Parallel Tensor Compression for Large-Scale Scientific Data

Woody Austin, **Grey Ballard**, Tamara G. Kolda

Sandia National Laboratories

April 14, 2016

SIAM Conference on Parallel Processing for Scientific Computing
MS 44/52: Parallel Algorithms for Tensor Computations

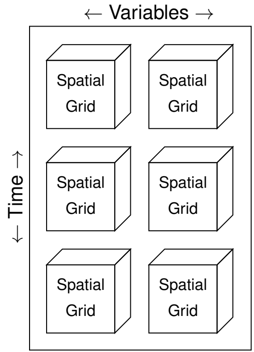
U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

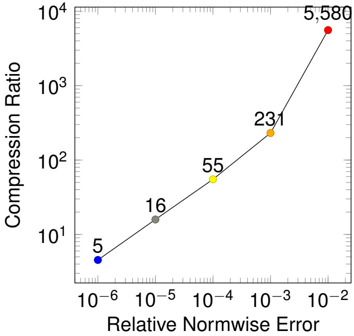
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

(SAND2015-9290 C)

Summary



Natural five-way multiway structure of scientific data

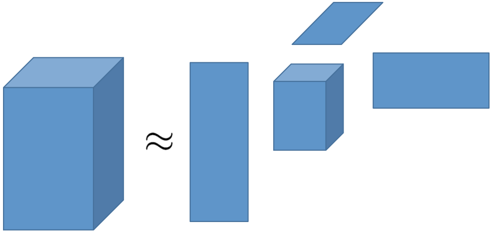


Compression rates as fidelity varies for 550 GB simulation dataset

- We use a Tucker model to approximate dense tensor data
- We introduce a parallel algorithm to compress large data sets

Tucker Tensor Model

Tucker model generalizes the matrix SVD



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

$\mathcal{G}$ is *core tensor*

$\mathbf{U}, \mathbf{V}, \mathbf{W}$ are *factor matrices*

Factor matrices usually tall and skinny with orthonormal columns

Notation conventions: scalar N, vector $\mathbf{v}$, matrix $\mathbf{M}$, tensor $\mathcal{T}$ [KB09]

Tensor-Times-Matrix (TTM)

Tensor version:

$$\mathcal{Y} = \mathcal{X} \times_2 \mathbf{M}$$

$$\mathcal{Y} \in \mathbb{R}^{I \times Q \times K} \qquad \mathcal{X} \in \mathbb{R}^{I \times J \times K} \qquad \mathbf{M} \in \mathbb{R}^{Q \times J}$$

Matrix version:

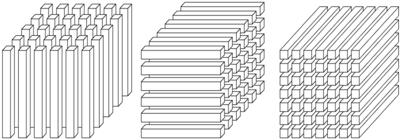
$$\mathbf{Y}_{(2)} = \mathbf{M}\mathbf{X}_{(2)}$$

$$\mathbf{Y}_{(2)} \in \mathbb{R}^{Q \times IK} \qquad \mathbf{X}_{(2)} \in \mathbb{R}^{J \times IK}$$

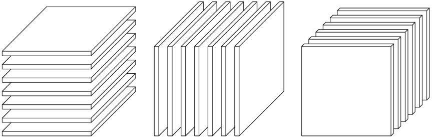
TTM is matrix multiplication with specified unfolding

Notation: $\mathbf{T}_{(n)}$ is unfolding of $\mathcal{T}$ in mode n

Fibers, Slices, and Unfoldings

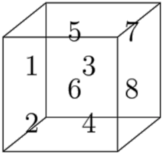


Fibers



Slices

$\mathcal{X} =$



$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

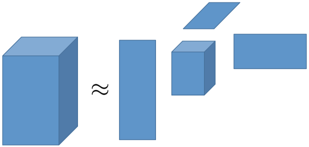
Unfoldings

A tensor can be reshaped into matrices called unfoldings

- columns correspond to fibers
- rows correspond to slices

Notation: $\mathbf{T}_{(n)}$ is unfolding of $\mathcal{T}$ in mode n

Tucker Optimization Problem



For fixed ranks, we want to solve

$$\min_{\mathcal{G},\mathbf{U},\mathbf{V},\mathbf{W}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|,$$

which turns out to be equivalent to

$$\max_{\mathbf{U},\mathbf{V},\mathbf{W}} \|\mathcal{G}\| \text{ subject to } \mathcal{G} = \mathcal{X} \times_1 \mathbf{U}^T \times_2 \mathbf{V}^T \times_3 \mathbf{W}^T,$$

a nonlinear, nonconvex optimization problem

Higher-Order Orthogonal Iteration (HOOI)

