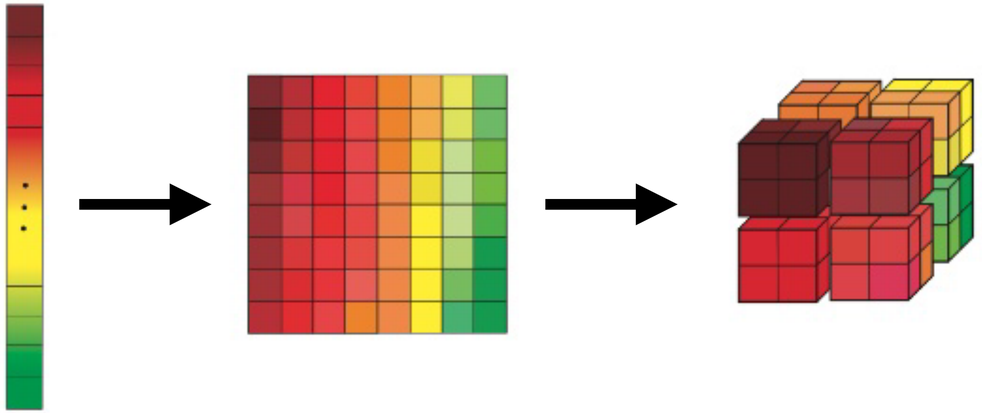


Sparse Tensor Algebra and its Relationship to Matrix and Graph Problems

Jiajia Li

Pacific Northwest National Laboratory

Feb 28, 2019 @ SIAM CSE'19

U.S. DEPARTMENT OF ENERGY

Office of Science

Pacific Northwest NATIONAL LABORATORY

Figure sources: "A brief survey of tensors" by Berton Earnshaw and NVIDIA Tensor Cores

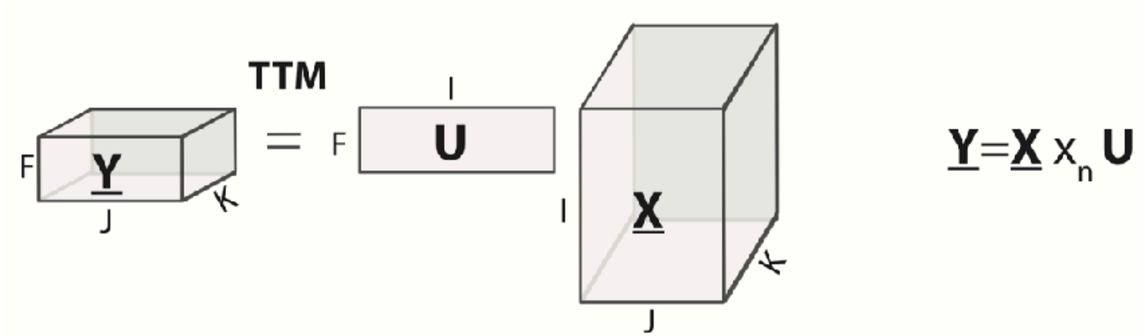
Overview

- Sparse Tensor Algebra <-> Sparse matrix Algebra
- Sparse Tensor Algebra <-> Graph Algebra
- Sparse Tensor Recent Development
  - HiCOO: sparse tensor format
  - PASTA: sparse tensor benchmark suite

Sparse Tensor Algebra <—> Sparse matrix Algebra

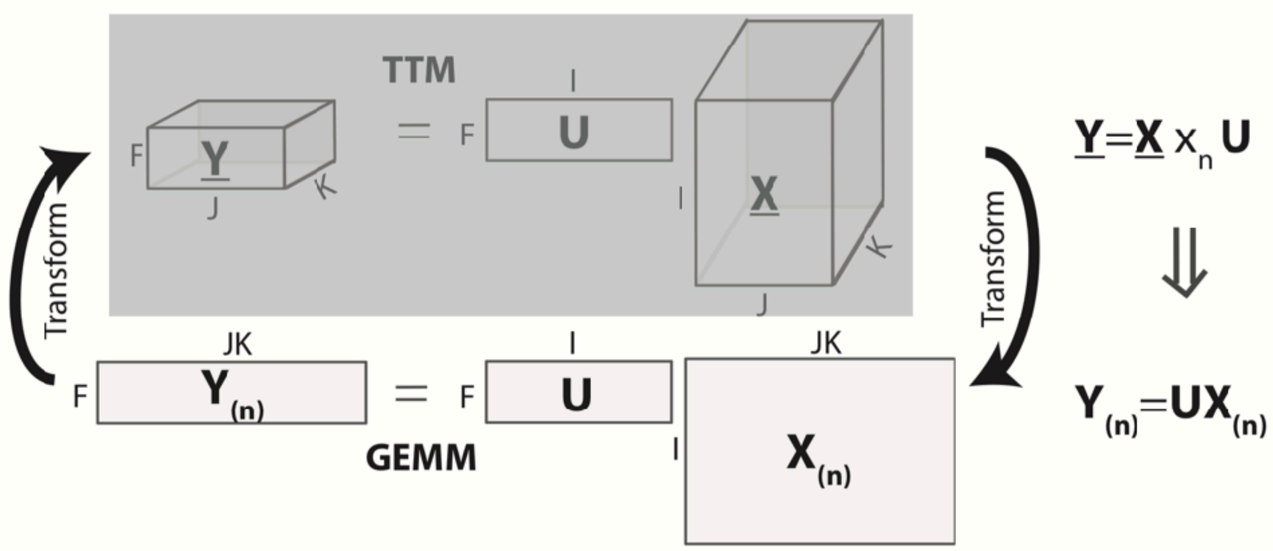
- Sparse tensor algebra <- sparse matrix algebra
  - Convert a tensor problem to an equivalent matrix/vector problem
  - Then use matrix algebra to solve
- Sparse matrix algebra <- sparse tensor algebra
  - Expose low-rank features of a matrix/vector by a tensor decomposition
  - Then solve a reduced matrix problem more efficiently

