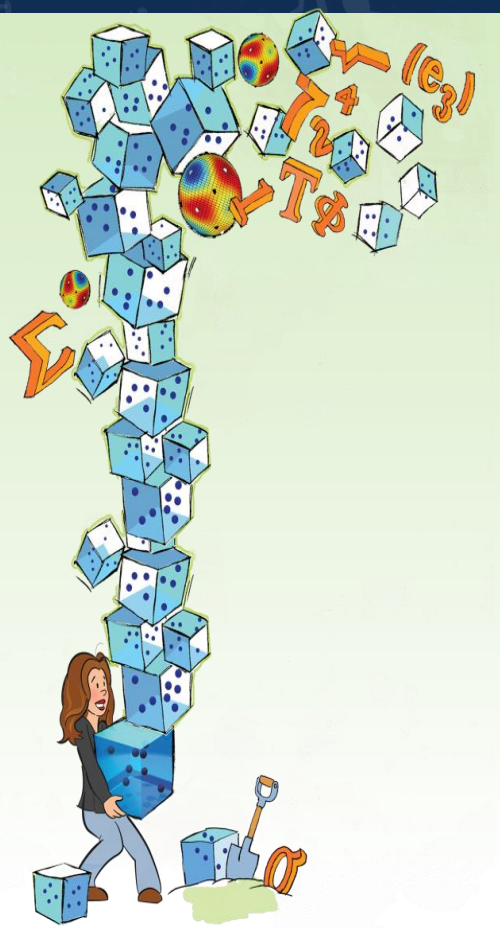
SAND2016-11901C

Exceptional service in the national interest

Sandia National Laboratories



Parallel Multiway Methods for Compression of Massive Data and Other Applications

Tamara G. Kolda[1], Woody Austin[1,2], Grey Ballard[1,3], Alicia Klinvex[1], Hemanth Kolla[1]

[1]Sandia National Laboratories, Livermore, CA
[2]University of Texas, Austin
[3]Wake Forest University, North Carolina

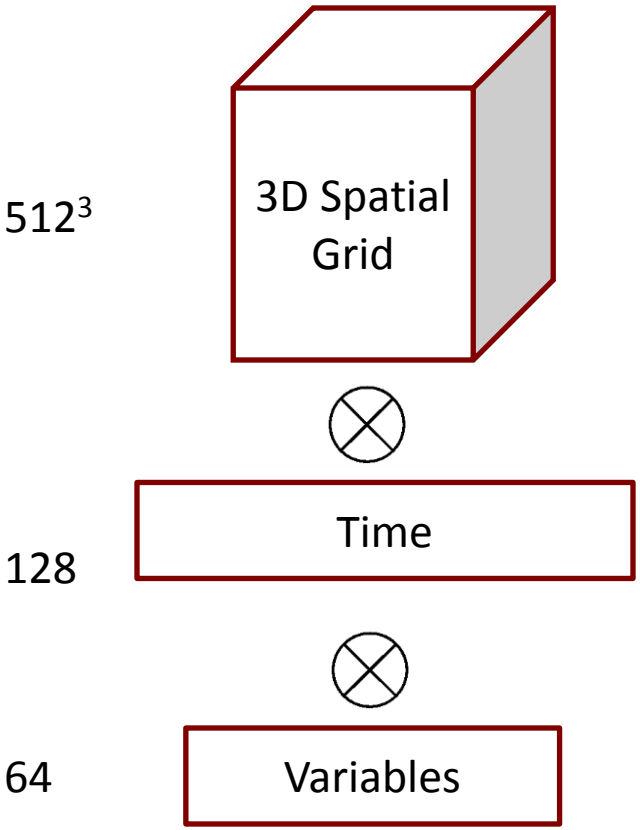
Supercomputing '16, Salt Lake City, November 17, 2016

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Motivation: Advanced Simulations and Experiments Deluged by Data

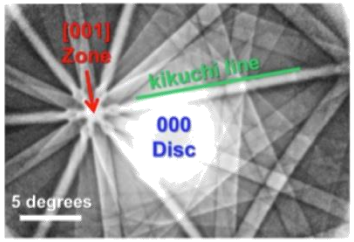
Sandia National Laboratories

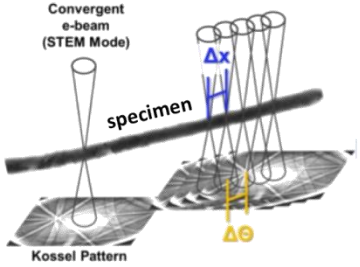


2^{40} elements

8TB (double precision)

- Combustion simulations
  - S3D code uses direct numerical simulation
  - Gold standard for comparisons, but…
  - Single experiment produces terabytes of data
  - Storage limits spatial, temporal resolutions
  - Difficult to analyze or transfer data
- Electron microscopy
  - New technology produces 1-D spectra and 2-D diffraction patterns *at each pixel*
  - Single experiment produces terabytes of data
  - Usually 10-100 experiments per
  - Limited hardware utilization due to storage limits

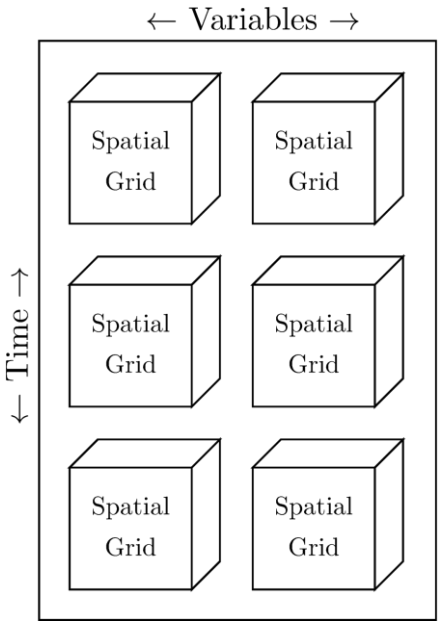




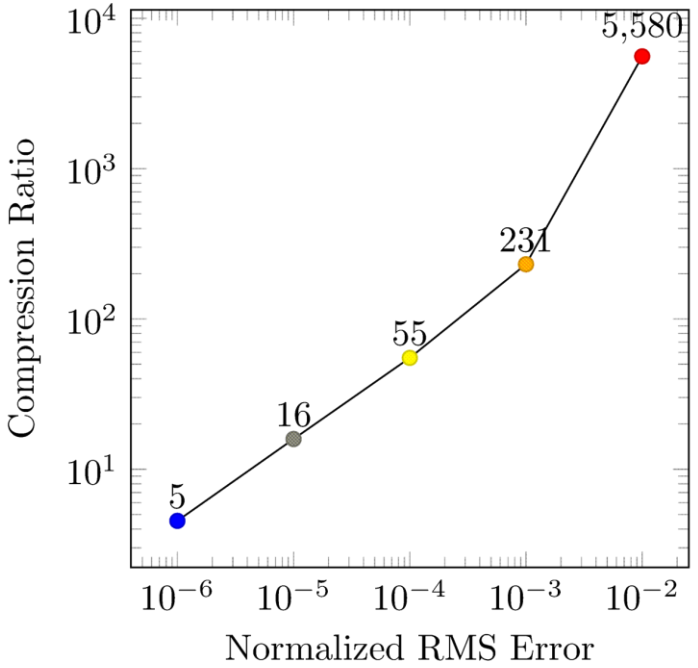
- Other applications
  - Telemetry
  - Cosmology Simulations
  - Climate Modeling

Claim: Tensor Tucker Compression Yields Up to 5000X Data Reduction

Sandia National Laboratories



Natural five-way multiway structure of scientific data



Compression rates as fidelity varies for 550GB simulation dataset

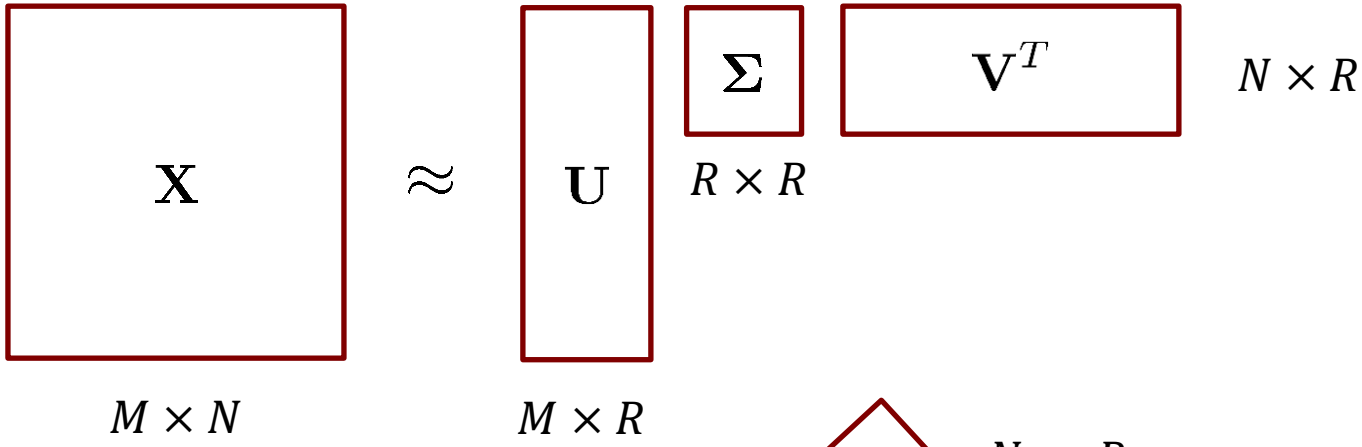
Our contributions:

- Distributed implementation of Tucker compression
- Demonstration of good compression on combustion data sets

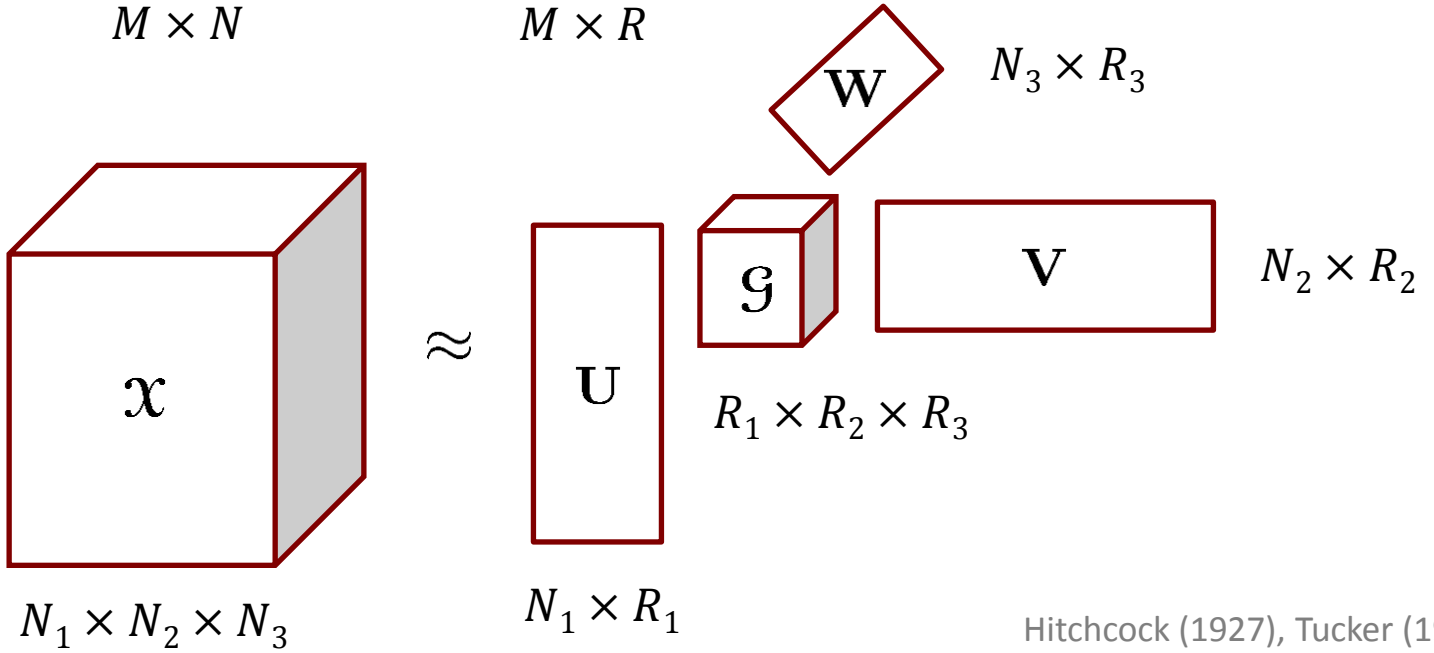
Tucker Compression: Extends the Matrix SVD to Multiway Arrays

Sandia National Laboratories

Matrix SVD

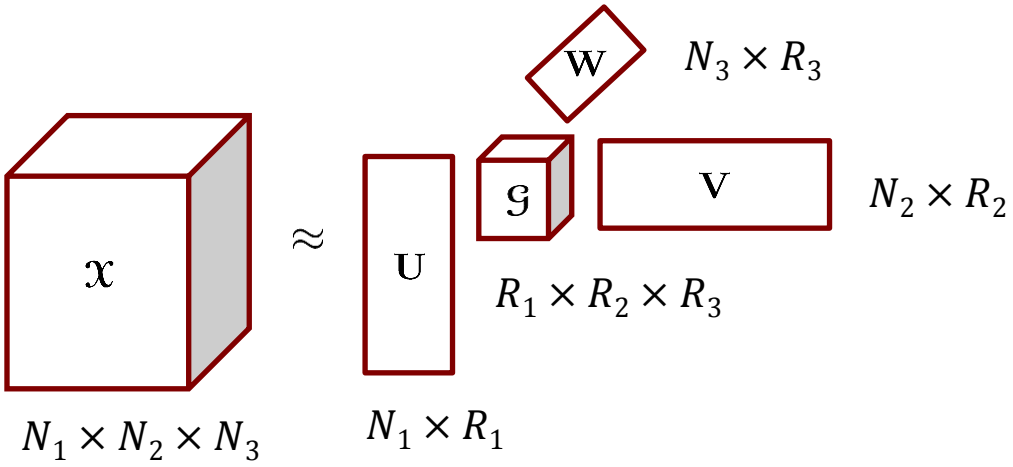


Three-way Tucker



Hitchcock (1927), Tucker (1966)

