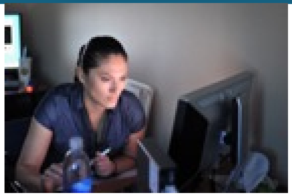
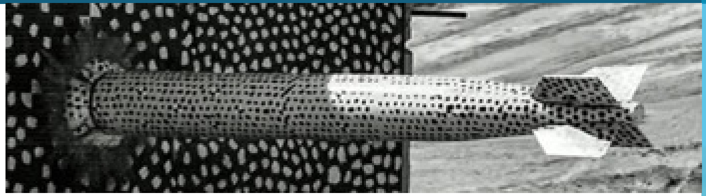
SAND2022-8440C

Sandia National Laboratories

Parallel Memory-Efficient Computation of Symmetric Higher-Order Joint Moment Tensors





Zitong Li liz20@wfu.edu (Wake Forest University)

Hemanth Kolla hnkolla@sandia.gov

Eric Phipps etphipp@sandia.gov

(Sandia National Laboratories)

Platform for Advanced

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Acknowledgments

- ExaLearn Co-design Center, part of the Exascale Computing Project (ECP), U.S. Dept. of Energy Office of Science & National Nuclear Security Administration.
- Base Math Program, Advanced Scientific Computing Research (ASCR), U.S. Dept. of Energy Office of Science.
- Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., for the U.S. Dept. of Energy's National Nuclear Security Administration.
- The views expressed in this presentation do not necessarily represent the views of the U.S. Department of Energy or the United States Government .

Outline

1. Higher-order joint moments and cumulants.
2. Computational aspects and scientific data.
3. Our algorithmic contributions.
4. HPC implementations.
5. Results.

1. Higher-order joint moments and cumulants.
2. Computational aspects and scientific data.
3. Our algorithmic contributions.
4. HPC implementations.
5. Results.

Higher-order joint statistical moments

- Scientific analyses rarely look at marginal/joint moments higher than 2nd (variance/covariance).
  - Financial modelling has used coskewness and cokurtosis.
- For multi-variate non-Gaussian statistics important information is present in higher joint moments.
- Definition: For a vector of random variables, $[\tilde{x}_1, \tilde{x}_1, \dots \dots, \tilde{x}_c]$, centered around mean i.e. $\mathbb{E}(\tilde{x}_1) = 0$:
  - <u>Moments:</u> $\widetilde{m}_{i,j} = \mathbb{E}(\tilde{x}_i\tilde{x}_j), \widetilde{m}_{i,j,k} = \mathbb{E}(\tilde{x}_i\tilde{x}_j\tilde{x}_k), \widetilde{m}_{i,j,k,l} = \mathbb{E}(\tilde{x}_i\tilde{x}_j\tilde{x}_k\tilde{x}_l)$ where $i, j, k, l \in \{1 \dots \dots c\}$
  - <u>Cumulants:</u> $q_{i,j} = \widetilde{m}_{i,j}$, $q_{i,j,k} = \widetilde{m}_{i,j,k}$, $q_{i,j,k,l} = \widetilde{m}_{i,j,k,l} - \widetilde{m}_{i,j}\widetilde{m}_{k,l} - \widetilde{m}_{i,k}\widetilde{m}_{j,l} - \widetilde{m}_{i,l}\widetilde{m}_{j,k}$,
- If the random variables are joint-Gaussian, all cumulants of order > 2 are zero.
- A d^{th}-order moment/cumulant is a 'supersymmetric' tensor of order- d:
  - e.g. for a 3rd-order moment, $\widetilde{m}_{i,j,k} = \widetilde{m}_{i,k,j} = \widetilde{m}_{j,i,k} = \widetilde{m}_{j,k,i} = \widetilde{m}_{k,i,j} = \widetilde{m}_{k,j,i}$ for any i, j, k.

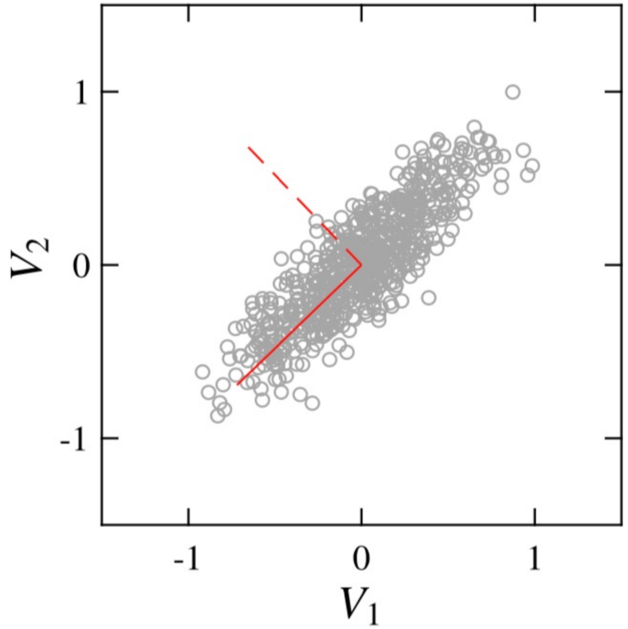
Applications of higher-order joint moments

- Hitherto mostly used in financial modelling; portfolio risk assessment and asset pricing.
- Independent Component Analysis (ICA) algorithms based on eigen decomposition of 4th-order cumulants (Cardoso 1989, Comon & Cardoso 1990).
- ICA has been used for assessment of climate models (Fodor & Kamath 2003), source identification in stream water temperatures (Middleton *et al.*, 2015).
- Hyperspectral imaging: band selection and small target detection (Geng *et al.*, 2015, Głomb *et al.*, 2018), ICA-based dimensionality reduction (Wang & Chang 2006).
- Medical electrodiagnostics: artifact detection in EEG (Delorme *et al.*, 2007), feature identification in EMG (Domino *et al.*, 2019), feature extraction and classification in ECG (Yu & Chou 2008, Kutlu & Kuntalp 2012).
- Anomaly detection in multi-variate data (Peña & Prieto 2001, Konduri *et al.*, 2019).