Fixing all but one factor matrix, we have a matrix problem:

$$\max_{\mathbf{V}} \left\| \mathcal{X} \times_1 \hat{\mathbf{U}}^\mathsf{T} \times_2 \mathbf{V}^\mathsf{T} \times_3 \hat{\mathbf{W}}^\mathsf{T} \right\|$$

or equivalently

$$\max_{\mathbf{V}} \left\| \mathbf{V}^\mathsf{T} \mathbf{Y}_{(2)} \right\|_F$$

where $\mathcal{Y} = \mathcal{X} \times_1 \hat{\mathbf{U}}^\mathsf{T} \times_3 \hat{\mathbf{W}}^\mathsf{T}$, which can be solved with SVD of $\mathbf{Y}_{(2)}$

HOOI [DDV00, KD80] works by alternating over factor matrices, updating one at a time by computing leading left singular vectors

Sequentially Truncated Higher-Order SVD

- HOOI needs accurate initialization
- Truncated Higher-Order SVD (T-HOSVD) typically used
- ST-HOSVD [VVM12] is more efficient than T-HOSVD, works by
  - initializing with identity matrices $\mathbf{U} = \mathbf{I}$, $\mathbf{V} = \mathbf{I}$, $\mathbf{W} = \mathbf{I}$
  - applying one iteration of HOOI
  - ranks can be chosen based on error tolerance

ST-HOSVD Algorithm

Input: $\mathcal{X}$

1. $\mathbf{S}^{(1)} \leftarrow \mathbf{X}_{(1)}\mathbf{X}_{(1)}^T$
2. $\mathbf{U} =$ leading eigenvectors of $\mathbf{S}^{(1)}$
3. $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}$
4. $\mathbf{S}^{(2)} \leftarrow \mathbf{Y}_{(2)}\mathbf{Y}_{(2)}^T$
5. $\mathbf{V} =$ leading eigenvectors of $\mathbf{S}^{(2)}$
6. $\mathcal{Z} = \mathcal{Y} \times_2 \mathbf{V}$
7. $\mathbf{S}^{(3)} \leftarrow \mathbf{Z}_{(3)}\mathbf{Z}_{(3)}^T$
8. $\mathbf{W} =$ leading eigenvectors of $\mathbf{S}^{(3)}$
9. $\mathcal{G} = \mathcal{Z} \times_3 \mathbf{W}$

Left singular vectors of $\mathbf{A}$ computed as eigenvectors of $\mathbf{A}^T\mathbf{A}$

Data Distribution and Parallelization

Key data entities in ST-HOSVD are

- tensors: data tensor $\mathcal{X}$, intermediates $\mathcal{Y}$ and $\mathcal{Z}$, core tensor $\mathcal{G}$
- matrices: factor matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$, Gram matrices $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \mathbf{S}^{(3)}$

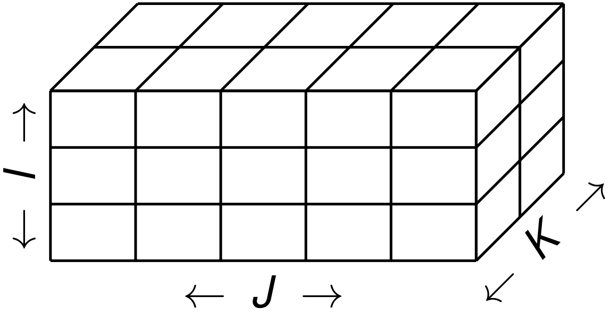
Key computations in ST-HOSVD are

- Gram: computing $\mathbf{S}^{(2)} = \mathbf{X}_{(2)}\mathbf{X}_{(2)}^{\mathsf{T}}$
- Eigenvectors: computing leading eigenvectors of $\mathbf{S}^{(2)}$
- TTM: computing $\mathbf{Y}_{(2)} = \mathbf{V}^{\mathsf{T}}\mathbf{X}_{(2)}$

Parallel Block Tensor Distribution

For N-mode tensor, use logical N-mode processor grid

Proc. grid: $P_I \times P_J \times P_K = 3 \times 5 \times 2$

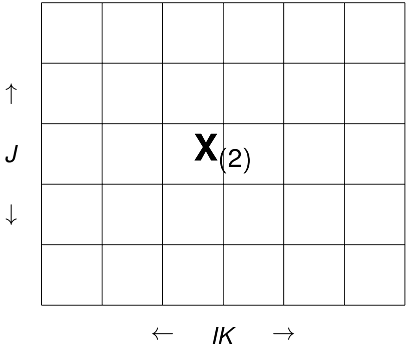


Local tensors have dimensions $\frac{I}{P_I} \times \frac{J}{P_J} \times \frac{K}{P_K}$