Tucker Compression (3-way)



$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

$\mathcal{G}$ = "Core Tensor" = Reduced representation, determined by factor matrices

$\mathbf{U}, \mathbf{V}, \mathbf{W}$ = "Factor Matrices" = Orthogonal matrices spanning high-variance subspaces

$\times_k$ = Tensor-times-matrix in mode k

Detail: $$x(i_1, i_2, i_3) \approx \sum_{j_1, j_2, j_3} g(j_1, j_2, j_3)\, u(i_1, j_1)\, v(i_2, j_2)\, w(i_3, j_3)$$

Tucker Compression (d-way)

$$\underbrace{\mathcal{X}}_{N_1 \times N_2 \cdots \times N_d} \approx \underbrace{\mathcal{G}}_{R_1 \times R_2 \cdots \times R_d} \times_1 \underbrace{\mathbf{U}^{(1)}}_{N_1 \times R_1} \times_2 \underbrace{\mathbf{U}^{(2)}}_{N_2 \times R_2} \cdots \times_d \underbrace{\mathbf{U}^{(d)}}_{N_d \times R_d}$$

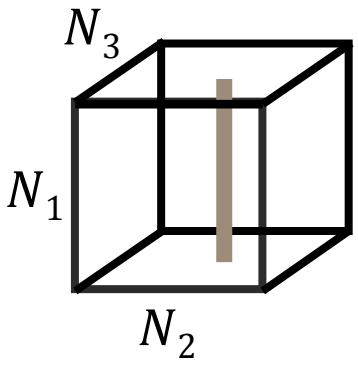
$\mathcal{G}$ = "Core Tensor" = Reduced representation, determined by factor matrices

$\mathbf{U}^{(k)}$ = kth "Factor Matrix" = Orthogonal matrix spanning high-variance subspaces

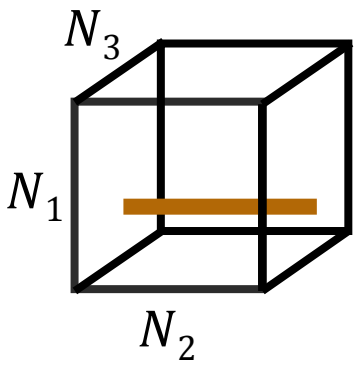
$\times_k$ = Tensor-times-matrix in mode k

Compression Ratio: $C \approx \prod_{k=1}^{d} \frac{N_k}{R_k}$

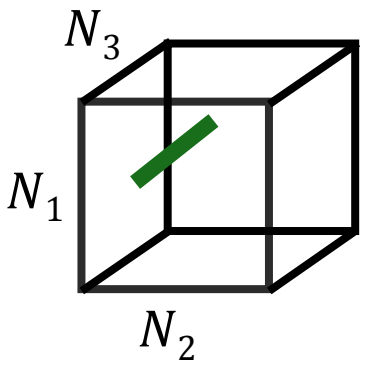
Matricization/Unfolding: Rearranging a Tensor as a Matrix



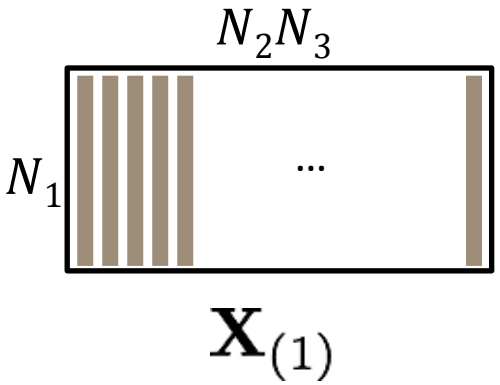
mode-1 fibers

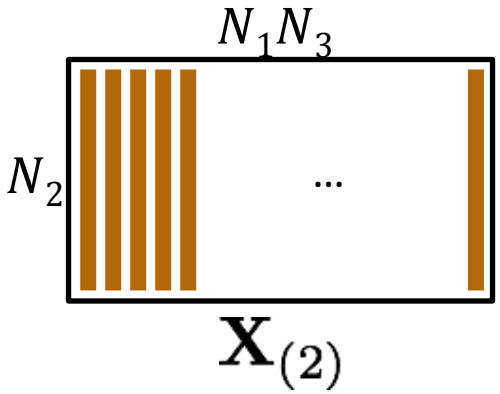


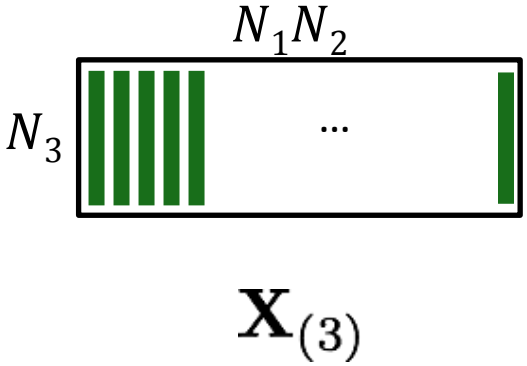
mode-2 fibers



mode-3 fibers

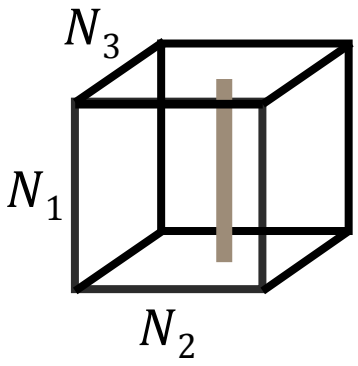




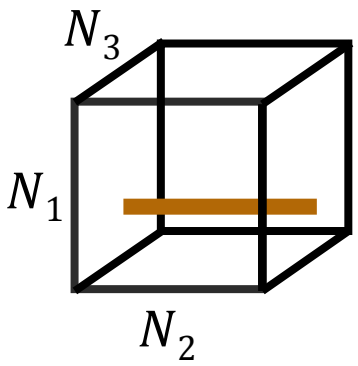


Mathematically convenient expression, but do not want to explicitly form the unfolded matrix due to the expense of the *data movement*.

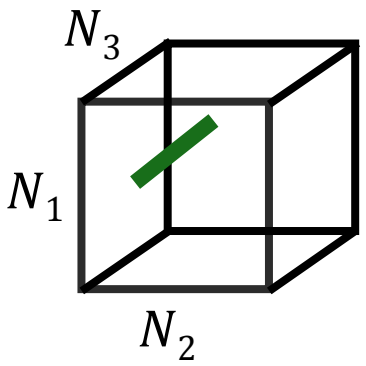
Tensor Times Matrix (in Mode 1)



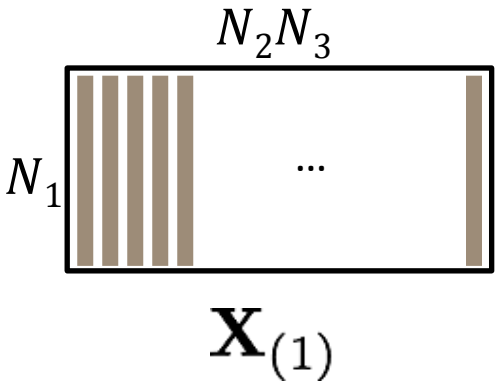
mode-1 fibers

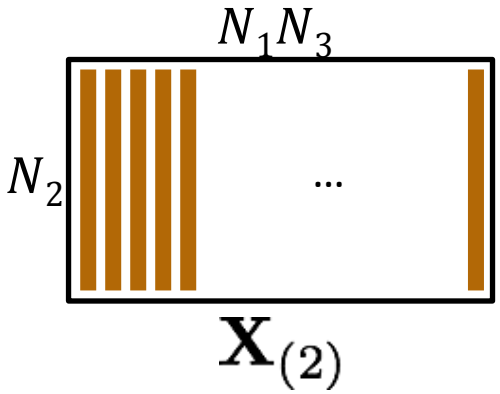


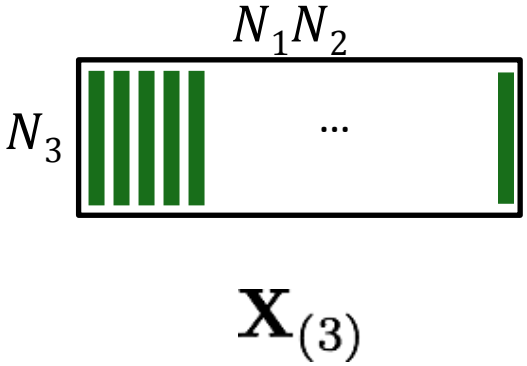
mode-2 fibers



mode-3 fibers

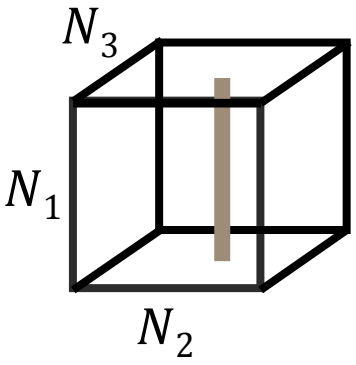




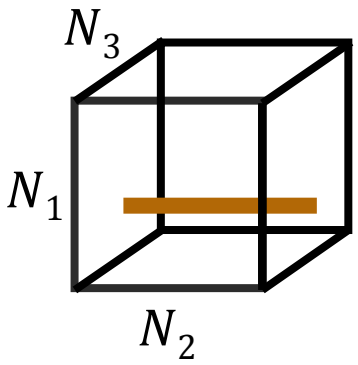


Mathematically convenient expression, but do not want to explicitly form the unfolded matrix due to the expense of the *data movement*.

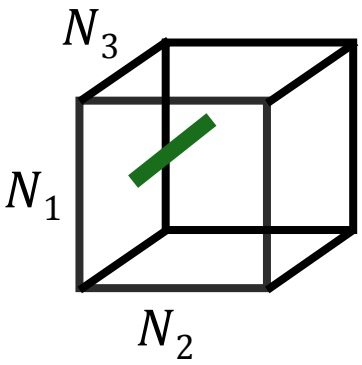
Tensor Times Matrix (in Mode 1)



mode-1 fibers



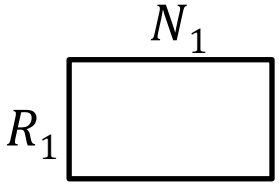
mode-2 fibers



mode-3 fibers

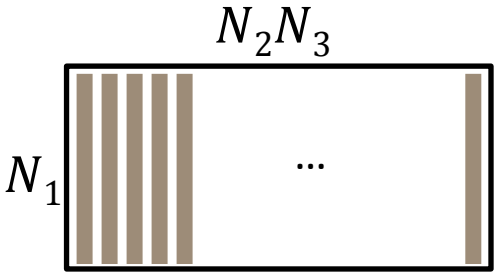
$N_1 \times N_2 \times N_3$

$\mathcal{X} \times_1 \mathbf{U}'$ ⇨



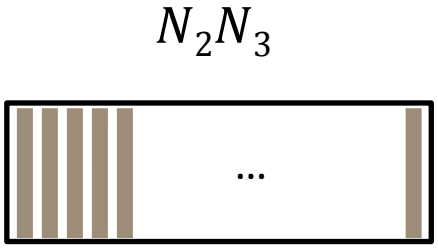
$R_1 \times N_1$

$\mathbf{U}'$

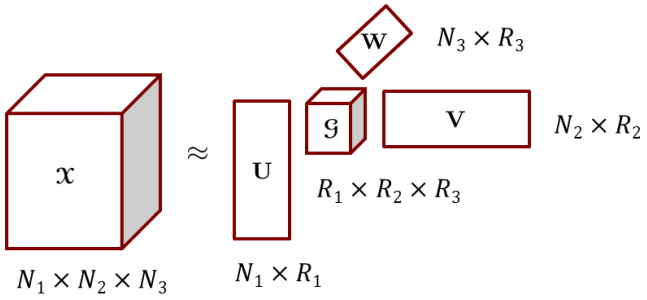


$\mathbf{X}_{(1)}$

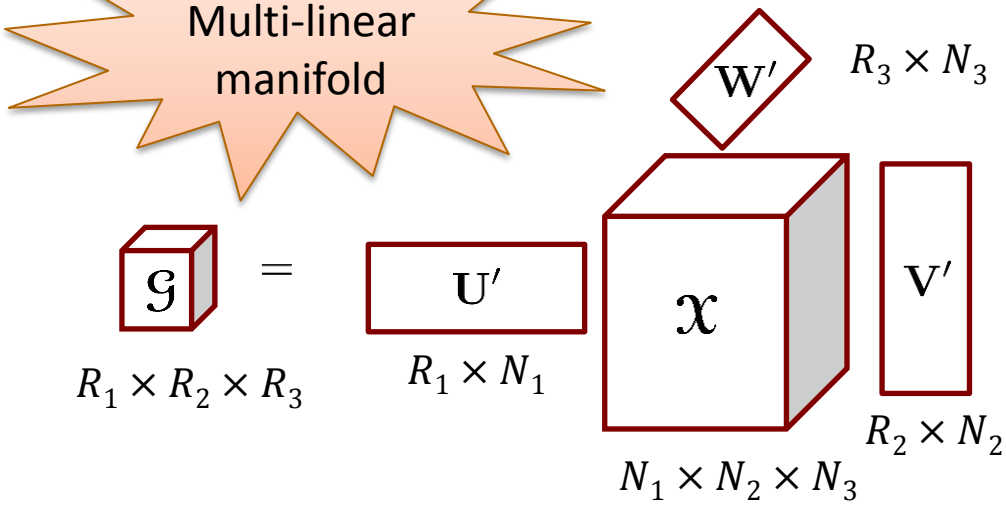
=



Choosing Tucker Ranks to Retain Accuracy



Multi-linear manifold



Find orthogonal matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ that reduce the size of tensor but retain its "mass"