Sparse tensor algebra <- sparse matrix algebra



Dense Tensor-Times-Matrix Multiply (TTM)

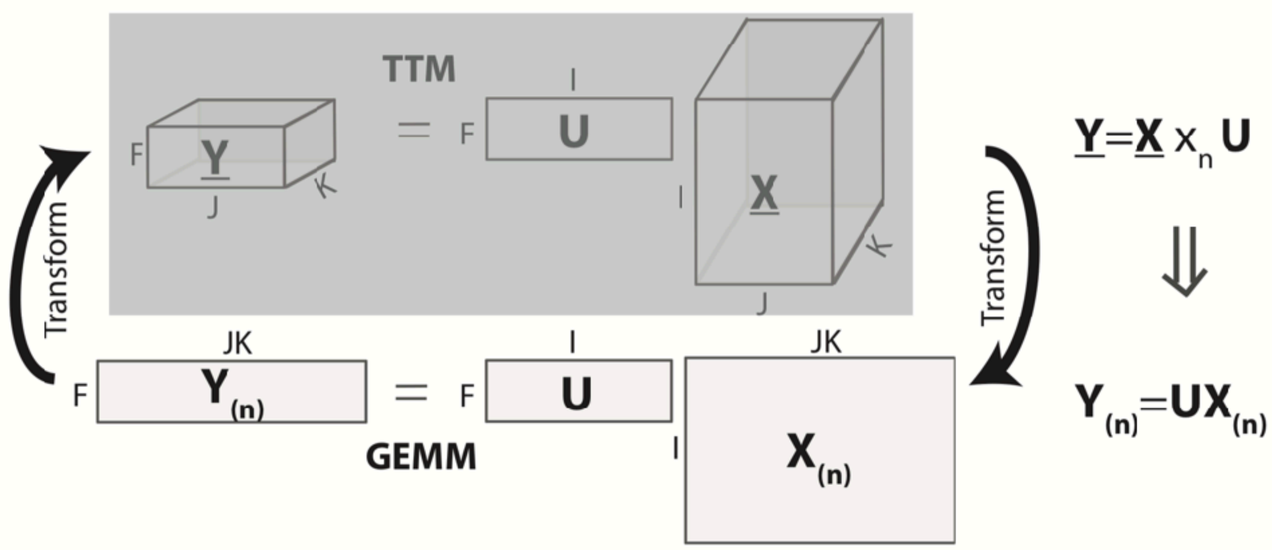
Sparse tensor algebra <- sparse matrix algebra



Dense Tensor-Times-Matrix Multiply (TTM)

- Why?
  - Use efficient matrix algebra libraries.
- Issues
  - Extra storage and computation
  - Fast transformation or merge-together: TTC, Batched BLAS, TBLIS, etc.

Sparse tensor algebra <- sparse matrix algebra



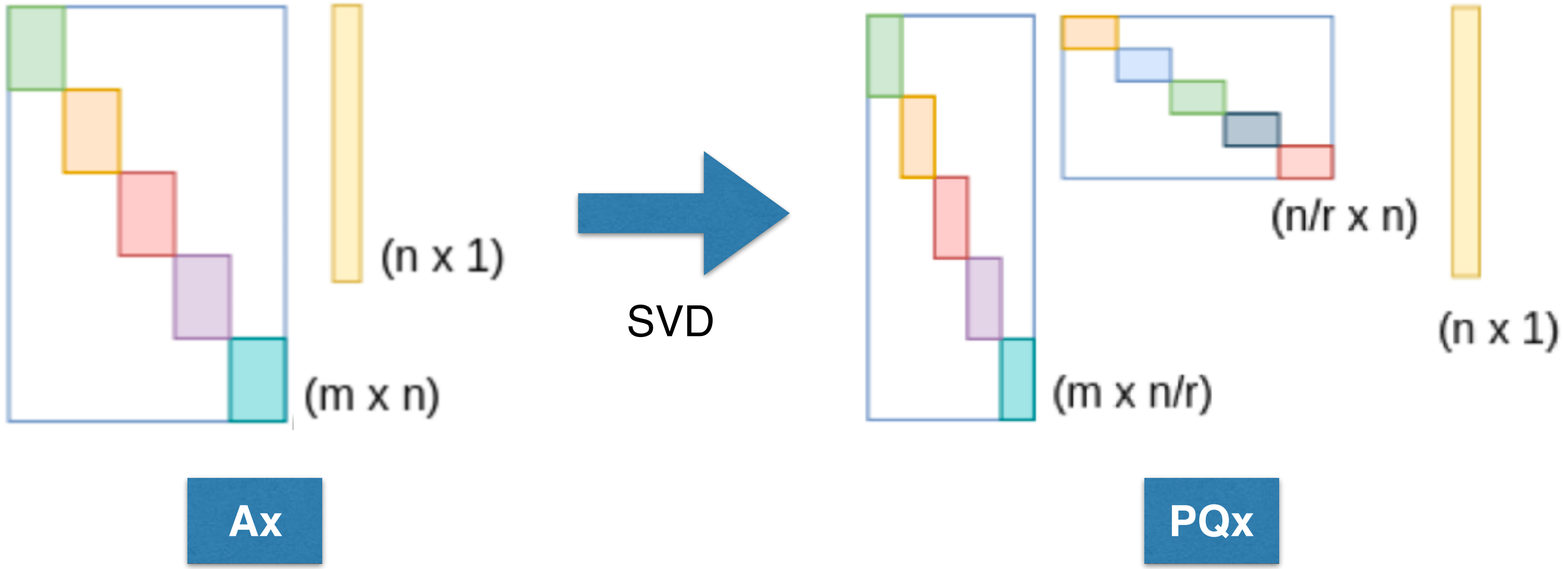
Dense Tensor-Times-Matrix Multiply (TTM)