Information in higher-order statistical moments

For non-Gaussian multi-variate statistical processes higher-order joint moments are informative

(co-skewness is 3rd-order tensor, co-kurtosis is 4th-order tensor)

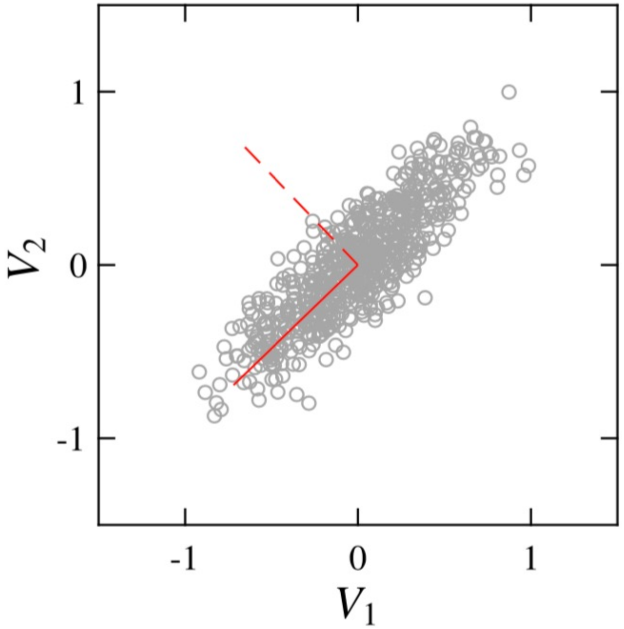


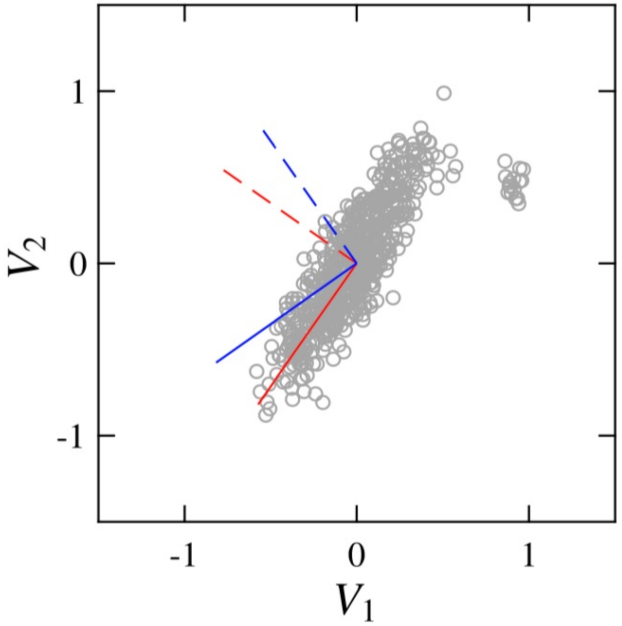
Red: Eigenvectors of Covariance (Principal Component Analysis). Denote directions of maximal variance

Information in higher-order statistical moments

For non-Gaussian multi-variate statistical processes higher-order joint moments are informative

(co-skewness is 3rd-order tensor, co-kurtosis is 4th-order tensor)