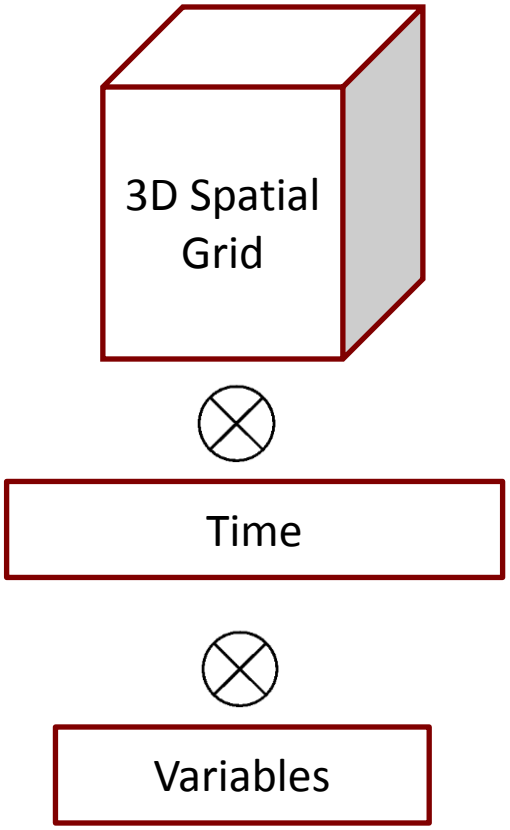
For a given relative error ϵ, choose projection ranks R_1, R_2, and R_3 such that:

$$\|\mathcal{X} - (\mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W})\| \leq \epsilon \|\mathcal{X}\|$$

Core tensor satisfies: $\mathcal{G} = \mathcal{X} \times_1 \mathbf{U}' \times_2 \mathbf{V}' \times_3 \mathbf{W}'$

$$\|\mathcal{X}\|^2 - \|\mathcal{G}\|^2 \leq \epsilon^2 \|\mathcal{X}\|^2$$

Convenient Implementation Assumption: Small Dimensions



$$N_1 \times N_2 \times \cdots \times N_d$$

We assume each $N_k \leq 2000$

Algorithm: ST-HOSVD (3-way)

Vannieuwenhoven, Vandebril, Meerbergen (SISC 2012)

Sandia National Laboratories

1. Choose $\mathbf{U}$ with projection rank R_1 such that: $\|\mathbf{X}_{(1)}\|^2 - \|\mathbf{U}'\mathbf{X}_{(1)}\|^2 \leq \epsilon^2\|\mathcal{X}\|^2/3$
   a) Compute gram matrix: $\mathbf{X}_{(1)}\mathbf{X}_{(1)}'$
   b) Use eigendecomposition of N_1 x N_1 matrix to choose R_1
   c) Set $\mathbf{U}$ = R_1 leading eigenvectors of gram matrix
2. Shrink to size R_1 x N_2 x N_3: $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}'$
3. Choose $\mathbf{V}$ with projection rank R_2 such that: $\|\mathbf{Y}_{(2)}\|^2 - \|\mathbf{V}'\mathbf{Y}_{(2)}\|^2 \leq \epsilon^2\|\mathcal{X}\|^2/3$
   a) Compute gram matrix: $\mathbf{Y}_{(2)}\mathbf{Y}_{(2)}'$
   b) Use eigendecomposition of N_2 x N_2 matrix to choose R_2
   c) Set $\mathbf{V}$ = R_2 leading eigenvectors of gram matrix
4. Shrink to size R_1 x R_2 x N_3: $\mathcal{Z} = \mathcal{Y} \times_2 \mathbf{V}'$
5. Choose $\mathbf{W}$ with projection rank R_3 such that: $\|\mathbf{Z}_{(3)}\|^2 - \|\mathbf{W}'\mathbf{Z}_{(3)}\|^2 \leq \epsilon^2\|\mathcal{X}\|^2/3$
   a) Compute gram matrix: $\mathbf{Z}_{(3)}\mathbf{Z}_{(3)}'$
   b) Use eigendecomposition of N_3 x N_3 matrix to choose R_3
   c) Set $\mathbf{W}$ = R_3 leading eigenvectors of gram matrix
6. Shrink to size R_1 x R_2 x R_3: $\mathcal{G} = \mathcal{Z} \times_3 \mathbf{W}'$

Key Kernels are Matrix-Matrix Multiplies

Sandia National Laboratories

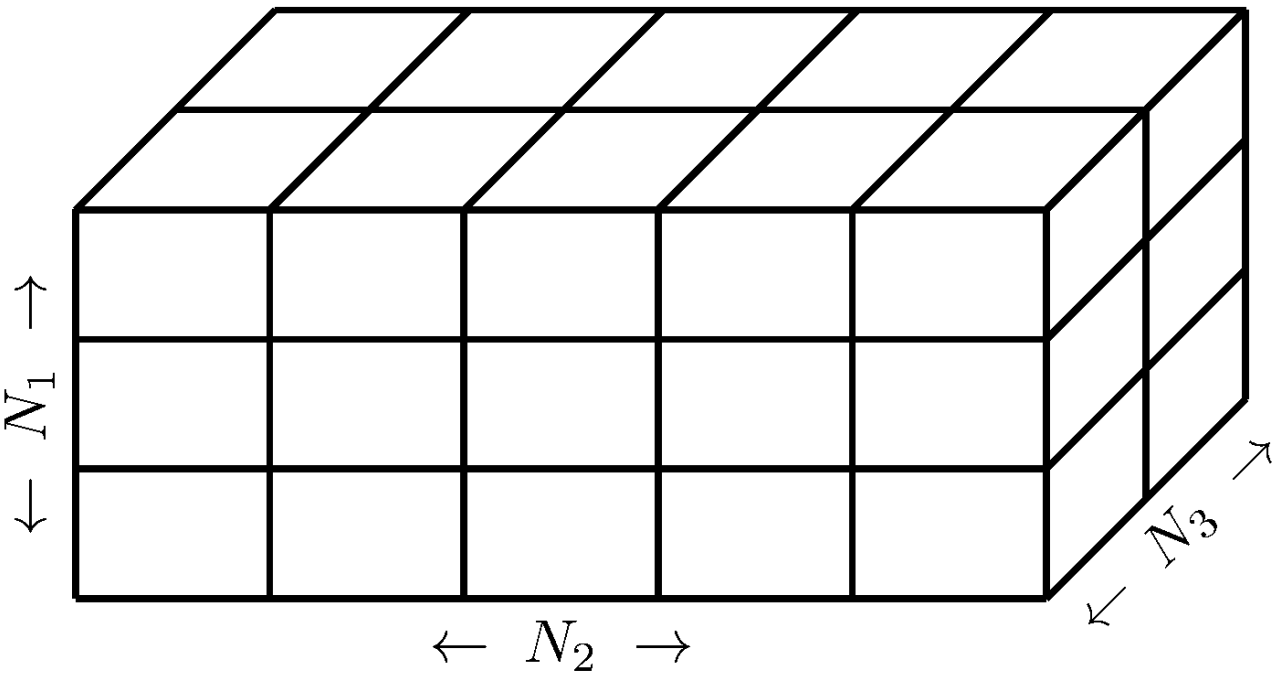
- Three main steps for mode k
  - **GRAM** – Compute matrix product of unfolded tensor with itself.
    - Unfolded matrix size is N_k x ($R_1 \cdots R_{k-1} N_{k+1} \cdots N_d$).
    - Result is N_k x N_k
  - **EVECS** – Compute eigenvalues and eigenvectors of N_k x N_k gram matrix. Use this to determine R_k
  - **TTM** – Tensor-times-matrix to shrink mode k from size N_k to size R_k.
    - Input is size R_1 x $\cdots$ x N_k x N_{k+1} x $\cdots$ x N_d
    - Result is size R_1 x $\cdots$ x R_k x N_{k+1} x $\cdots$ x N_d

New Contribution: Parallel Tucker Decomposition

Sandia National Laboratories

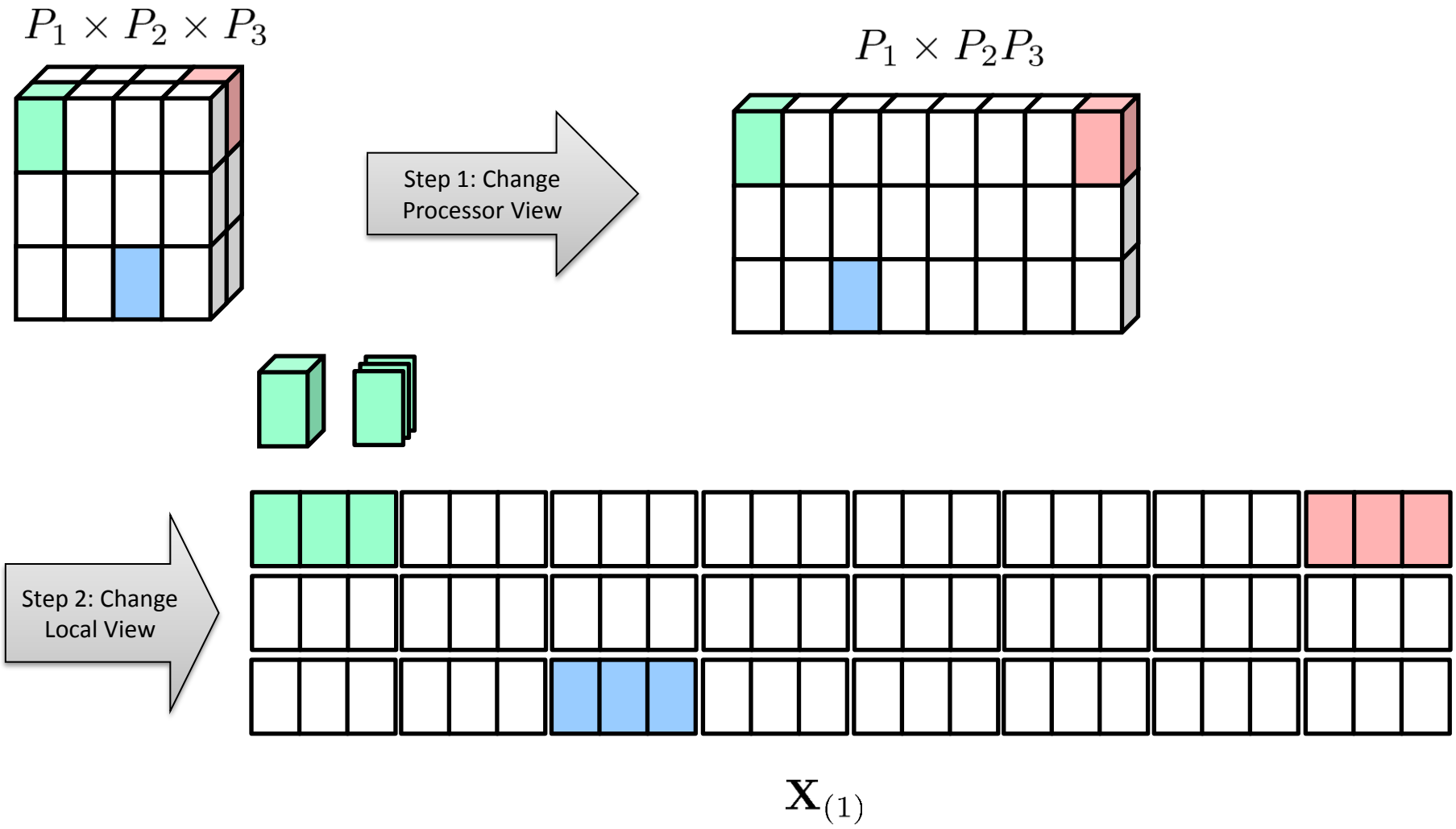
For N-way tensor, Cartesian block distribution on N-way processor grid

$$P_1 \times P_2 \times P_3 = 3 \times 5 \times 2$$



Local block size: $\frac{N_1}{P_1} \times \frac{N_2}{P_2} \times \frac{N_3}{P_3}$

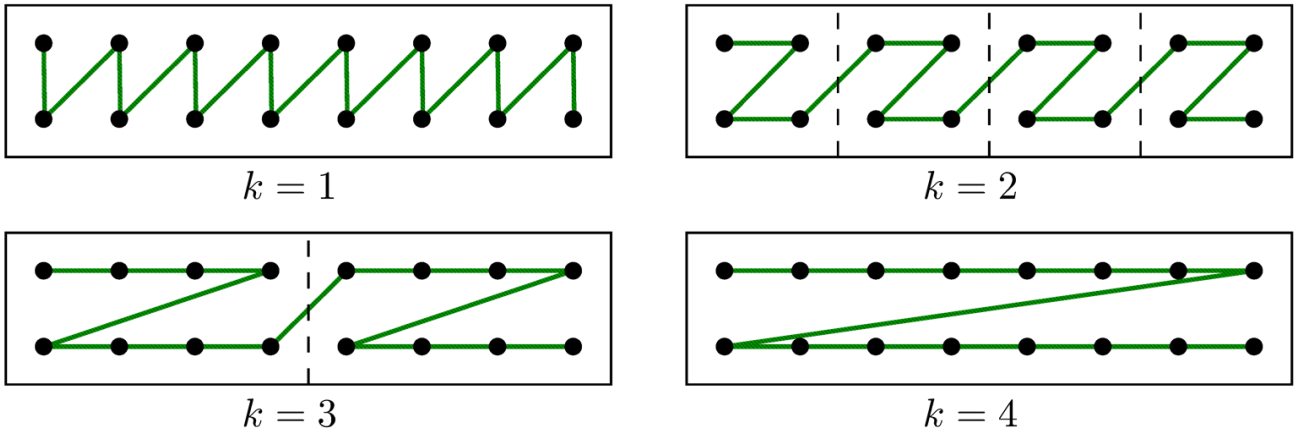
No Data Movement in Parallel Unfolding – Conceptual Transforms