- Why?
  - Use efficient matrix algebra libraries.
- Issues
  - Extra storage and computation
  - Fast transformation or merge-together: TTC, Batched BLAS, TBLIS, etc.

Similar for sparse case

- Why?
  - More algorithms for sparse matrix algebra
- Issues
  - Extra storage and computation
  - It is harder to do
    - fast transformation: sorting is more expensive.
    - merge-together: need more data structures designed.

Sparse matrix algebra <- sparse tensor algebra



- Explore the low-rank feature inside.
- After SVD or tensor decomposition, we use the output to do less expensive computation.
- Applications: math, deep learning, etc.

Source: “ANTMAN: SPARSE LOW-RANK COMPRESSION TO ACCELERATE RNN INFERENCE”. 2019

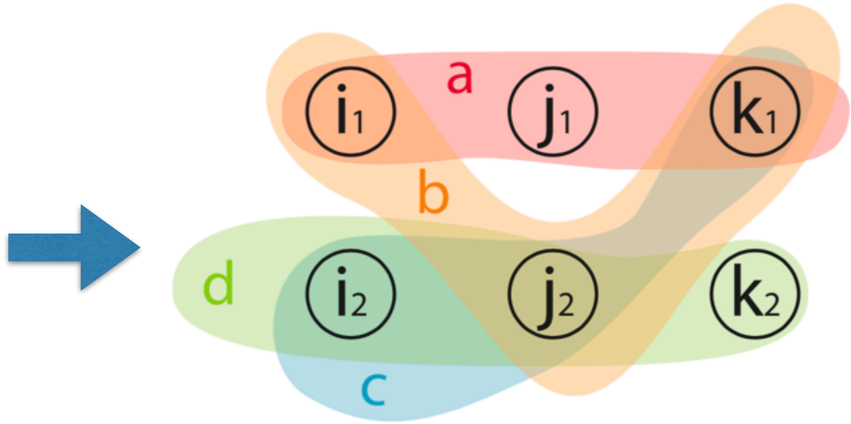
Sparse Tensor Algebra <-> Graph Algebra

- Sparse tensor algebra <- Graph algebra
  - Graph problems exist in tensor algebra, e.g., finding a better nonzero ordering
- Graph algebra <- sparse tensor algebra
  - Multi-attribute graphs can be represented as tensors naturally.

Sparse tensor algebra <- Graph algebra

i	j	k	val
i_1	j_1	k_1	a
i_1	j_2	k_1	b
i_2	j_2	k_1	c
i_2	j_2	k_2	d

(a) Tensor

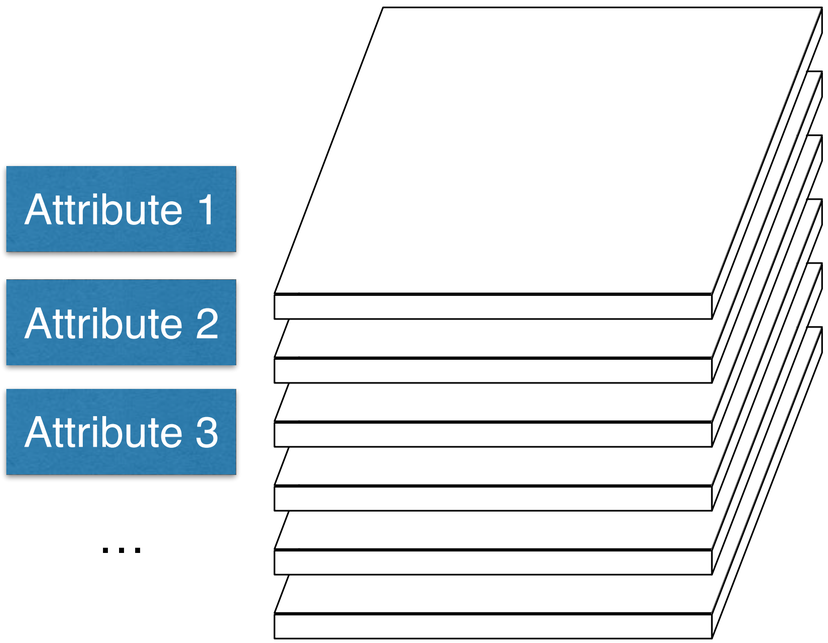


(b) Hypergraph

- Find the index map for tensor reordering.
- Pursue better locality of nonzeros and then better cache behavior for tensor algorithms.

- Efficient and Effective Sparse Tensor Reordering. Jiajia Li, Bora Ucar, Umit Catalyurek, Kevin Barker, Richard Vuduc. Technical Report. 2019
- SPLATT: Efficient and parallel sparse tensor-matrix multiplication. Shaden Smith, Niranjay Ravindran, Nicholas D Sidiropoulos, and George Karypis. IPDPS'15.
- Scalable Sparse Tensor Decompositions in Distributed Memory Systems, O. Kaya and B. Uçar, SC '15.