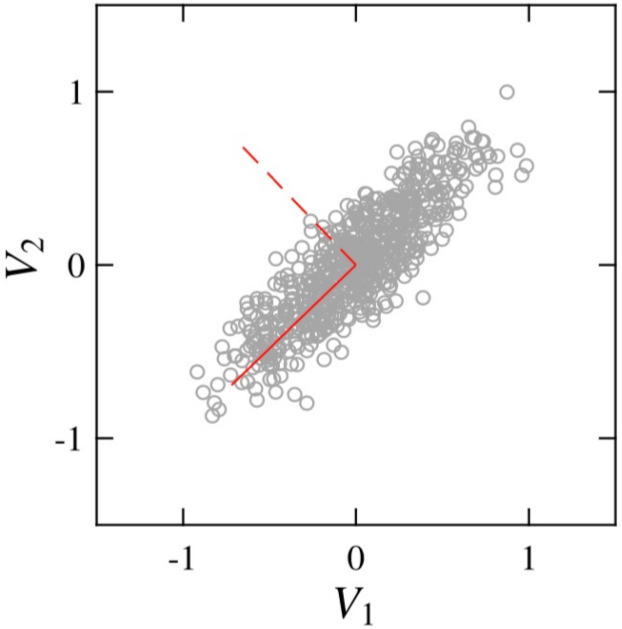


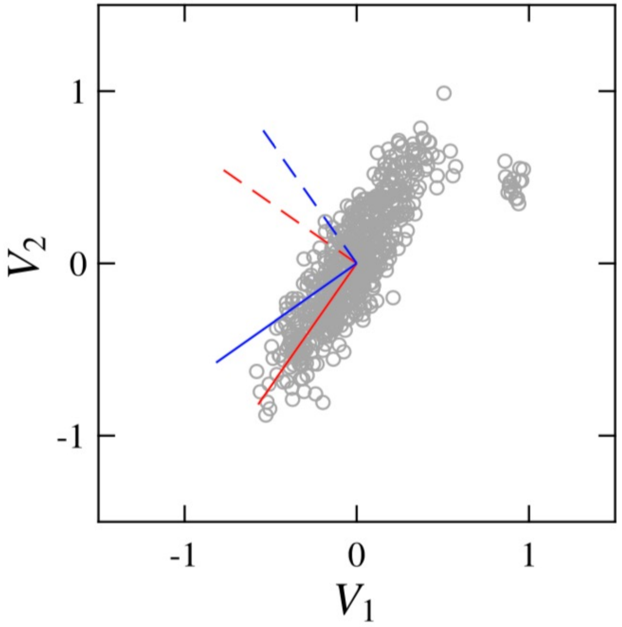
Red: Eigenvectors of Covariance (Principal Component Analysis). Denote directions of maximal variance

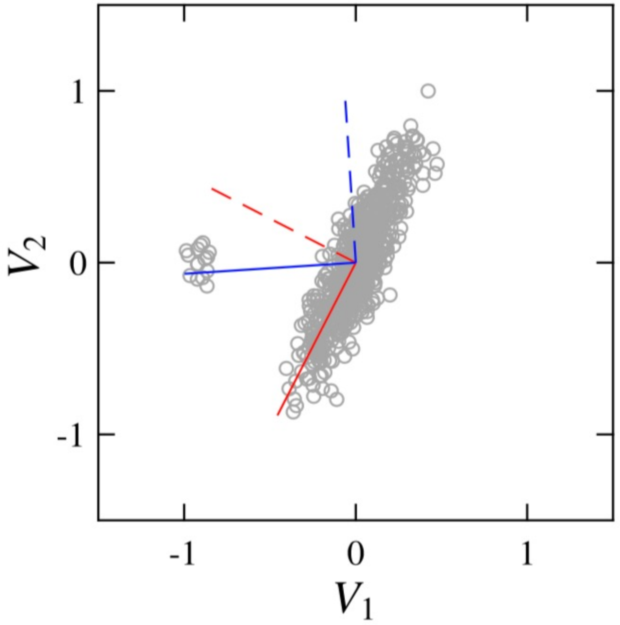
Blue: 'Principal Kurtosis Vectors'. Obtained through HOSVD of co-kurtosis tensor.

Information in higher-order statistical moments

PCA vectors not sensitive to outliers, Principal Kurtosis Vectors are.







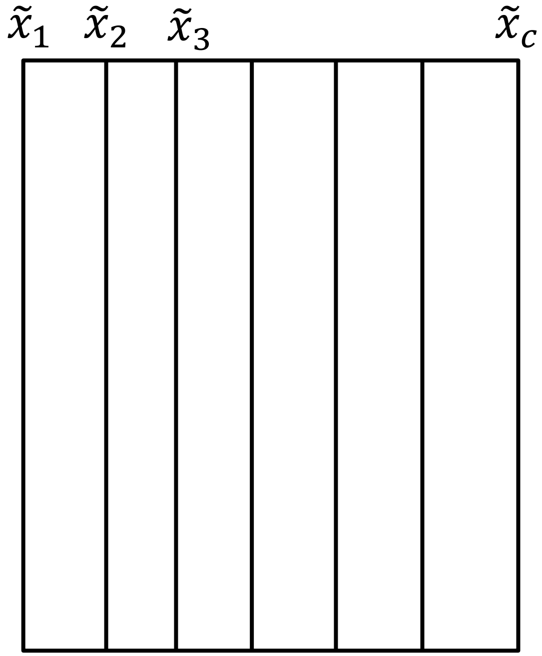
Red: Eigenvectors of Covariance (Principal Component Analysis). Denote directions of maximal variance

Blue: 'Principal Kurtosis Vectors'. Obtained through HOSVD of co-kurtosis tensor.

1. Higher-order joint moments and cumulants.
2. Computational aspects and scientific data.
3. Our algorithmic contributions.
4. HPC implementations.
5. Results.

Naïve computation of moment tensor

$X \in \mathbb{R}^{r \times c}$




```
for i1 = 1:c
  for i2 = 1:c
    ...........
      for id = 1:c
        for row = 1:r
          m_{i1,i2,..id} += X(row,i1)*....*X(row,id)
```

- Input matrix, r – grid points, time steps
- Typically $r \gg c$

- Naïve computation of d^{th}-order moment .
- Computational complexity $\sim\mathcal{O}(rdc^d)$.

Leveraging symmetry

- Symmetry: Full moment tensor has c^d elements, but many are duplicated
  - Number of unique elements: $\binom{c+d-1}{d}$
- Blocked Compact Symmetric Storage (BCSS) (Schatz *et al.*, 2014).
  - Number of blocks to compute: $\binom{(c/s)+d-1}{d}$
  - Number of elements per block: s^d
  - Number of elements to be computed: $s^d\binom{(c/s)+d-1}{d}$
  - Potential savings $\sim\mathcal{O}(d!)$