No Data Movement in Local Unfolding – Block Structure

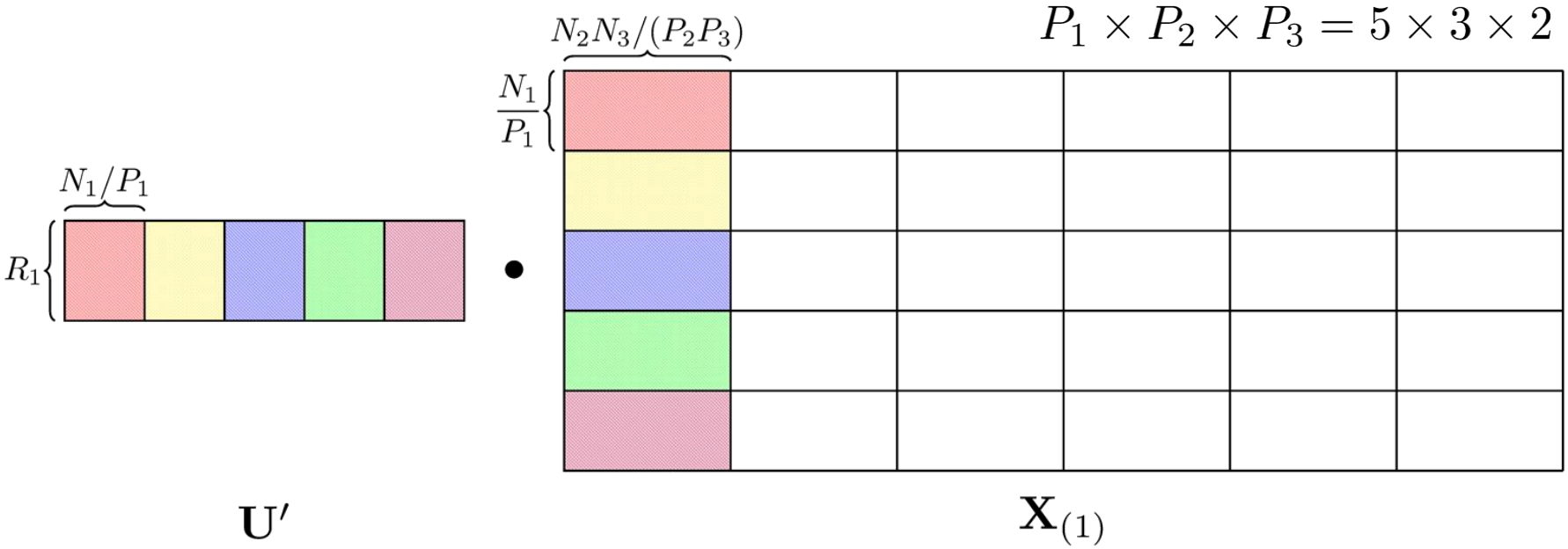
- Assume local data is stored so that mode-1 unfolding is in column major order
- All unfoldings are blocked, with different block sizes
- Rather than rearrange data (standard practice), exploit structure with block operations

Local Layout: 2 x 2 x 2 x 2 (4-way Tensor)



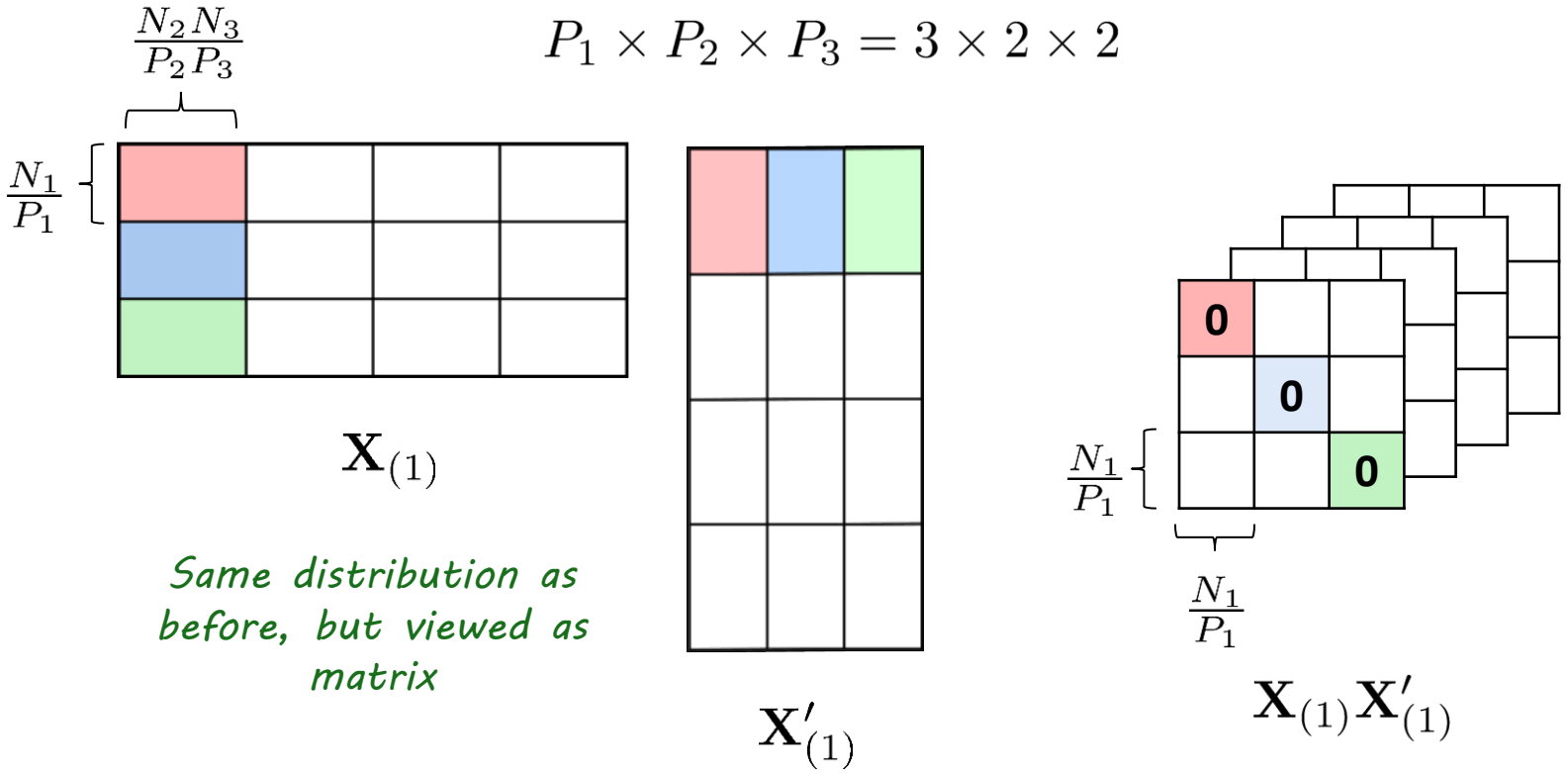
Also independently discovered by Li, Battaglino, Perros, Sun, Vuduc, SC'15

Tensor-Times-Matrix Kernel: Parallel Computation



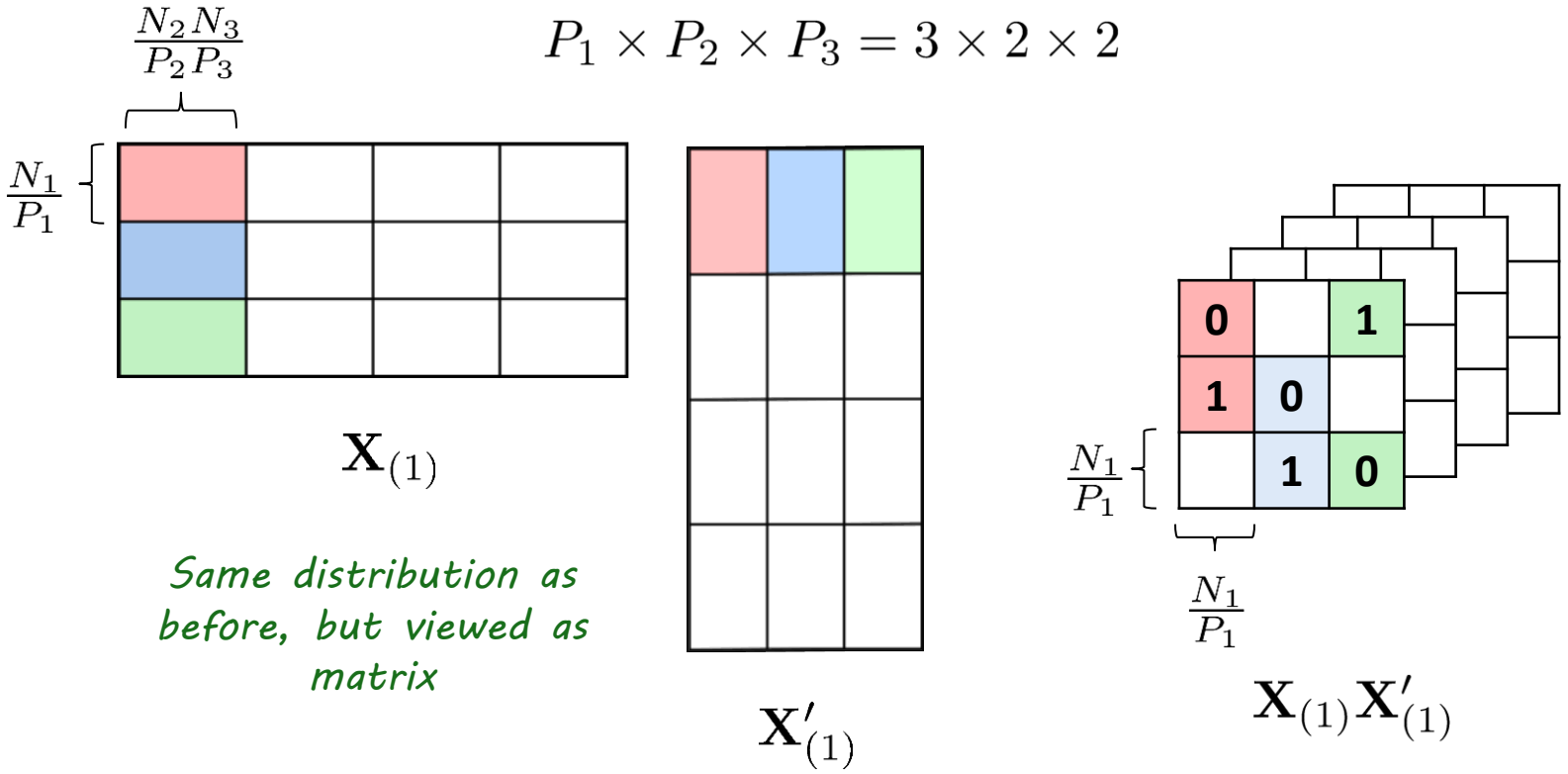
- Each node locally computes product of its part of $\mathbf{U}$ with its part of $\mathbf{X}_{(1)}$
- Communication is a reduce-scatter on the result within set of P_1 nodes
- Output is distributed same as $\mathbf{X}_{(1)}$ but with a smaller first dimension, i.e., size R_1/P_1
- Works the same for every mode (with different views)
- P_1 = 1 means no communication

Gram Matrix Kernel: Parallel Computation (Version 1)



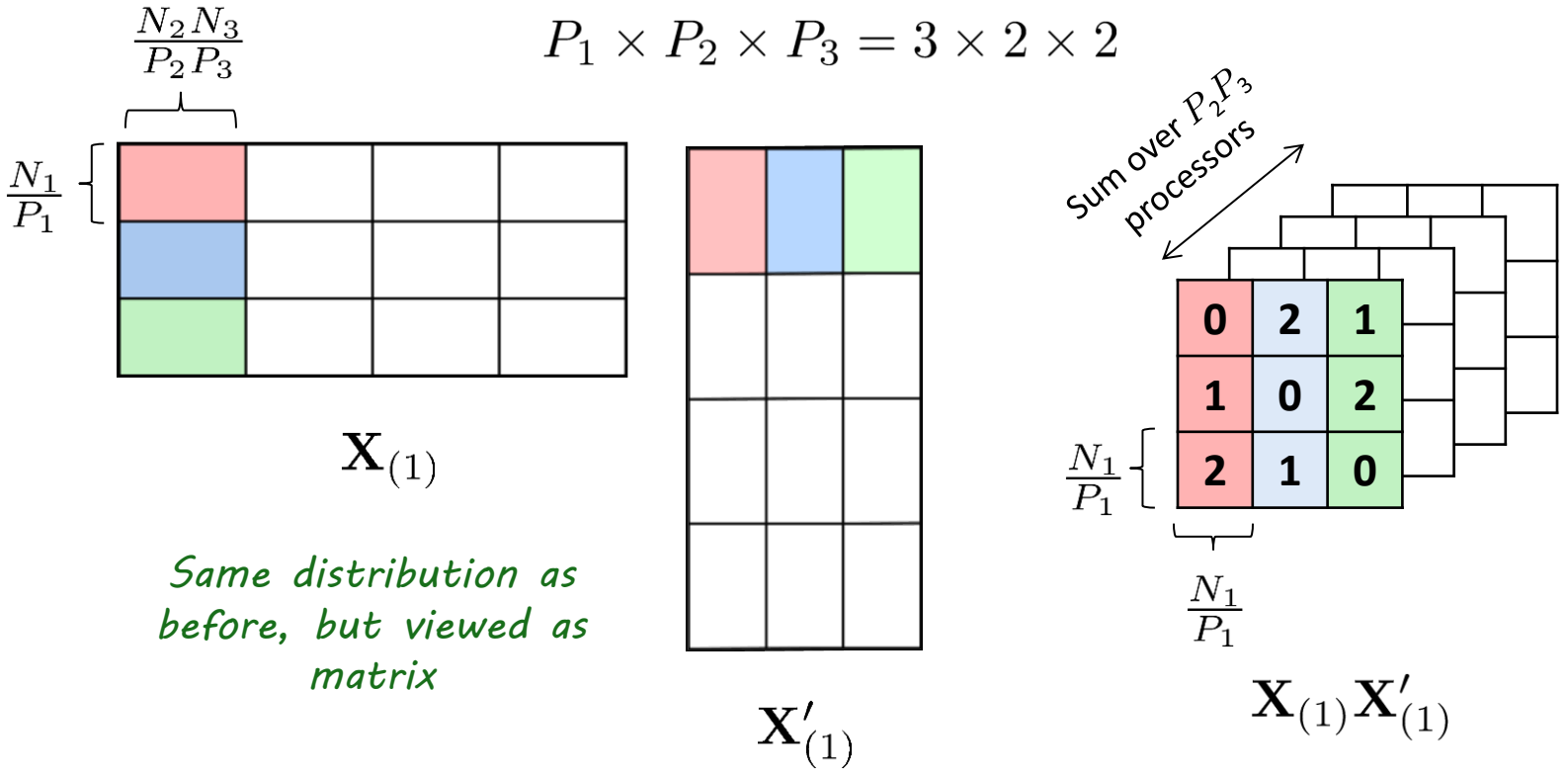
- Each node locally computes product of its part of $\mathbf{X}_{(1)}$ with itself