Graph algebra <- Sparse tensor algebra



- Use tensor decompositions to do clustering and other unsupervised learning to reveal hidden relationship.

Sparse Tensor Recent Development

- HiCOO: sparse tensor format
- PASTA: sparse tensor benchmark suite

HiCOO Format

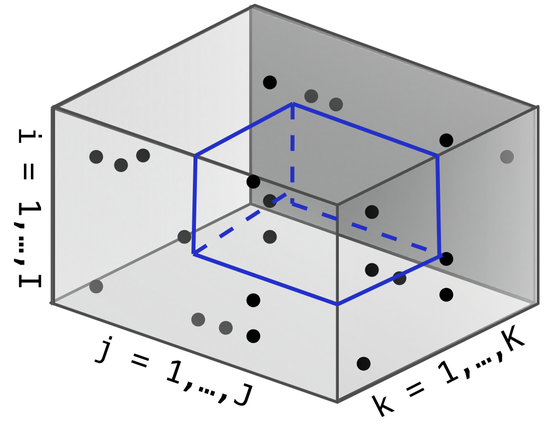
- Store a sparse tensor in units of small sparse blocks

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

COO

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO

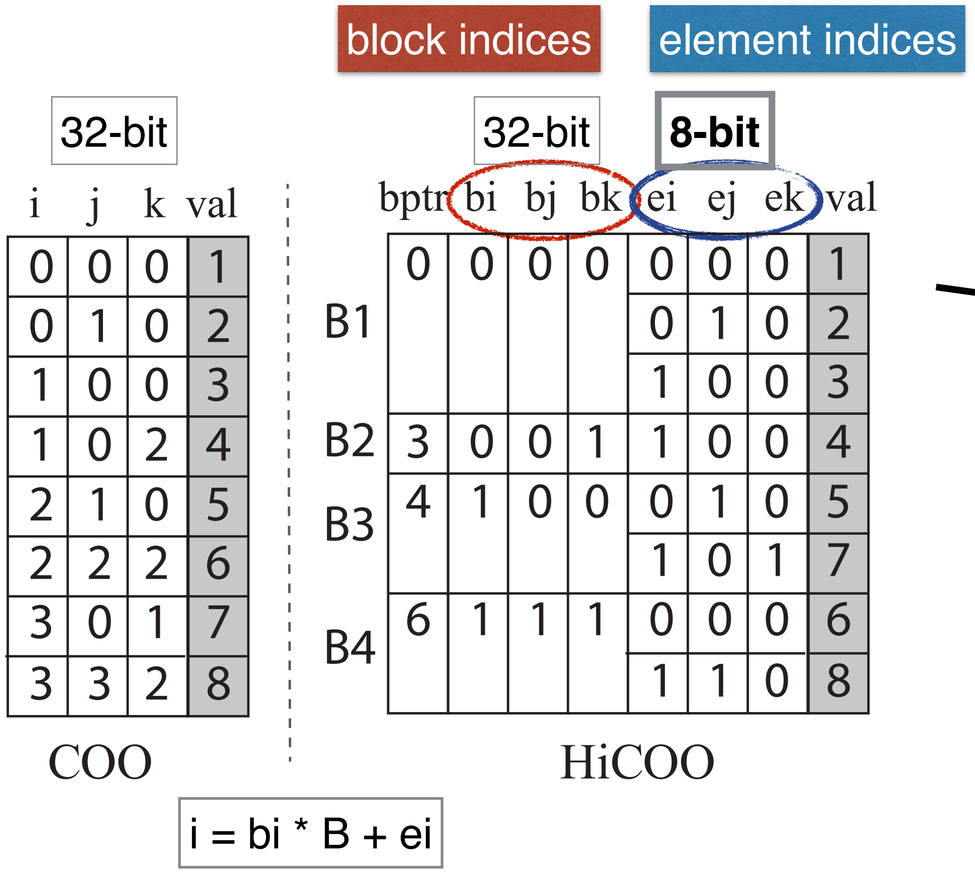


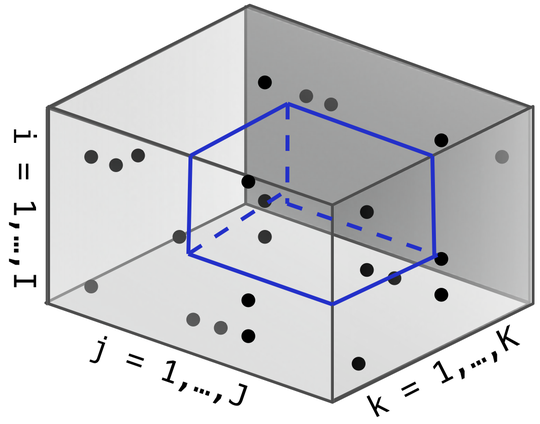
Block size: 2*2*2

Extension from Compressed Sparse Blocks (CSB) format by Buluc et al. SPAA. 2009.