Unfolded Tensor Distribution

Key idea: each unfolded matrix is 2D block distributed

Proc. grid: $P_I \times P_J \times P_K = 3 \times 5 \times 2$



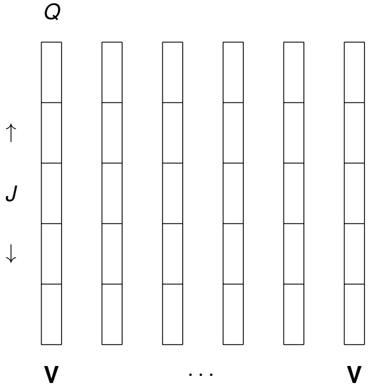
Logical mode-2 2D processor grid: $P_J \times P_I P_K$

Local unfolded matrices have dimensions $\frac{J}{P_J} \times \frac{IK}{P_I P_K}$

Factor Matrix Distribution

Key idea: store factor matrix redundantly across processor fibers

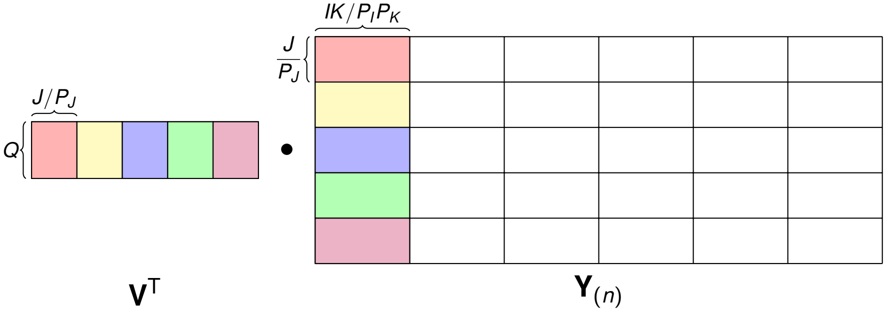
Proc. grid: $P_I \times P_J \times P_K = 3 \times 5 \times 2$



V stored in 1D layout across each fiber of $P_J = 5$ processors

$P_I P_K = 6$ copies of **V** stored

Parallel TTM



Data distributions match TTM computation perfectly

- fibers work independently
- communication consists of reduce-scatter(s) within fiber
- output distributed to match tensor block distribution

ST-HOSVD Algorithm (again)

Input: $\mathcal{X}$

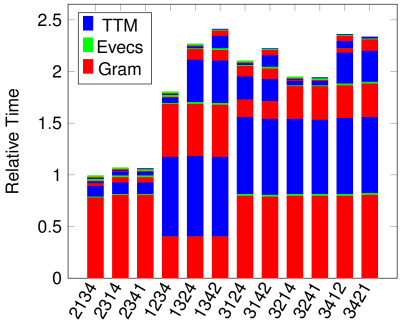
1. $\mathbf{S}^{(1)} \leftarrow \mathbf{X}_{(1)}\mathbf{X}_{(1)}^T$ $\leftarrow$ Gram must also be parallelized
2. $\mathbf{U} =$ leading eigenvectors of $\mathbf{S}^{(1)}$ $\leftarrow$ done redundantly
3. $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}$ $\leftarrow$ TTM makes next step smaller
4. $\mathbf{S}^{(2)} \leftarrow \mathbf{Y}_{(2)}\mathbf{Y}_{(2)}^T$
5. $\mathbf{V} =$ leading eigenvectors of $\mathbf{S}^{(2)}$
6. $\mathcal{Z} = \mathcal{Y} \times_2 \mathbf{V}$
7. $\mathbf{S}^{(3)} \leftarrow \mathbf{Z}_{(3)}\mathbf{Z}_{(3)}^T$
8. $\mathbf{W} =$ leading eigenvectors of $\mathbf{S}^{(3)}$
9. $\mathcal{G} = \mathcal{Z} \times_3 \mathbf{W}$

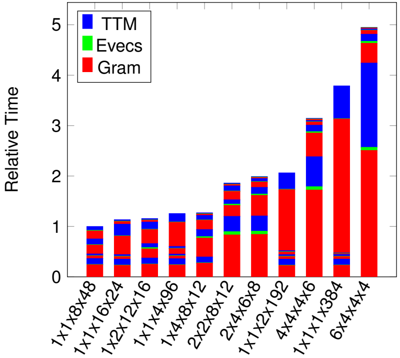
Tuning Parameters

Given a tensor $\mathcal{X}$ and number of processors *P*,

- ST-HOSVD can be performed in any mode order
  - affecting both computation and communication costs
  - yielding (slightly) different results
- *P* can be logically decomposed into many processor grids
  - having large effects on communication cost