Gram Matrix Kernel: Parallel Computation (Version 1)



- Each node locally computes product of its part of $\mathbf{X}_{(1)}$ with itself
- Then round-robin sharing to each other processor in group of P_1 nodes

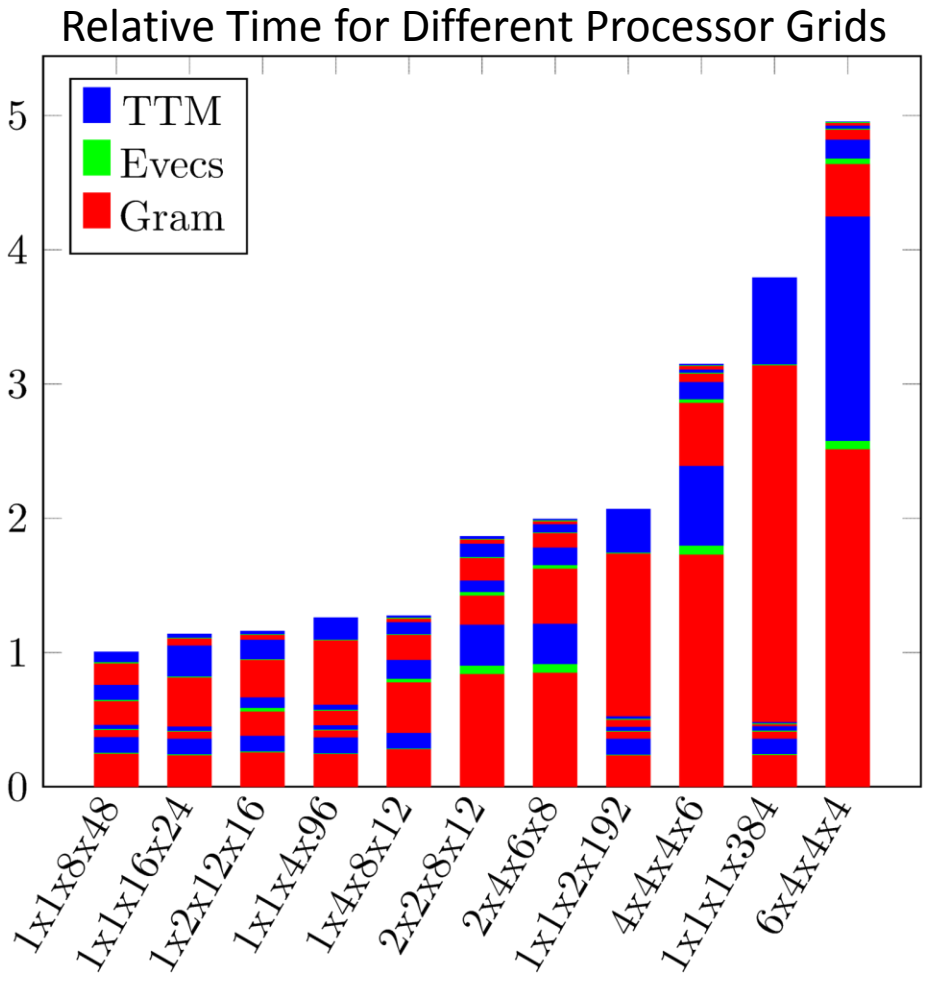
Gram Matrix Kernel: Parallel Computation (Version 1)



- Each node locally computes product of its part of $\mathbf{X}_{(1)}$ with itself
- Then round-robin sharing to each other processor in group of P_1 nodes
- Sum across all $P_2 P_3$ groups with all-reduce
- P_1 =1 means no communication to do multiplies, just all-reduce

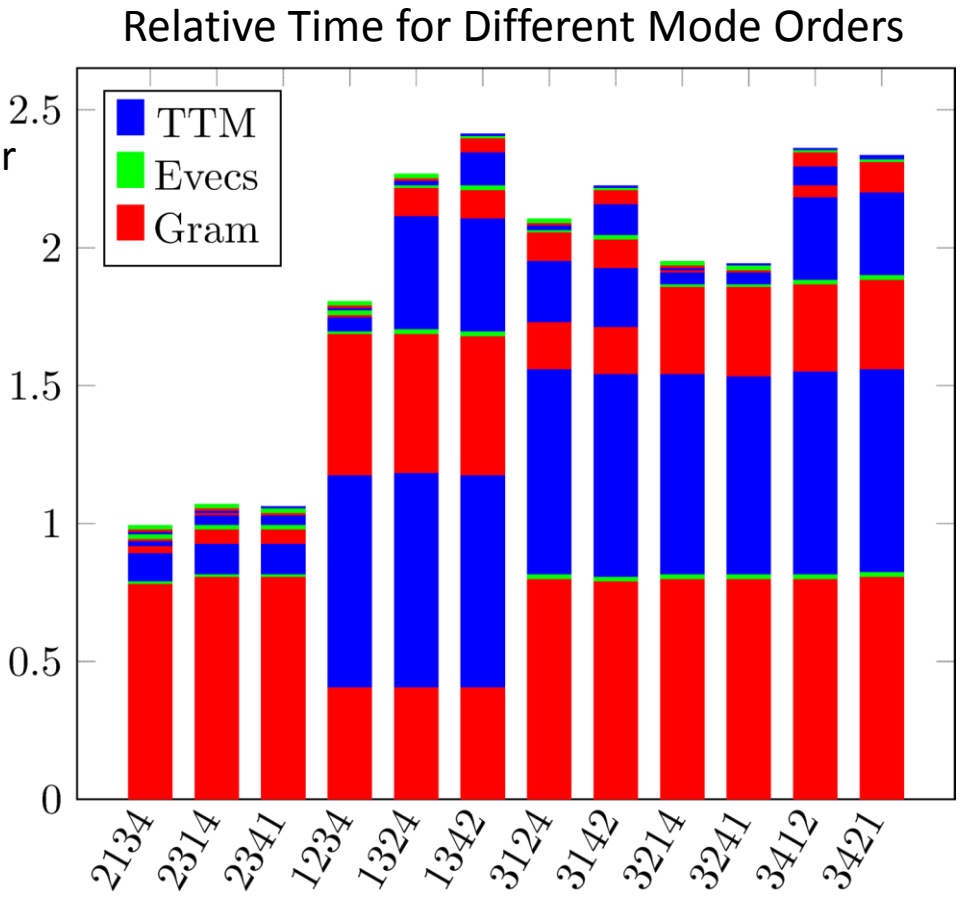
Processor Grid: Fewer Processors in Early Modes

- 4-way Tensor, Reduce by Factor of 4^4
  - Tensor = 384 x 384 x 384 x 384
  - Reduced Tensor = 96 x 96 x 96 x 96
  - # Processes= 384 (multiple of 12)
- Showing relative time for different choices of P_1 x P_2 x P_3 x P_4
  - First operation is at bottom
  - Cycle: Gram, Evecs, TTM
- Compute Platform
  - Edison (NERSC), Cray XC30
  - 12-core Intel Ivy Bridge
  - Peak 19.2 GFLOPS/core
  - 64GB/node
  - LibSci BLAS/LAPACK
- *Result: Want fewer processors up front because minimizes communication in Gram*



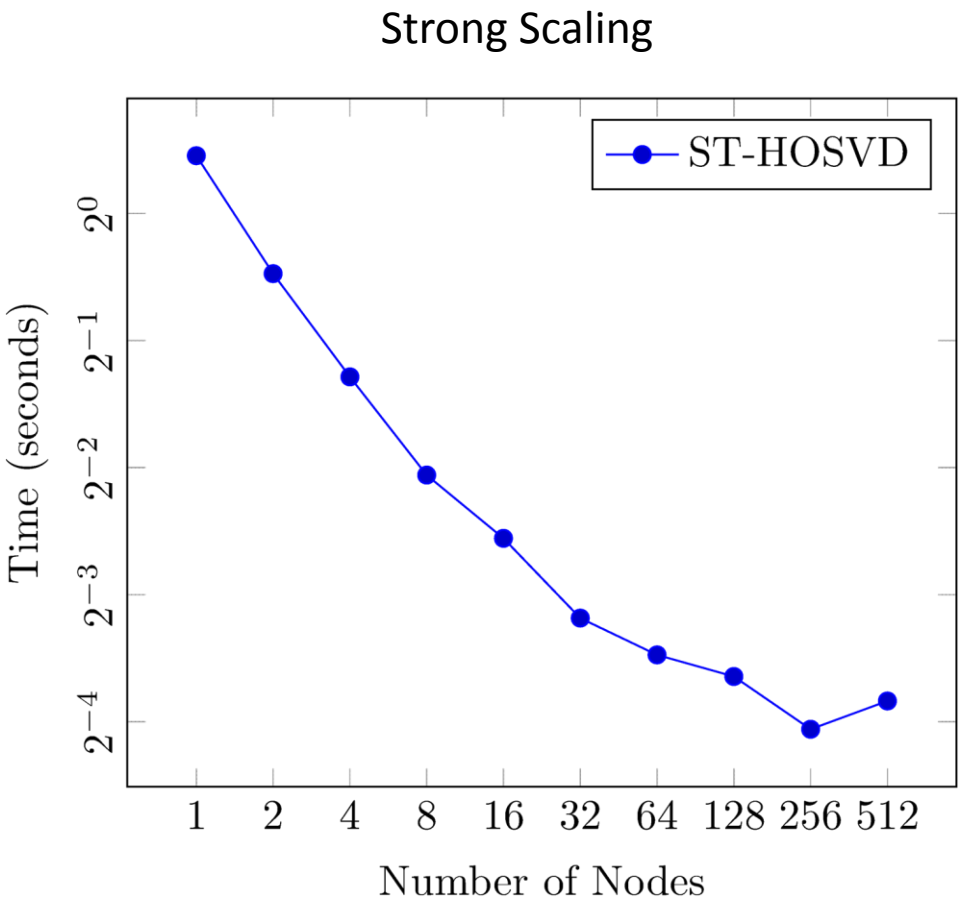
Mode Order: Target Largest Mode with Biggest Reduction

- 4-way tensor
  - 25 x 250 x 250 x 250 Tensor
  - 10 x 10 x 100 x 100 Reduced Tensor
  - 2.5 x 25 x 2.5 x 2.5 Reduction
  - 16 processes in 2 x 2 x 2 x 2 grid
- Showing relative time for different mode orders
  - First operation is at bottom
  - Cycle: Gram, Evecs, TTM
- Compute Platform
  - Edison (NERSC), Cray XC30
  - 12-core Intel Ivy Bridge
  - Peak 19.2 GFLOPS/core
  - 64GB/node
- *Result: Process largest mode with biggest compression ratio first*



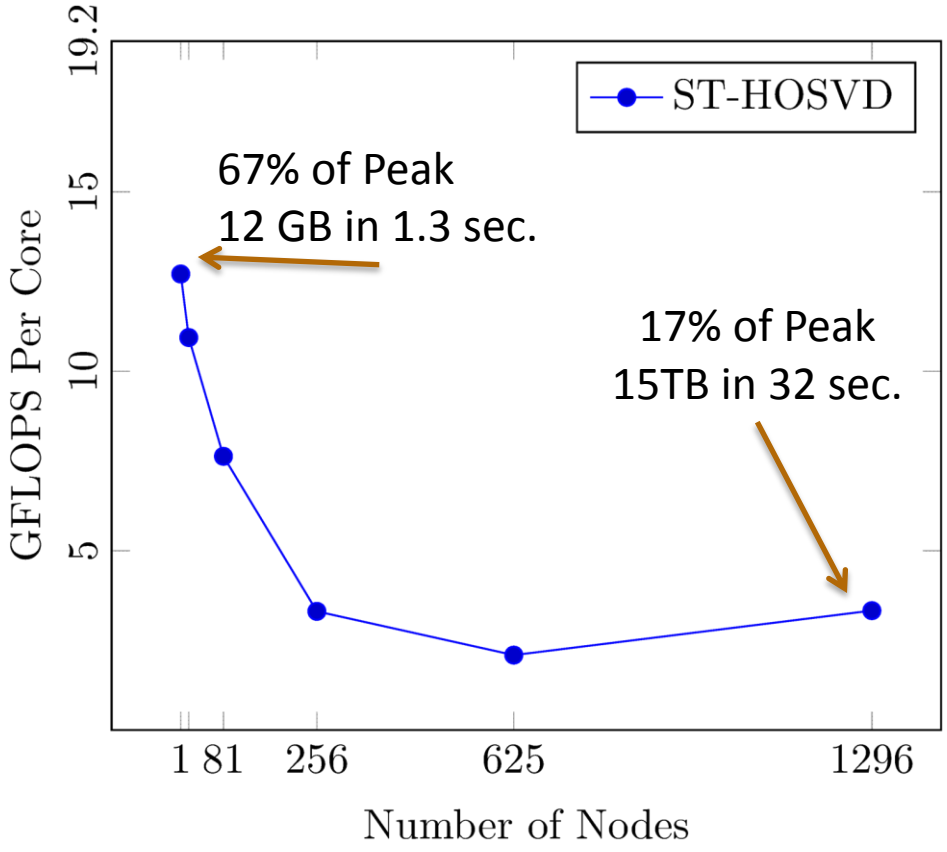
Strong Scaling

- Problem Setup
  - 200 x 200 x 200 x 200 Tensor (12 GB)
  - 20 x 20 x 20 x 20 Reduced Tensor
  - 24 x 2^k processes for k=0,…,9
  - Turned processor grid
- Results
  - 1 node: 12GB memory, 1.3 seconds, 310 GFLOPS (67% of peak)
  - 512 nodes: 1MB memory, 0.07 seconds, 6 TFLOPS
- Compute Platform
  - Edison (NERSC), Cray XC30
  - 12-core Intel Ivy Bridge
  - Peak 19.2 GFLOPS/core
  - 64GB/node