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices

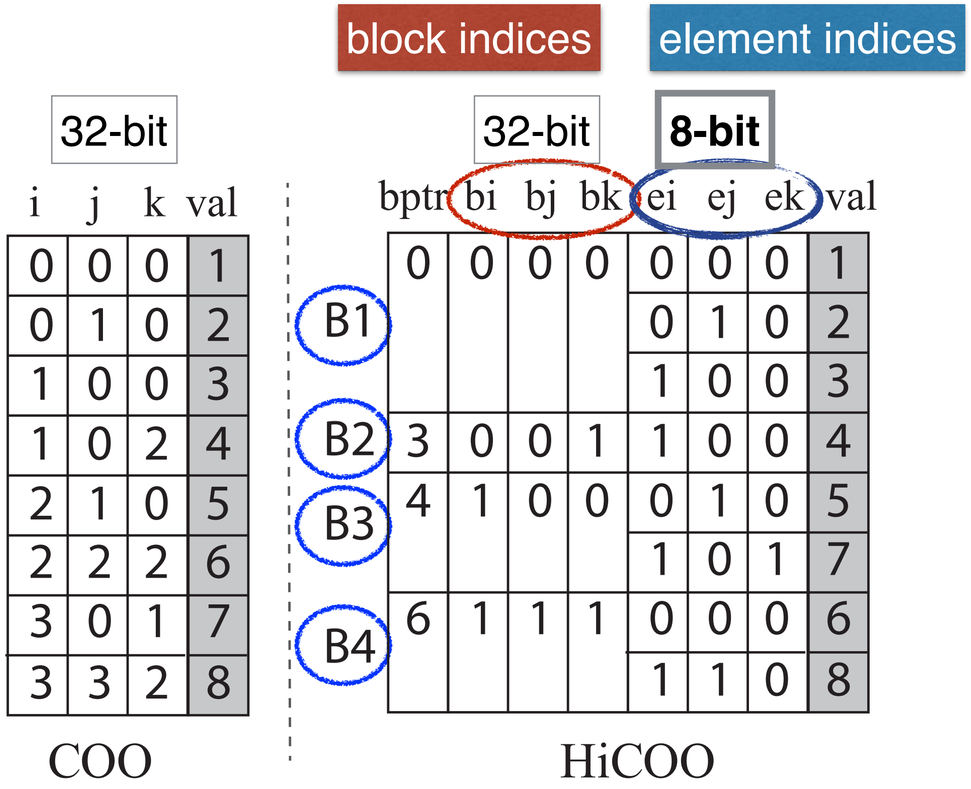




Block size: 2*2*2

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices



i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

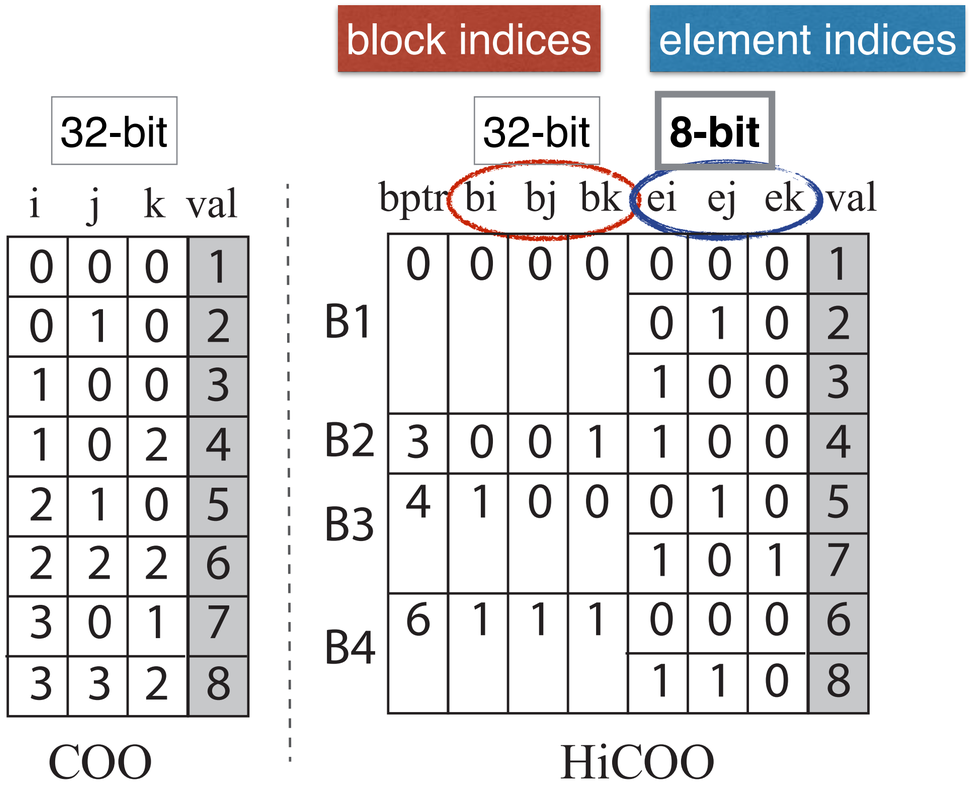
COO

bptr	bi	bj	bk	ei	ej	ek	val
0	0	0	0	0	0	0	1
				0	1	0	2
				1	0	0	3
3	0	0	1	1	0	0	4
4	1	0	0	0	1	0	5
				1	0	1	7
6	1	1	1	0	0	0	6
				1	1	0	8