Parameter Tuning Experiments

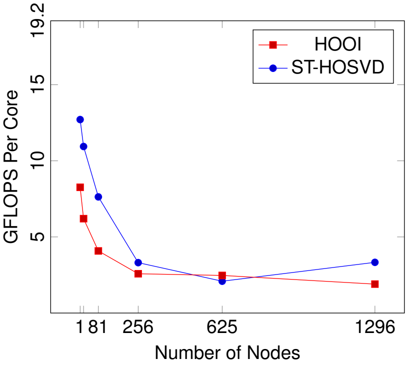


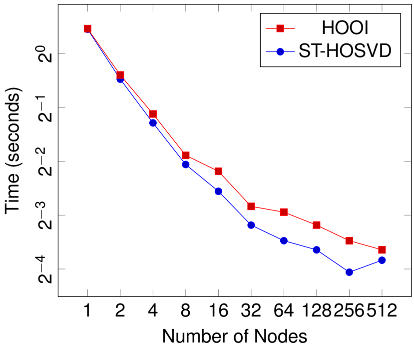


Varying mode order for tensor of size 25×250×250×250 with reduced size 10×10×100×100.

Varying processor grid for tensor of size 384×384×384×384 with reduced size of 96×96×96×96.

Parallel Scaling





Weak scaling for $200k \times 200k \times 200k \times 200k$ tensor with reduced size $20k \times 20k \times 20k \times 20k$, using k^4 nodes for $1 \leq k \leq 6$.

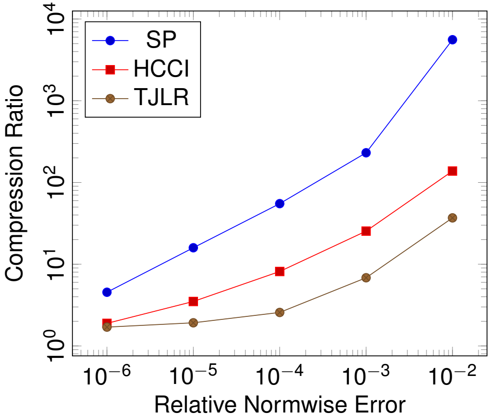
Strong scaling for $200 \times 200 \times 200 \times 200$ tensor with reduced size $20 \times 20 \times 20 \times 20$, using 2^k nodes for $0 \leq k \leq 9$.

Compression of Scientific Simulation Data

We applied ST-HOSVD to compress multidimensional data from numerical simulations of combustion, including the following data sets:

- **HCCI:**
  - Dimensions: $672 \times 672 \times 33 \times 627$
  - 672×672 spatial grid, 33 variables over 627 time steps
  - Total size: 70 GB
- **TJLR:**
  - Dimensions: $460 \times 700 \times 360 \times 35 \times 16$
  - $460 \times 700 \times 360$ spatial grid, 35 variables over 16 time steps
  - Total size: 520 GB
- **SP:**
  - Dimensions: $500 \times 500 \times 500 \times 11 \times 50$
  - $500 \times 500 \times 500$ spatial grid, 11 variables over 50 time steps
  - Total size: 550GB

Compression of Scientific Simulation Data



Compression ratio: $\frac{IJK}{PQR+IP+JQ+KR}$ Relative Normwise Error: $\frac{\|\mathcal{X}-\hat{\mathcal{X}}\|}{\|\mathcal{X}\|}$

Summary

- Tucker is a powerful tool for multidimensional compression
- Large data sets require efficient parallel algorithms
- We propose an algorithm that performs and scales well for dense data sets

For more details:

Parallel Tensor Compression for Large-Scale Scientific Data
Woody Austin, Grey Ballard, and Tamara G. Kolda
International Parallel and Distributed Processing Symposium 2016
`http://arxiv.org/abs/1510.06689`

References

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle.
On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors.
SIAM J. Matrix Analysis and Applications, 21(4):1324–1342, 2000.

Tamara G. Kolda and Brett W. Bader.
Tensor decompositions and applications.
SIAM Review, 51(3):455–500, September 2009.

Pieter M. Kroonenberg and Jan De Leeuw.
Principal component analysis of three-mode data by means of alternating least squares algorithms.
Psychometrika, 45(1):69–97, 1980.

Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen.
A new truncation strategy for the higher-order singular value decomposition.
SIAM Journal on Scientific Computing, 34(2):A1027–A1052, 2012.