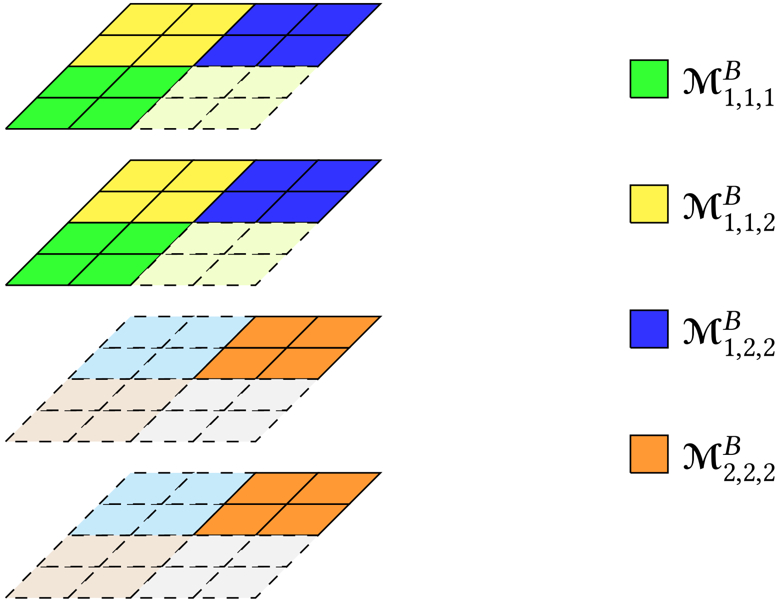
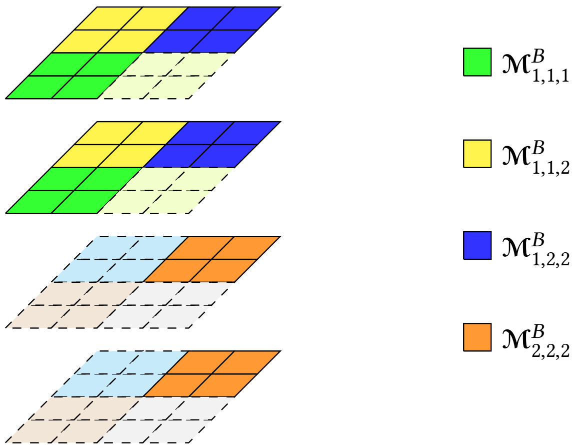


Fig. A symmetric 4x4x4 tensor divided into 8 blocks of 2x2x2 as per the BCSS format

Leveraging symmetry

- Domino *et al.*, (2018) leverage symmetry and BCSS to compute only unique subset of blocks.
- Focus was on computation of cumulant tensors:
  - Presented a formula for $\mathcal{C}_d = f(\mathcal{M}_d, \mathcal{C}_2, \ldots, \mathcal{C}_{d-2})$
  - Involves sum of outer-products of $\mathcal{C}_2, \ldots, \mathcal{C}_{d-2}$
- Compute moment tensor ($\mathcal{M}_d$) subblocks using nested loops
- Parallelize along the row dimension
- Speedup of $\mathcal{M}_4 \sim 24$ (relative to naïve full tensor computation)
- Speedup of $\mathcal{C}_4 \sim 100$




```
for i1 = 1:s
  for i2 = 1:s
    ..........
      for id = 1:s
        for row = 1:r
          mi1,i2,..id += X(row,i1)*....*X(row,id)
```

Refactoring of moment tensor block computation

- We propose refactoring using following ingredients:
  - $x_j^{\circ d} = \underbrace{x_j \circ x_j \circ \cdots x_j}$ (d-way outer product)
  - Matricisation of a d-way symmetric tensor: $Mat_p : \mathbb{R}^{(c \times c \times \cdots c)} \rightarrow \mathbb{R}^{c^p \times c^{d-p}}$
  - Matrix Khatri-Rao product $\odot$ - Kronecker product of individual columns of two matrices
- Motivation: $\mathcal{M}_d$ can be expressed as sum of outer products of row vectors of X (Sherman & Kolda 2020):
  - $\mathcal{M}_d = \frac{1}{r}\sum_{j=1}^{r} \boldsymbol{x}_j^{\circ d}$, where $x_j = \mathrm{X}[\, j, :],$
  - Can be easily pictured for covariance: $\mathcal{M}_2 = {}^1/_r \, (\mathrm{X}^T\mathrm{X})$
- Final result (proof & details in paper): $Mat_p(\mathcal{M}) = \frac{1}{r}\left(\bigodot^{p} X^T\right)\left(\bigodot^{d-p} X^T\right)^T$
- Recommendation: $p = d/2$

Refactoring of moment tensor block computation

- Refactored expression applies identically to sub moment tensor sub blocks: $Mat_p(\mathcal{M}) = \frac{1}{r}\left(\bigodot^{p} X^T\right)\left(\bigodot^{d-p} X^T\right)^T$
  - Input would be column slices of X instead of the entire matrix

- Computational complexity:

Khatri-Rao products — gemm — # of blocks

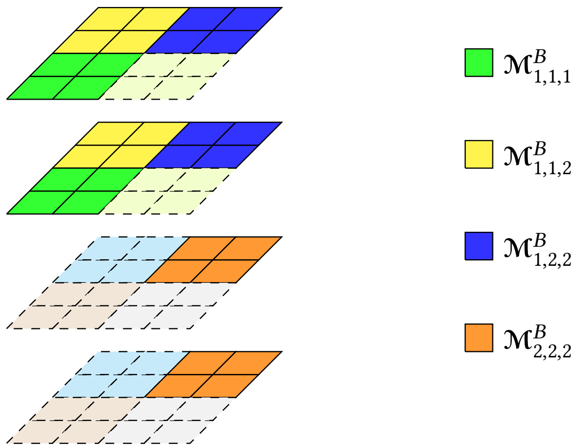
$$\left(2rs^{d/2} + (2r-1)s^d \right) \binom{(c/s)+d-1}{d}.$$