Weak Scaling

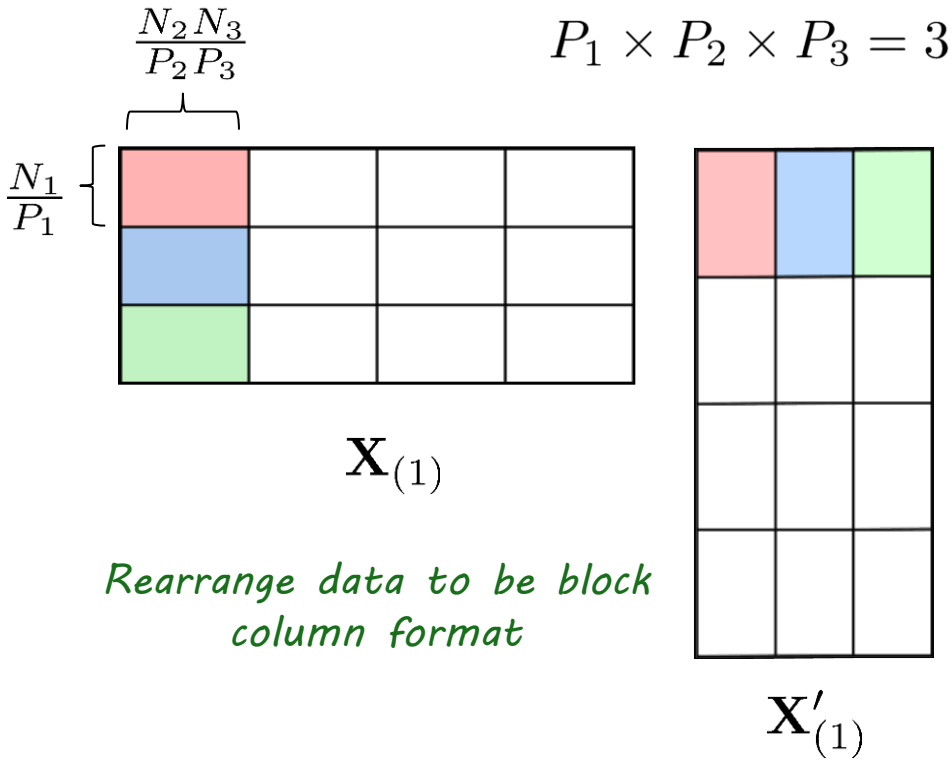
- Problem Setup
  - $200k$ x $200k$ x $200k$ x $200k$ Tensor
  - $20k$ x $20k$ x $20k$ x $20k$ Reduced
  - 24 x k^4 processors for k=0,…,6
  - Tuned processor grid
- Results
  - 1 node (k=1): 67% of peak
    - 12GB in 1.3 sec
  - 1296 nodes (k=6): 17% of peak
    - 15TB in 32 sec
- Compute Platform
  - Edison (NERSC), Cray XC30
  - 12-core Intel Ivy Bridge
  - Peak 19.2 GFLOPS/core
  - 64GB/node



Gram Matrix Kernel: Parallel Computation (Version 2)

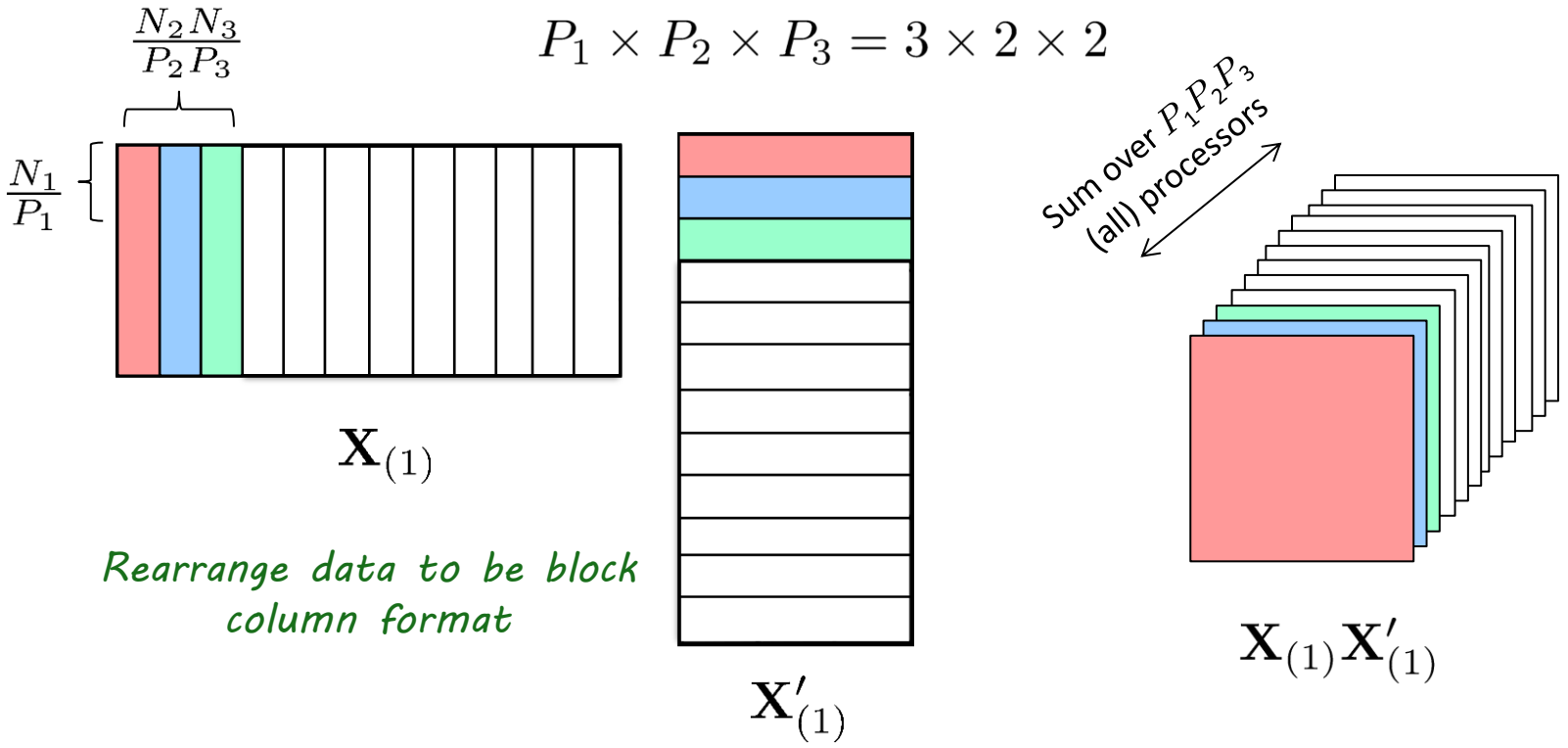
Sandia National Laboratories

$$P_1 \times P_2 \times P_3 = 3 \times 2 \times 2$$



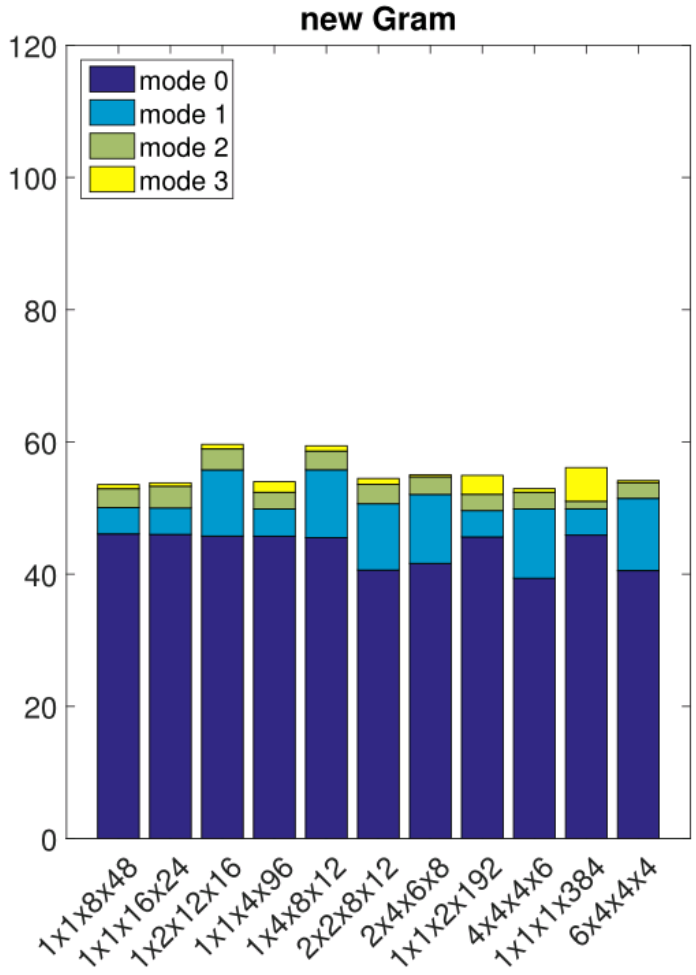
Rearrange data to be block column format

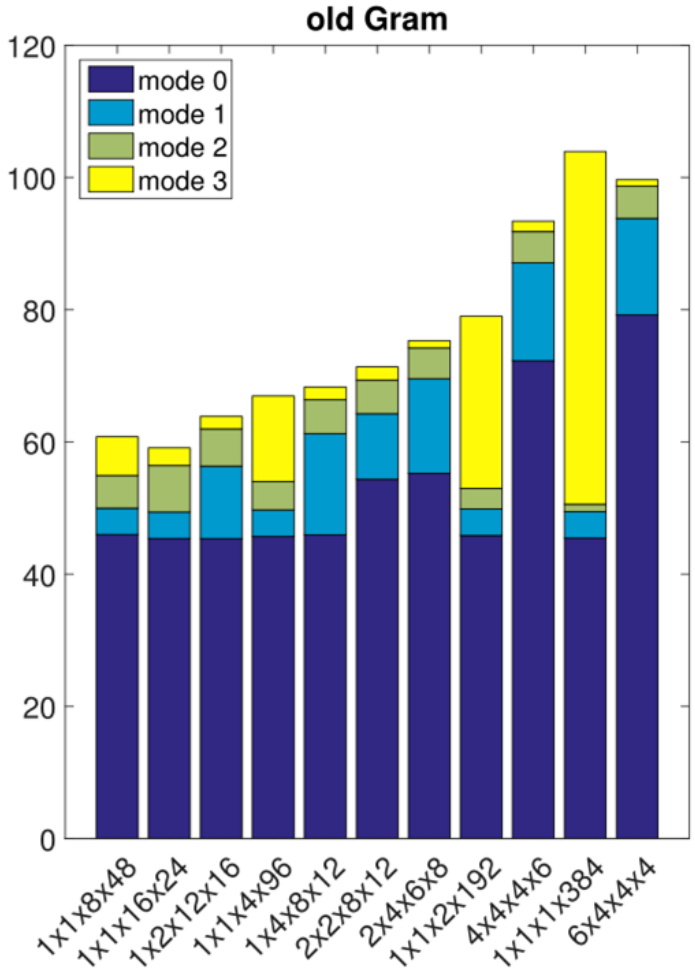
Gram Matrix Kernel: Parallel Computation (Version 2)



- Each processor column rearranges its data (with P_1 nodes)
- Then computes local outer product
- Sum across all $P_1 P_2 P_3$ groups with all-reduce
- P_1 =1 means no communication in rearrangement, just all-reduce

New Gram Improves Worst-Case Grid Choice Performance

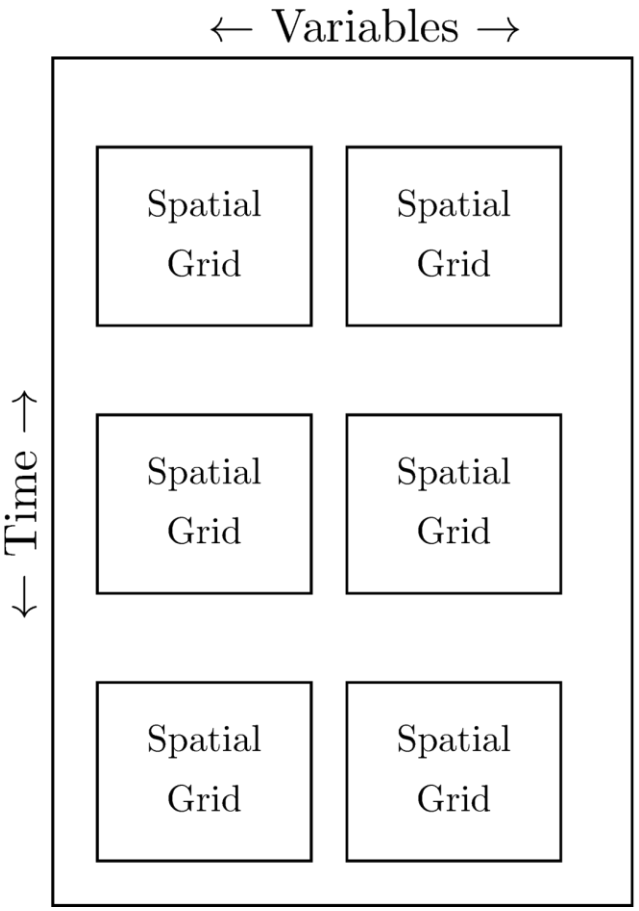




4-way Tensor of size 384^4 reduced to size 96^4.