HiCOO

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices



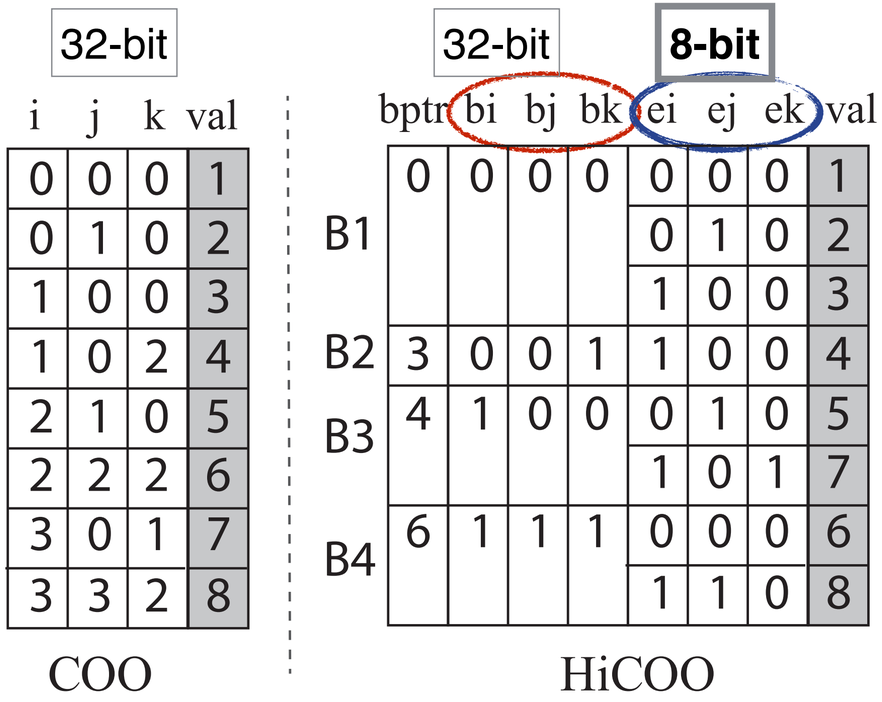
COO indices:
= nnz * 3 * **32**

HiCOO indices:
= nnz * 3 * **8** + **nnb** * (3 * 32 + 32)

nnz: #Nonzeros; nnb: #Non-zero blocks

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices
  - For arbitrary-order sparse tensors.



For the tensor: Reduce its storage and memory footprints

For matrices: Better data locality

Platform and Dataset

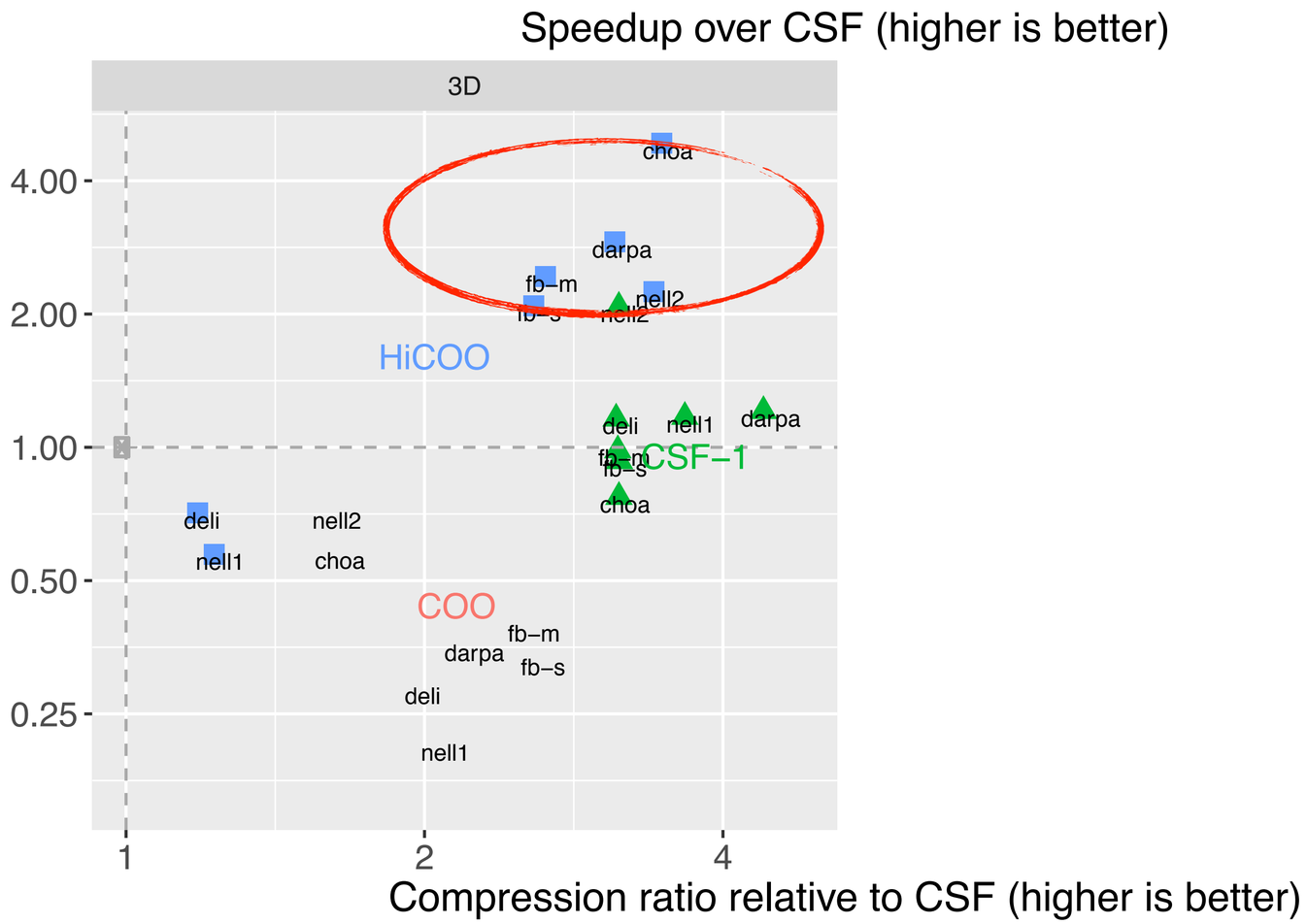
- **Platform**: Intel Xeon CPU E7-4850 v3 platform consisting 56 physical cores with icc 18.0.2 and parallelized by OpenMP.
- **Dataset**: FROSTT [Smith et al. 2017], HaTen2 [Jeon et al. 2015], and healthcare data [Perros et al. 2017].