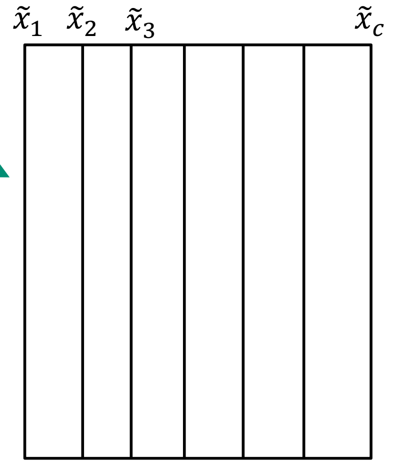
  - Less expensive than naïve (nested for-loops) by factor $\approx d/2$.
- Cache complexity: Matrix Khatri-Rao products are very cache-friendly, leading to fewer cache misses:
  - We recommend X to be row-major order. Matrix transpose (if need be) is $\sim\mathcal{O}(\frac{rc}{L})$ cache misses.
  - Refactoring incurs fewer cache misses by at least $\sim\mathcal{O}(s^{d/2})$ (various scenarios detailed in paper).

1. Higher-order joint moments and cumulants.
2. Computational aspects and scientific data.
3. Our algorithmic contributions.
4. HPC implementations:
5. Results.

Hierarchical parallelism

- Each subblock of $\mathcal{M}_d$ can be computed independently:
  - For distributed-memory this could be along nodes/processes/ranks.
  - For shared-memory this can be along thread parallel units (e.g. GPU warps).
- Traversal along row-dimension of X can be parallel-reduced:
  - For distributed-memory this is aligned with domain decomposition.
  - For shared-memory this can be tiling (to reduce memory requirements).
- The matrix Khatri-Rao product (due to refactoring) exposes another level of parallelism.





Performance portable implementation with Kokkos

- Kokkos – a C++library for performance portable implementations.
- Detailed semantics to express data and compute parallelism. Hardware details of memory layout and parallelism units are abstracted out.
- We provide Kokkos implementations with three levels of parallelism:
  - "Thread Team" parallelism (e.g. GPU warp)
  - Tile parallelism (row dimension of X)
  - Thread-level parallelism

Algorithm 3 Parallel Algorithm for computing the 4th moment tensor

function $\mathcal{M}^{(4)}$ = 4THMOMENTTENSOR($\mathbf{X}, n, t, s$)
  ▷ s = block size, t = tile size, n = # of teams, $\mathbf{X} \in \mathbb{R}^{r\times c}$
  $\bar{r} = r/t$ ▷ # of row tiles
  $nbm = ceil(c/s)$ ▷ # of blocks on each mode
  $nb = \binom{nbm+3}{4}$ ▷ total # of unique blocks in $\mathcal{M}^{(4)}$
  $\overline{nb} = nb/n$ ▷ # of blocks each team computes
  teams_parallel_for($i = 1, \ldots, n$)
    [$b_s = (i-1)\overline{nb} + 1, b_e = i(\overline{nb})$] ▷ blocks of this team
    for $j = b_s, \ldots, b_e$ **do**
      $\mathbb{T}(j; nbm) \mapsto (i_1, i_2, i_3, i_4)$ ▷ multi-index this block
      $\mathbf{X}_{i_l} = \mathbf{X}_{:,(i_l-1)s+1:i_l s}, i_l = (i_1, \ldots i_4)$ ▷ column slices
      for $k = 1, \ldots, \bar{r}$ **do**
        $\mathbf{X}_{k,i_l} = \mathbf{X}_{(k-1)t+1:kt,\, i_l}$ ▷ k^{th} row tile
        $\mathbf{Y}$ = KHATRIRAOPRODUCT($\mathbf{X}^T_{k,i_2}, \mathbf{X}^T_{k,i_i}$)
        $\mathbf{Z}$ = KHATRIRAOPRODUCT($\mathbf{X}^T_{k,i_4}, \mathbf{X}^T_{k,i_3}$)
        $\mathcal{M}^B_{(i_1,i_2,i_3,i_4)} \mathrel{+}= \frac{1}{r}\mathbf{Y}\mathbf{Z}^T$

Algorithm 2 Parallel Algorithm for the Khatri-Rao Product

function $\mathbf{C}$ = KHATRIRAOPRODUCT($\mathbf{A}, \mathbf{B}$)
  ▷ $\mathbf{A} \in \mathbb{R}^{a\times c}, \mathbf{B} \in \mathbb{R}^{b\times c}$
  threads_parallel_for($i = 1, \ldots, ab$)
    vector_parallel_for($j = 1, \ldots, c$)
      $(\alpha, \beta) = \mathbb{T}(i; a, b)$
      $\mathbf{R}_{i,j} = \mathbf{X}_{\alpha,j}\mathbf{X}_{\beta,j}$

1. Higher-order joint moments and cumulants.
2. Computational aspects and scientific data.
3. Our algorithmic contributions.
4. HPC implementations.
5. Results.