Comparisons on Kahuna @ Sandia: 120 Nodes x 2 Intel Haswell 14-core x 256 GB

Combustion Simulation Results



4-way Tensor, 672 x 672 x 32 x 626
Storage: 74GB

Reduced size determined automatically to satisfy error criteria:

$$\|\mathcal{X} - \hat{\mathcal{X}}\| \leq \epsilon \|\mathcal{X}\|$$

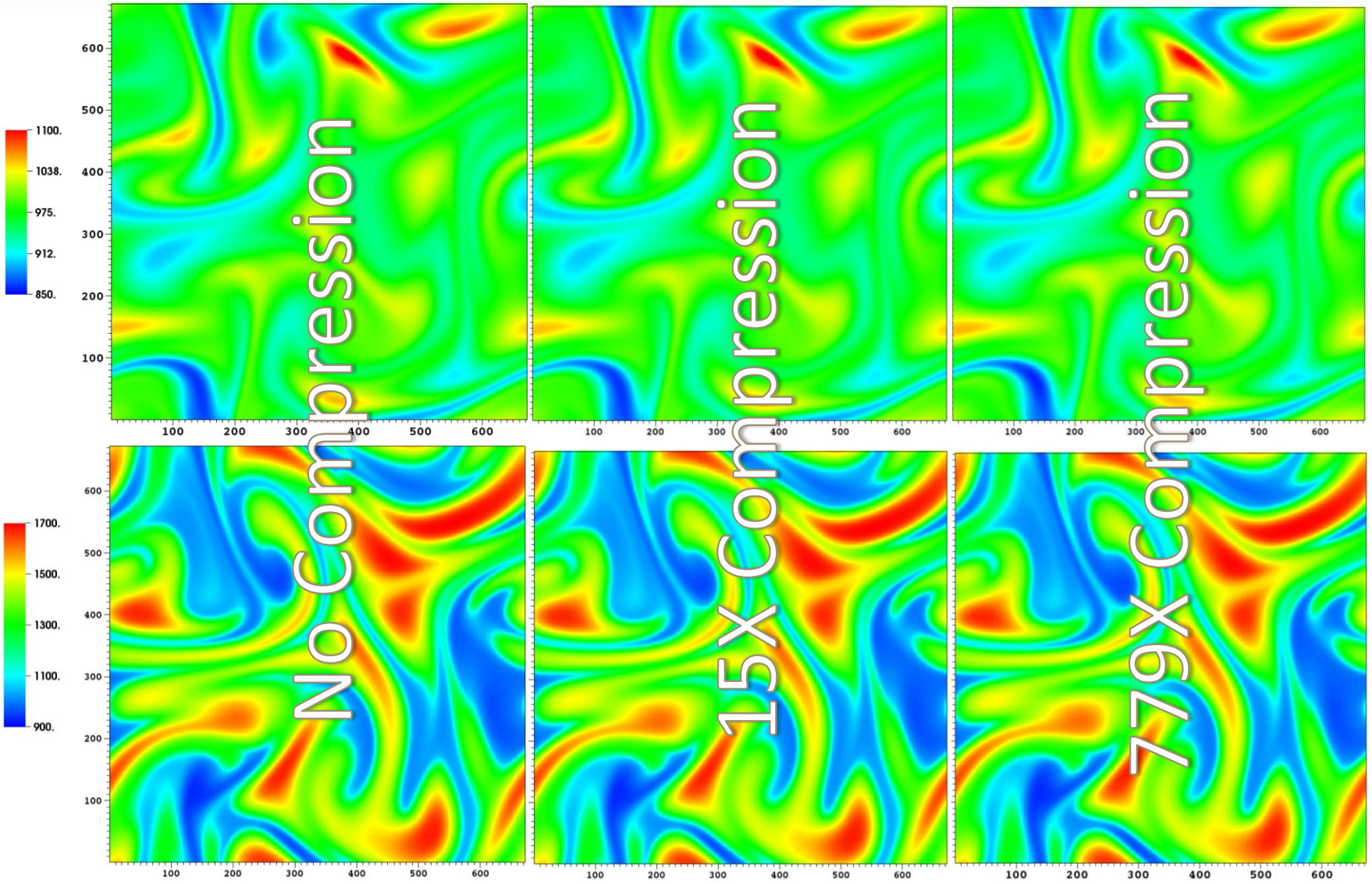
Compression ratio:

$$C = \prod_{k=1}^{d} N_k \Big/ \left(\prod_{k=1}^{d} R_k + \sum_{k=1}^{d} N_k R_k \right)$$

- Combustion simulation: HCCI Ethanol
  - 2D Spatial Grid: 672 x 672
  - Species/Variables: 32
  - Time steps: 626
- *Scaled each species by its max value*
- Experiment 1: $\epsilon = 10^{-2}$
  - New Core: 111 x 105 x 22 x 46 (<100MB)
  - Compression: 779X
- Experiment 2: $\epsilon = 10^{-4}$
  - New Core: 330 x 310 x 31 x 199 (640MB)
  - Compression: 15X

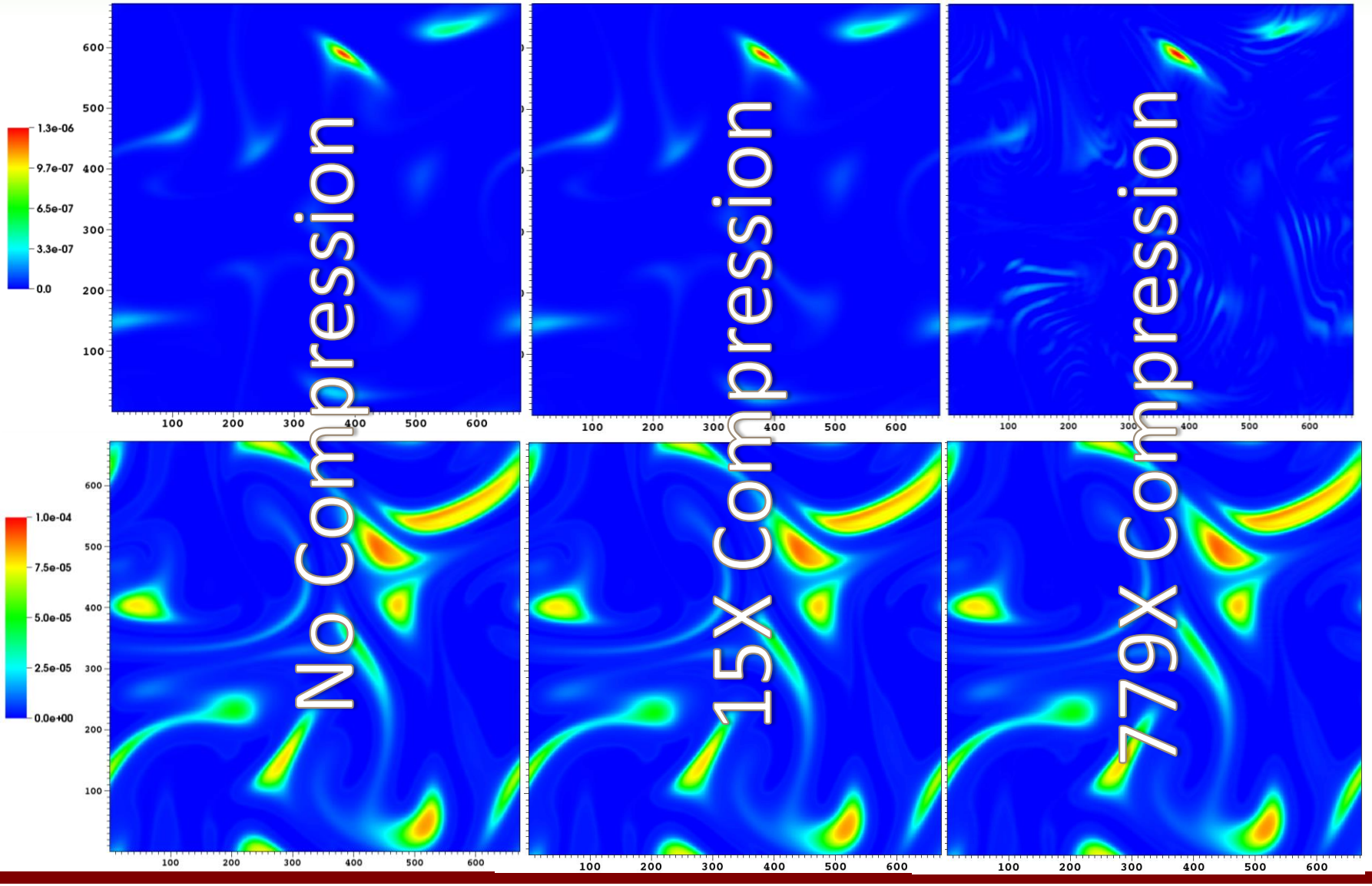
Combustion Simulation Results: Temperature @ Disparate Timesteps

Sandia National Laboratories

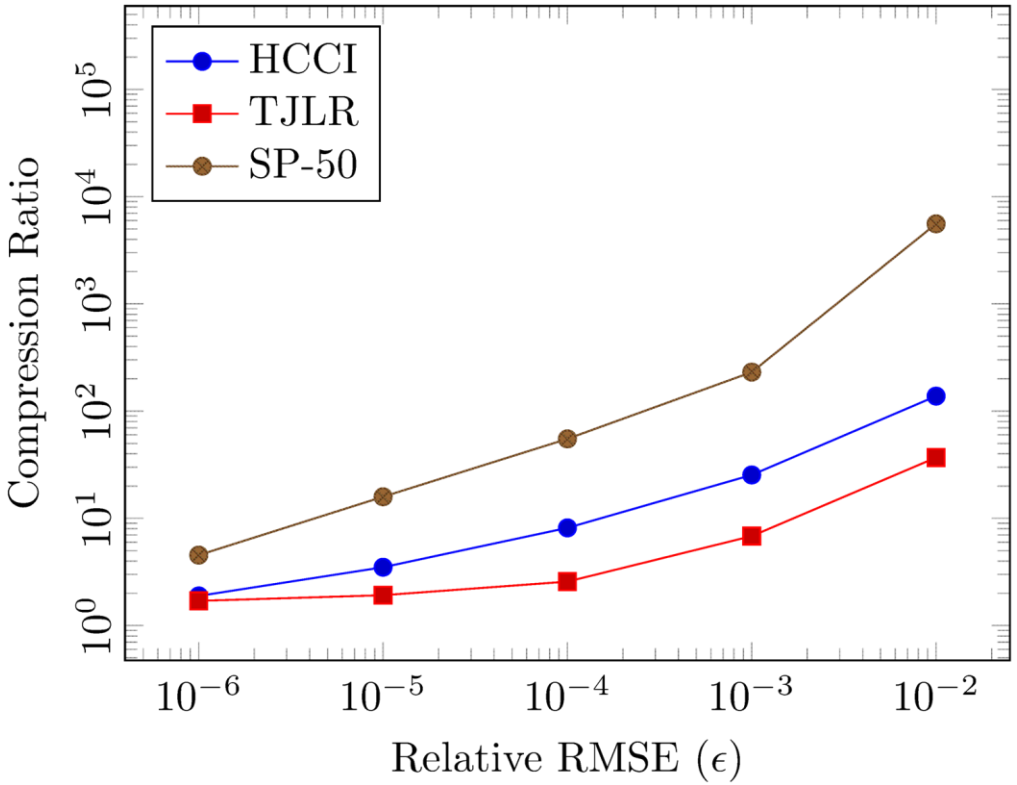


Combustion Simulation Results: OH @ Disparate Timesteps

Sandia National Laboratories



Compression on Different Data Sets

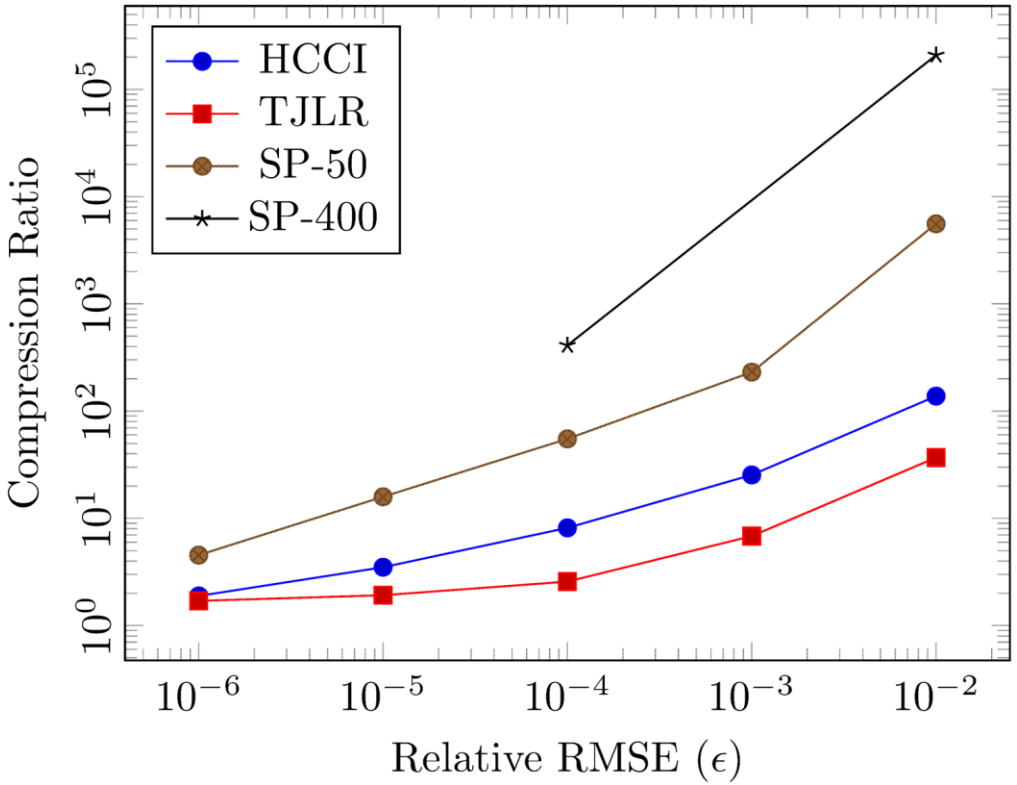


$$\|\mathcal{X} - \hat{\mathcal{X}}\| \leq \epsilon \|\mathcal{X}\|$$

$$C = \prod_{k=1}^{d} N_k \Big/ \left(\prod_{k=1}^{d} R_k + \sum_{k=1}^{d} N_k R_k \right)$$

- HCCI: 672 x 672 x 33 x 627
- TJLR: 460 x 700 x 360 x 35 x 16
- SP-50: 500 x 500 x 500 x 11 x 50

Compression on Different Data Sets



$$\|\mathcal{X} - \hat{\mathcal{X}}\| \leq \epsilon \|\mathcal{X}\|$$

$$C = \prod_{k=1}^{d} N_k \Big/ \left(\prod_{k=1}^{d} R_k + \sum_{k=1}^{d} N_k R_k \right)$$

- HCCI: 672 x 672 x 33 x 627
- TJLR: 460 x 700 x 360 x 35 x 16
- SP-50: 500 x 500 x 500 x 11 x 50
- SP-400: 500 x 500 x 500 x 11 x 400

New results just in on 4.4 TB tensor for SP of size 500 x 500 x 500 x 11 x 400.
Achieved 410X compression at 1e-4 RMSE, reducing to just 10 GB!
Run time is 58 seconds on ORNL's Rhea platform, using 1104 processes.

