

Possible Contributions: cuSPARSE, MPI Collectives, NCCL

Conclusions

Large-Scale Sparse Applications

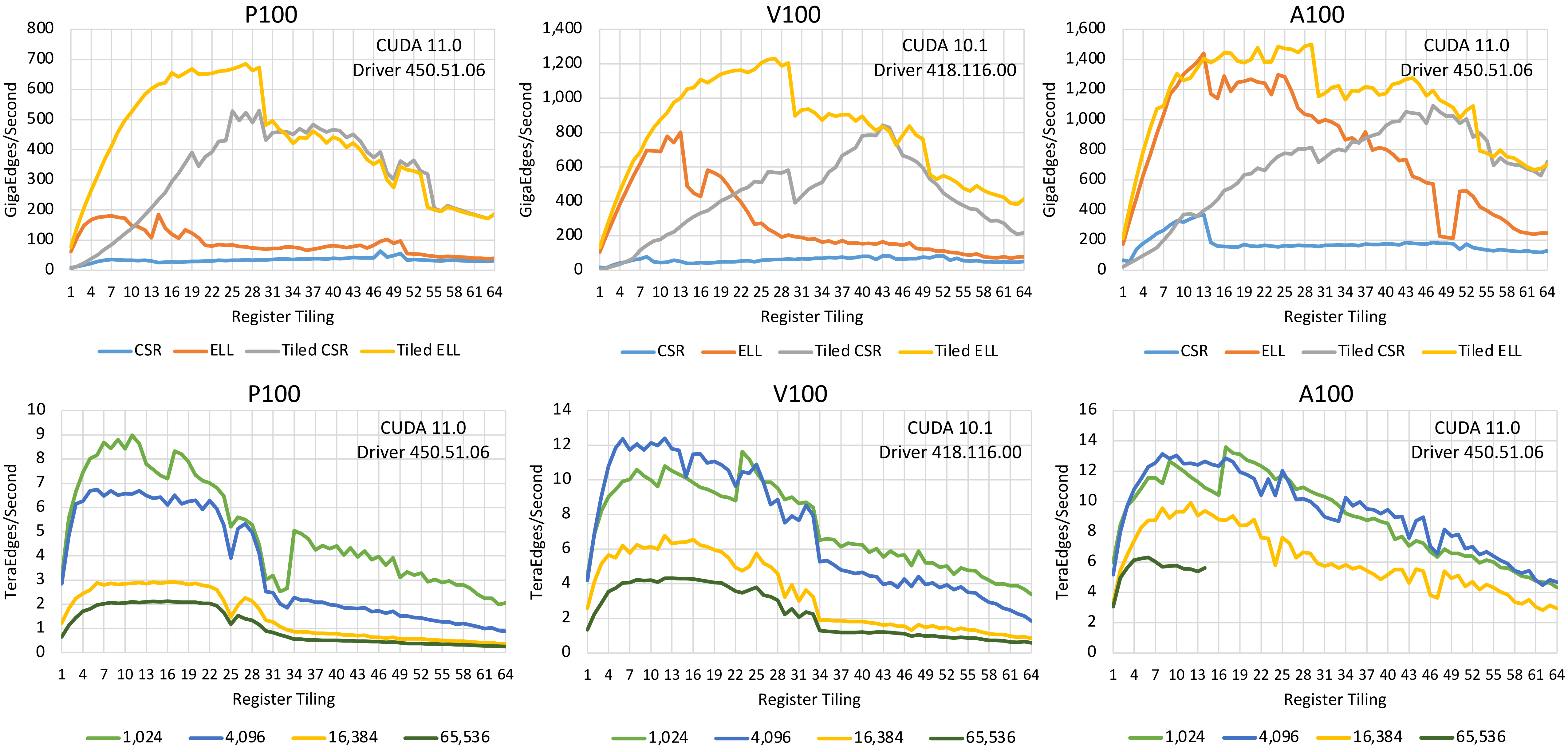
Multi-GPU Node Architecture

Single GPU: Tiled SpMM

GPU Cluster: Hierarchical Comm.

Possible Contributions: cuSPARSE, MPI Collectives, NCCL

Further Dissertation Research


P100
CUDA 11.0
Driver 450.51.06
GigaEdges/Second
Register Tiling
CSR
ELL
Tiled CSR
Tiled ELL
V100
CUDA 10.1
Driver 418.116.00
GigaEdges/Second
Register Tiling
CSR
ELL
Tiled CSR
Tiled ELL
A100
CUDA 11.0
Driver 450.51.06
GigaEdges/Second
Register Tiling
CSR
ELL
Tiled CSR
Tiled ELL
P100
CUDA 11.0
Driver 450.51.06
TeraEdges/Second
Register Tiling
1,024
4,096
16,384
65,536
V100
CUDA 10.1
Driver 418.116.00
TeraEdges/Second
Register Tiling
1,024
4,096
16,384
65,536
A100
CUDA 11.0
Driver 450.51.06
TeraEdges/Second
Register Tiling
1,024
4,096
16,384
65,536