DESCRIPTION OF SPARSE TENSORS.

Tensors	Order	Dimensions	#Nonzeros	Density
nell2	3	$12K \times 9K \times 29K$	77M	2.4×10^{-5}
choa	3	$712K \times 10K \times 767$	27M	5.0×10^{-6}
darpa	3	$22K \times 22K \times 24M$	28M	2.4×10^{-9}
fb-m	3	$23M \times 23M \times 166$	100M	1.1×10^{-9}
fb-s	3	$39M \times 39M \times 532$	140M	1.7×10^{-10}
deli	3	$533K \times 17M \times 2.5M$	140M	6.1×10^{-12}
nell1	3	$3M \times 2M \times 25M$	144M	9.1×10^{-13}
crime	4	$6K \times 24 \times 77 \times 32$	5M	1.5×10^{-2}
nips	4	$2K \times 3K \times 14K \times 17$	3M	1.8×10^{-6}
enron	4	$6K \times 6K \times 244K \times 1K$	54M	5.5×10^{-9}
flickr	4	$320K \times 28M \times 2M \times 731$	113M	1.1×10^{-14}
deli4d	4	$533K \times 17M \times 2M \times 1K$	140M	4.3×10^{-15}

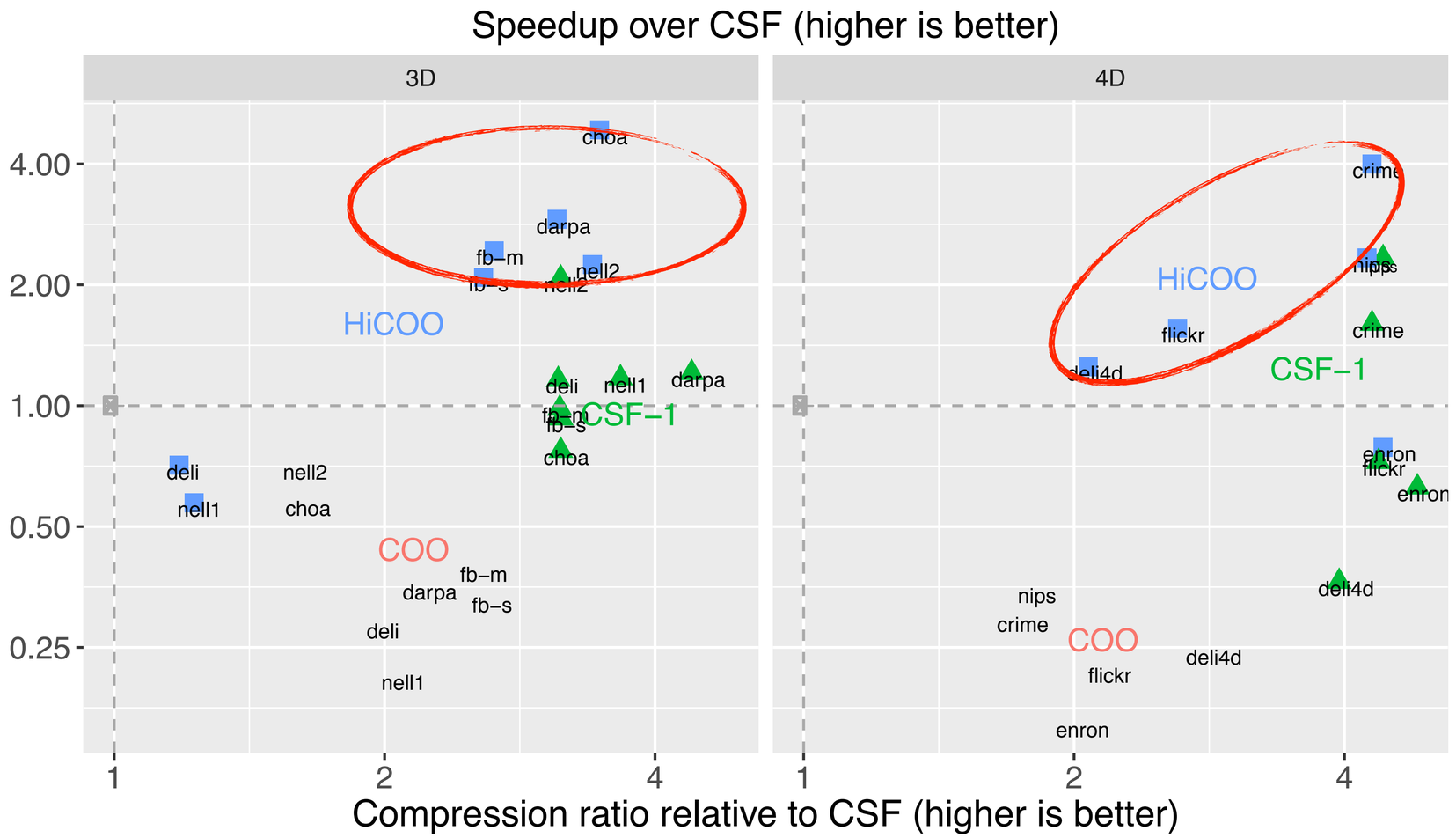
Multicore CP-ALS

- HiCOO outperforms COO by 6.2× and CSF by 2.1× on average.