Serial performance

- All experiments for 4th-order moment tensor, $\mathcal{M}_4$.
- Serial Julia implementation compared against reference implementation (Domino *et al.*, 2018), AMD EPYC 7302.
- Parameters: r – rows of X, c – cols of X, s – size of each subblock

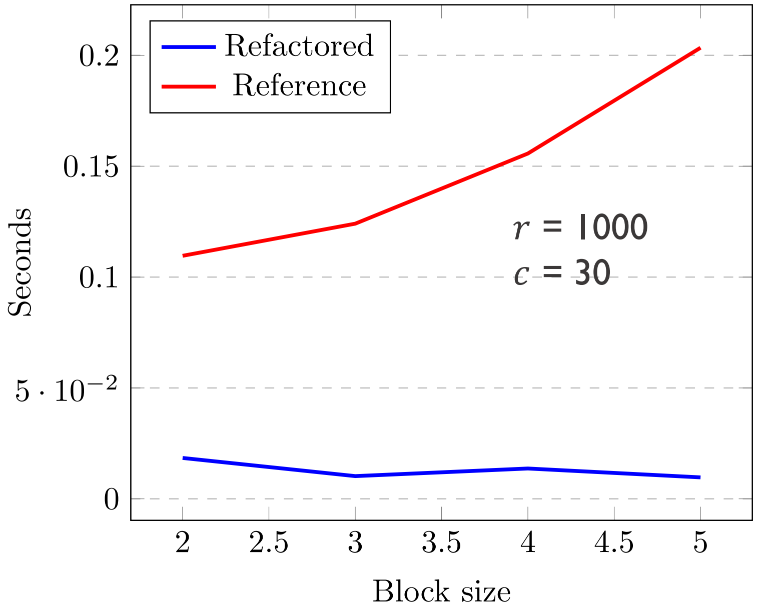


Fig. Trend with varying block size

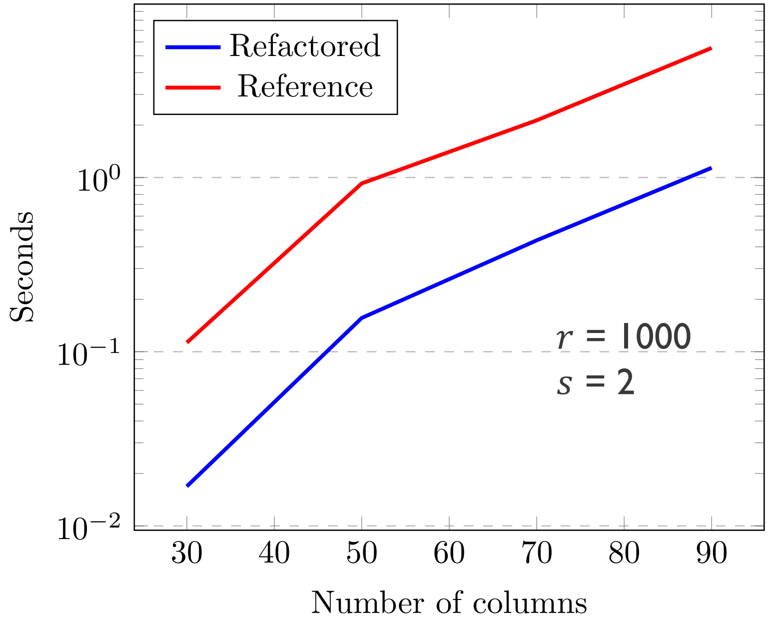


Fig. Trend with varying #columns

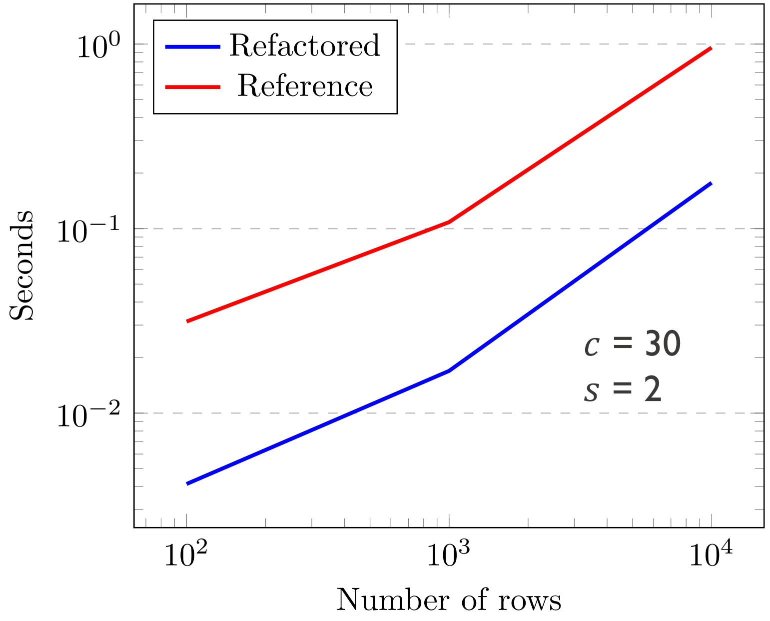


Fig. Trend with varying #rows

Parallel performance

- All experiments for 4th-order moment tensor, $\mathcal{M}_4$.
- Kokkos implementation compared against reference implementation (Domino *et al.*, 2018), NVIDIA Tesla V100.
- Parameters: r – rows of X, c – cols of X, s – size of each subblock