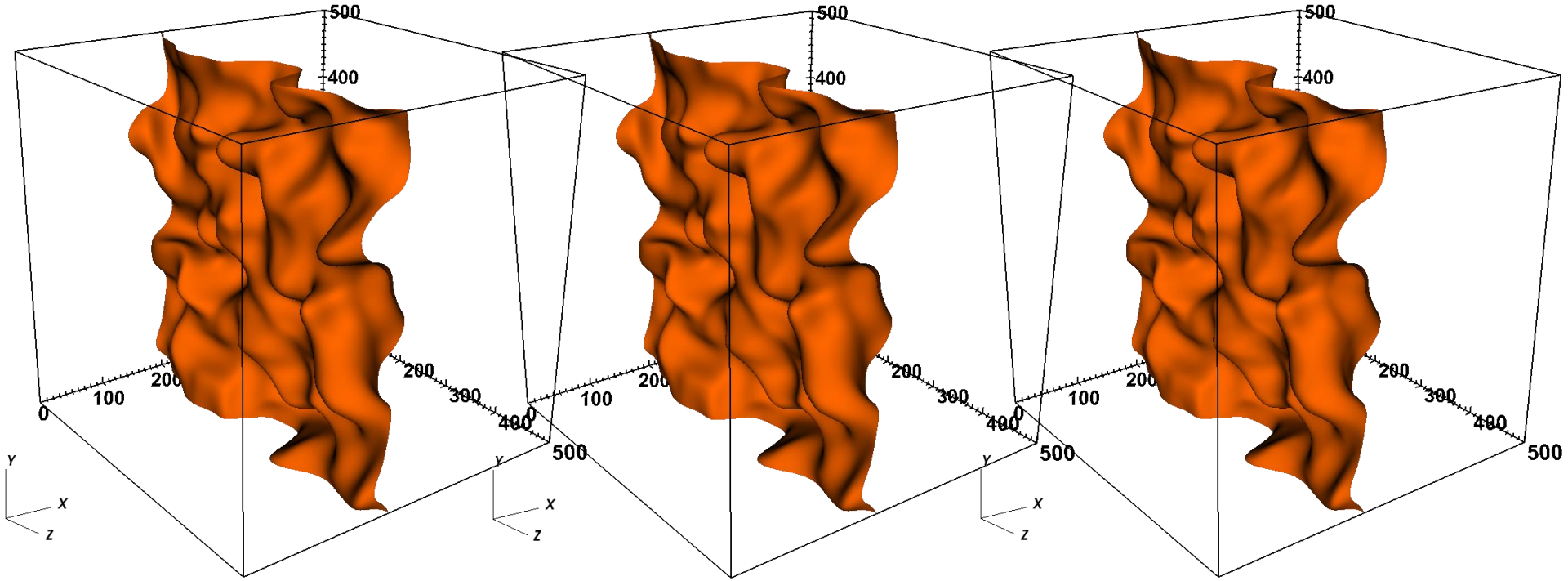
SP-400 4.4TB Dataset: Can you guess which is the original?

Sandia National Laboratories

Original

ϵ = 1e-4

ϵ = 1e-2



400X Compression

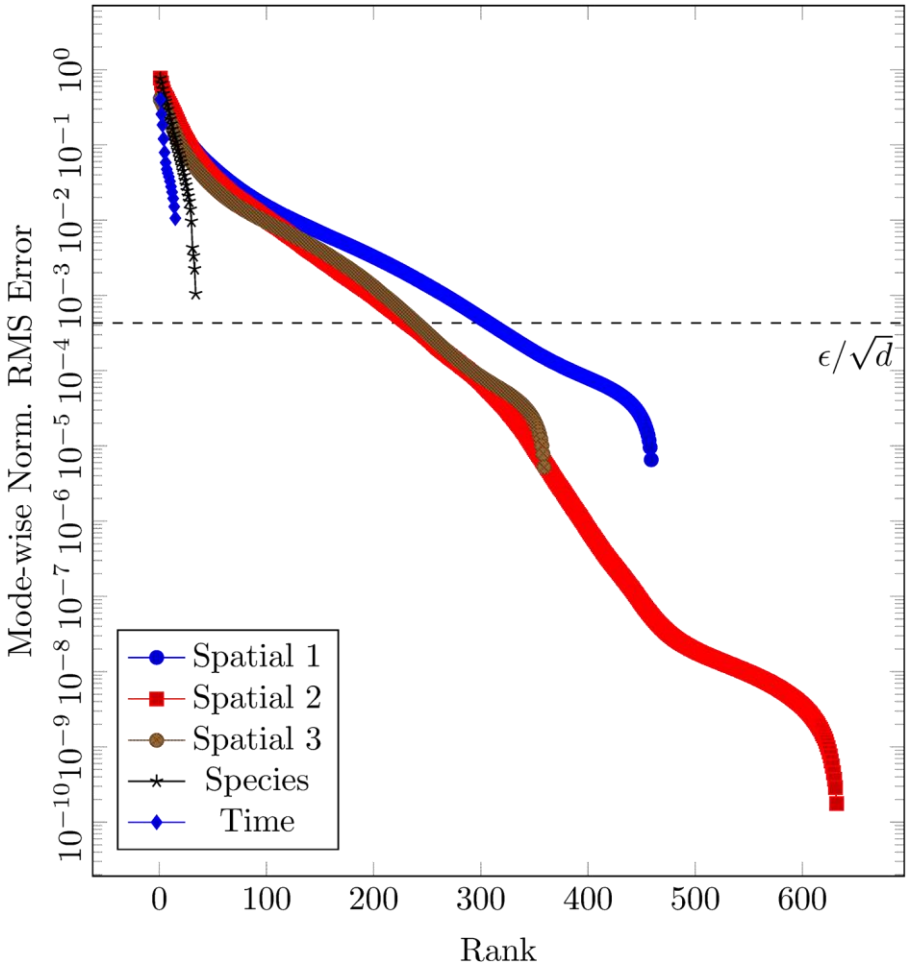
20000X Compression

Flame surface at single time step. Using temperature variable (iso-value is 2/3 of max).

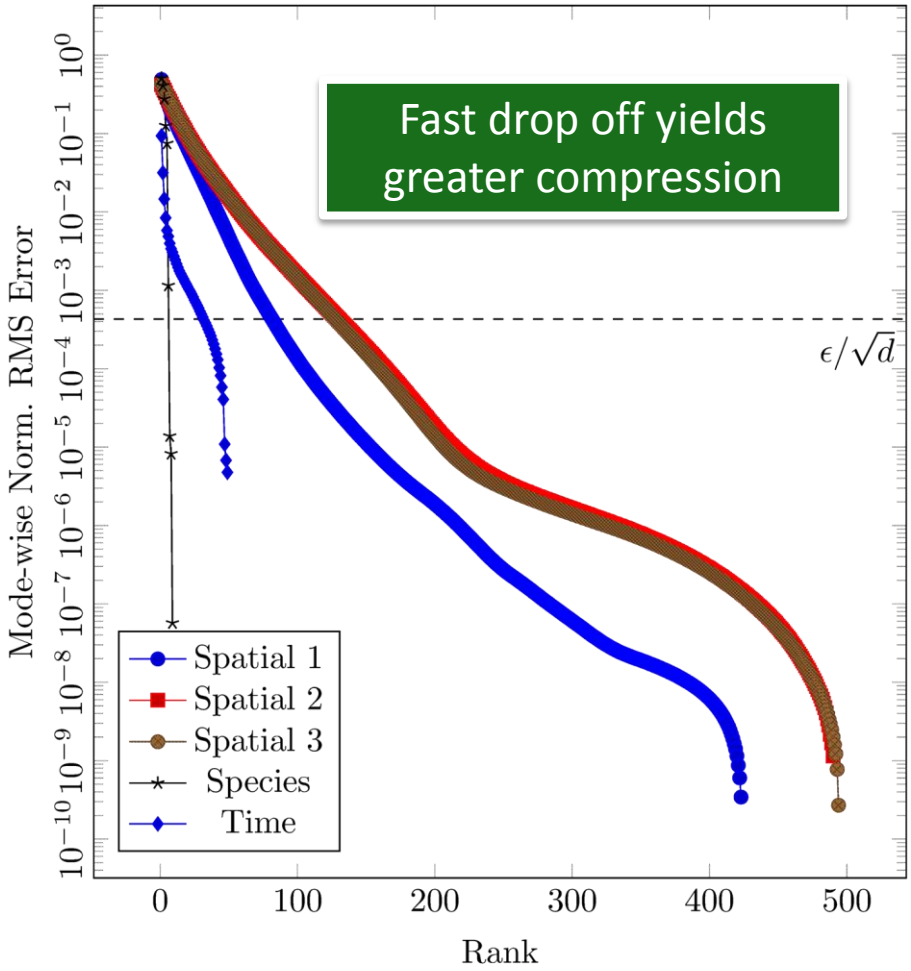
Compression Depends on Data Redundancy (Eigenvalues Don't Lie)

Sandia National Laboratories

TJLR: 460 x 700 x 360 x 35 x 16



SP-50: 500 x 500 x 500 x 11 x 50



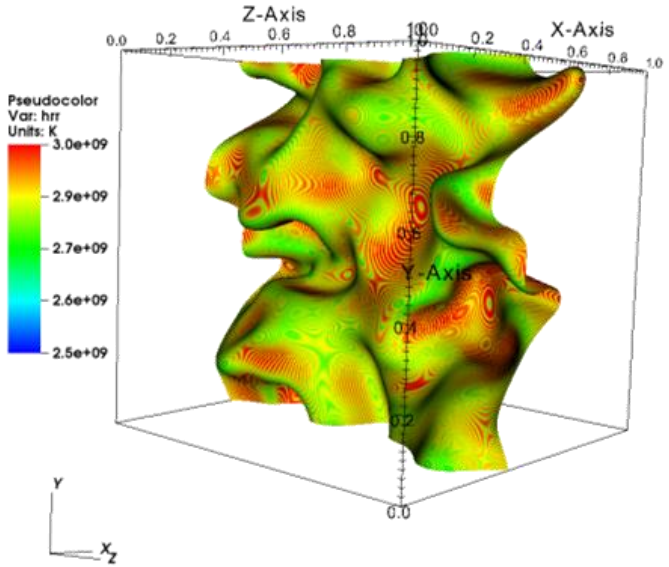
Key Feature: Need Only Do Partial Reconstruction on Laptops, etc.

Sandia National Laboratories

Reconstruction requires as much space as the original data!

$$\hat{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)} \times_5 \mathbf{U}^{(5)}$$

$$N_1 \times N_2 \times N_3 \times N_4 \times N_5$$



But we can just reconstruct the portion that we need at the moment:

$$\bar{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{C}^{(3)}\mathbf{U}^{(3)} \times_4 \mathbf{C}^{(4)}\mathbf{U}^{(4)} \times_5 \mathbf{C}^{(5)}\mathbf{U}^{(5)}$$

$$N_1 \times N_2 \times \frac{N_3}{2} \times 1 \times 1$$

$$\mathbf{C}^{(3)} = \begin{bmatrix} 1/2 & 0 & \cdots & 0 \\ 1/2 & 0 & \cdots & 0 \\ 0 & 1/2 & \cdots & 0 \\ 0 & 1/2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

Downsample

$$\mathbf{C}^{(4)} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

Pick single variable

$$\mathbf{C}^{(5)} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

Pick single time step

Software: TuckerMPI

`git@gitlab.com:tensors/TuckerMPI.git`

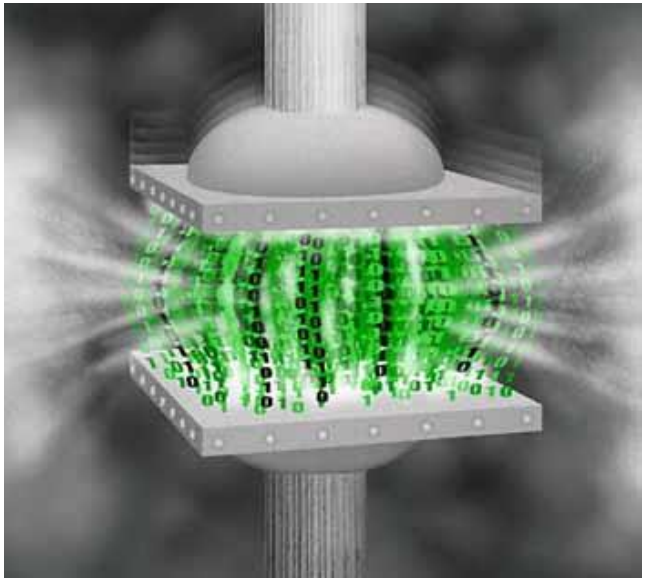
Alicia Klinvex, Woody Austin, Grey Ballard, Hemanth Kolla, Tammy Kolda

- Open source code for computing Tucker compression
- MPI/BLAS/LAPACK/C++11
- Still in development but available for testing
- Looking for new applications and users
- Interesting in partnering

Tensor Tucker Decomposition for Compression for Scientific Data

Sandia National Laboratories

- First parallel implementation of Tucker decomposition
  - 5-way data using regular grid
  - Process 4TB data in < 1 min
- Up to 20000X compression on real-world data
  - Specify desired relative RMSE
  - Discovers latent multi-linear structure
  - Enables “smart” compression rather than discarding data that may be useful
- More work to do…
  - In situ computations
  - Real-time visualization for computational steering, etc.
  - Adaptive and non-uniform grids
  - Experimental data
  - Randomization
  - Extensive testing



http://www.analyticbridge.com/profiles/blogs/how-much-is-big-data-compressible-an-interesting-theorem

Tammy Kolda
tgkolda@sandia.gov

W. Austin, G. Ballard, and T. G. Kolda, ***Parallel Tensor Compression for Large-Scale Scientific Data***, IPDPS’16 (arXiv:1510.06689)