Multicore CP-ALS

- HiCOO outperforms COO by 6.2× and CSF by 2.1× on average.



HiCOO: Next Steps

- Shorten pre-processing time
- Make it more flexible and scalable
- Include more tensor methods.
- Accelerate on more architectures.

PASTA: A Sparse Tensor Benchmark Suite

Data Structures/ Algorithms	Platforms	TEW (Element-Wise)	TS (Tensor-scalar)	TTV (Tensor-Times-Vector)	TTM (Tensor-Times-Matrix)	MTTKRP (Matriced Tensor-Times-Khatri-Rao Product)
COO	Single-core CPUs	✓	✓	✓	✓	✓
	Multi-core CPUs	✓	✓	✓	✓	✓

PASTA Workloads

Arbitrary shape and nonuniform nonzero pattern

Data Structures/ Algorithms	Platforms	TEW (Element-Wise)	TS (Tensor-scalar)	TTV (Tensor-Times-Vector)	TTM (Tensor-Times-Matrix)	MTTKRP (Matriced Tensor-Times-Khatri-Rao Product)
COO	Single-core CPUs	✓	✓	✓	✓	✓
	Multi-core CPUs	✓	✓	✓	✓	✓

PASTA Workloads

		Parallelize nonzero partitions	Parallelize nonzeros	Parallelize nonzero fibers		Parallelize nonzeros with atomics
Data Structures/ Algorithms	**Platforms**	**TEW** (Element-Wise)	**TS** (Tensor-scalar)	**TTV** (Tensor-Times-Vector)	**TTM** (Tensor-Times-Matrix)	**MTTKRP** (Matriced Tensor-Times-Khatri-Rao Product)
COO	Single-core CPUs	✓	✓	✓	✓	✓
	Multi-core CPUs	✓	✓	✓	✓	✓

Memory-Bound Workloads

Table 2. The analysis of data storage and their algorithms for third-order cubical tensors ($\mathcal{X} \in \mathbb{R}^{I \times I \times I}$). We consider all input tensors with M nonzero entries and M_F fibers, $I \ll M_F \ll M$. The indices use 32 bits, and values are single-precision floating-point numbers with 32 bits.

Workloads	Storage (Bytes)	Work (Flops)	Memory Access (Bytes)	Arithmetic Intensity (AI)
TEW	$48M$	M	$36M$	$1/36$
TS	$32M$	M	$32M$	$1/32$
TTV	$(16M + 12M_F)$	$2M$	$(12M + 20M_F)$	$\sim 1/6$
TTM	$(16M + 16M_F R + 4IR)$	$2MR$	$4MR + 8M + 12M_F R + 8M_F$	$\sim 1/2$
MTTKRP	$(16M + 12IR)$	$3MR$	$12MR + 16M$	$\sim 1/4$

PASTA: Next Step

Data Structures/ Algorithms	Platforms	TEW (Element-Wise)	TS (Tensor-scalar)	TTV (Tensor-Times-Vector)	TTM (Tensor-Times-Matrix)	MTTKRP (Matriced Tensor-Times-Khatri-Rao Product)	TTV (Tensor-Times-Sparse-Vector)	TTM (Tensor-Times-Sparse-Matrix)	MTTKRP (Matriced Tensor-Times-Sparse-Khatri-Rao Product)
COO	Single-core CPUs	✓	✓	✓	✓	✓	✗	✗	✗
	Multi-core CPUs	✓	✓	✓	✓	✓	✗	✗	✗
	GPUs	✓	✓	✗	✓	✓	✗	✗	✗
HiCOO	Single-core CPUs	✓	✓	✗	✗	✓	✗	✗	✗
	Multi-core CPUs	✗	✗	✗	✗	✓	✗	✗	✗
	GPUs	✗	✗	✗	✗	✓	✗	✗	✗

Sparse Tensor Algebra and its Relationship to Matrix and Graph Problems

Take Aways:

- Close relationship: Tensor <-> Matrix <-> Graph
- HiCOO: a sparse tensor format
  - Code: https://github.com/hpcgarage/ParTI (v1.0.0)
- PASTA: a sparse tensor benchmark suite
  - Code: https://gitlab.com/tensorworld/pasta

Related talks: MS305: “On Tensor Orderings for HiCOO”

Other tensor talks: MS249. MS305. MS337. MS370

U.S. DEPARTMENT OF ENERGY
Office of Science

Jiajia Li
Pacific Northwest National Laboratory
Feb 28, 2019 @ SIAM CSE’19

Pacific Northwest NATIONAL LABORATORY

Figure sources: “A brief survey of tensors” by Berton Earnshaw and NVIDIA Tensor Cores

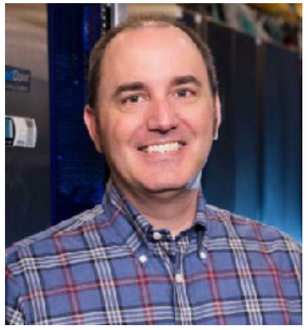
Collaborators

Richard Vuduc
Associate Professor
GaTech

Jimeng Sun
Associate Professor
GaTech

Bora Ucar
CNRS researcher
ENS-LYON

Umit Catalyurek
Professor
GaTech



Kevin Barker
Computer Scientist
PNNL

Ang Li
Computer Scientist
PNNL

Cathie Olschanowsky
Assistant Professor
Boise State University

Yuchen Ma
Undergrad, HDU

Acknowledgement