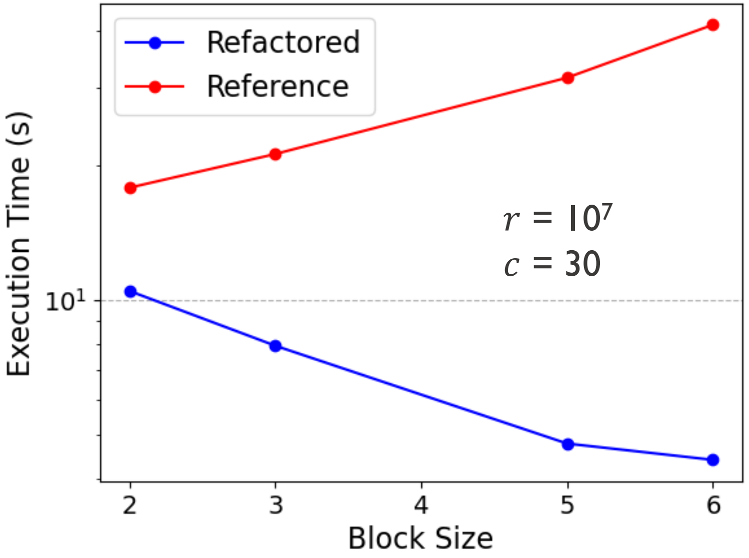


Fig. Trend with varying block size

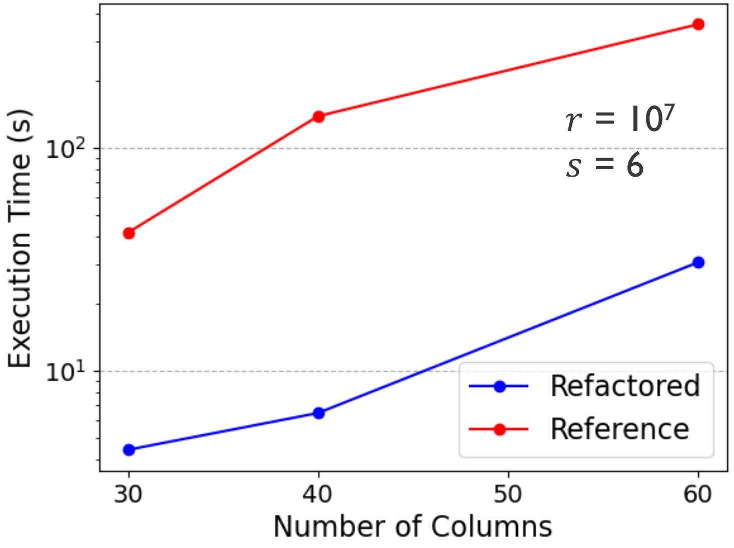


Fig. Trend with varying #columns

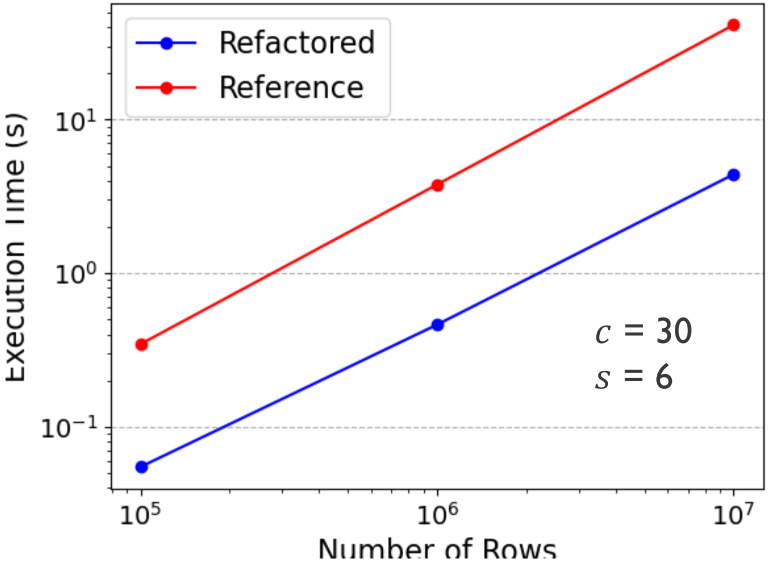


Fig. Trend with varying #rows

Cache performance

- All experiments for 4th-order moment tensor, $\mathcal{M}_4$.
- Kokkos implementation compared against reference implementation (Domino *et al.*, 2018), NVIDIA Tesla V100.
- Flops and MemOps (L1-cache) measured with `nvprof`.

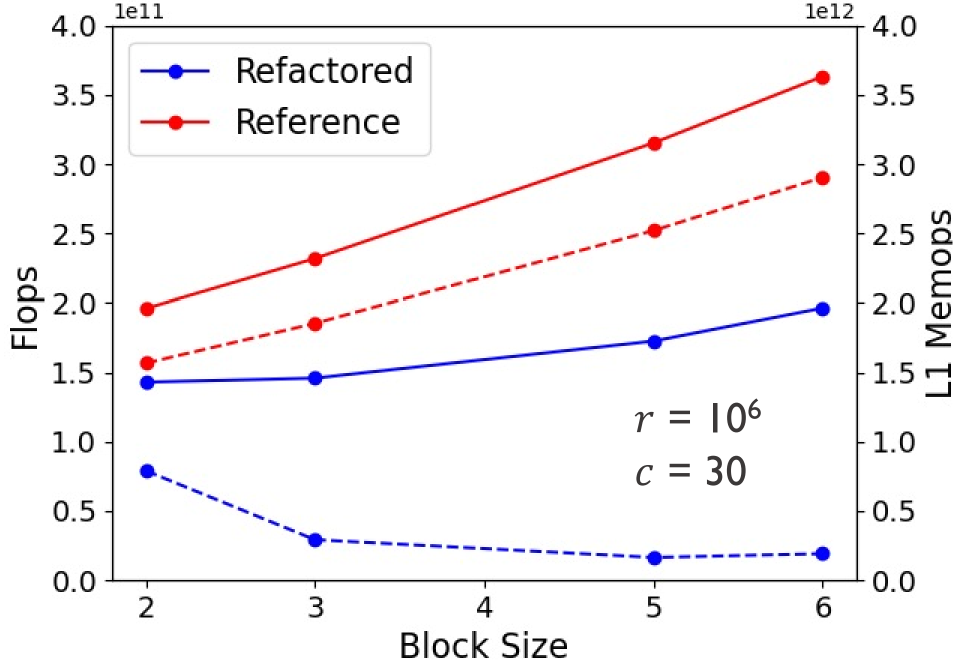


Fig. Trend with varying block size

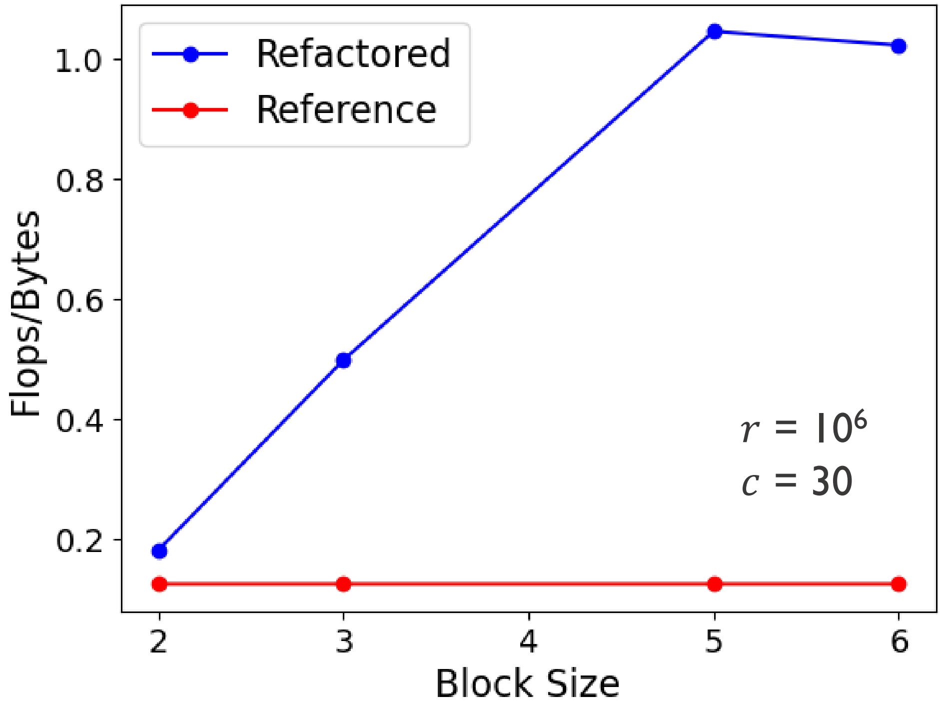


Fig. Trend with varying block size

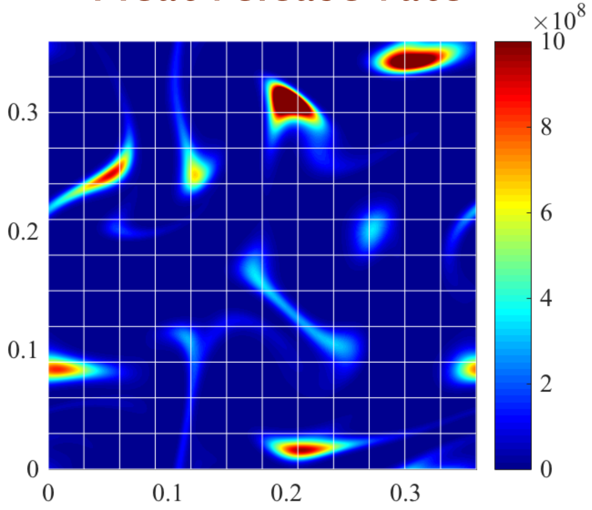
Thank You

Backup

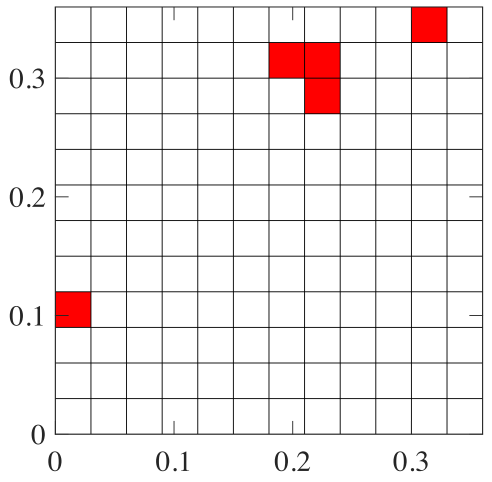
Formalizing Distributed Rare Event Detection

- Compute Principal Kurtosis Vectors on each data partition (e.g. processor).
- Compare the vectors amongst partitions in space and/or time:
  - Proposed Feature moment metrics (fraction of the kurtosis attributable to each variable) to quantify orientation of Kurtosis vectors.
  - FMMs sum to unity, akin to discrete distribution.
  - Divergence metric (Hellinger distance) to compare across partitions.
- Most computation (cokurtosis tensor and principal vectors) is local.
- Communication only of a small vector of numbers (FMMs).

Heat release rate



Anomalous partitions

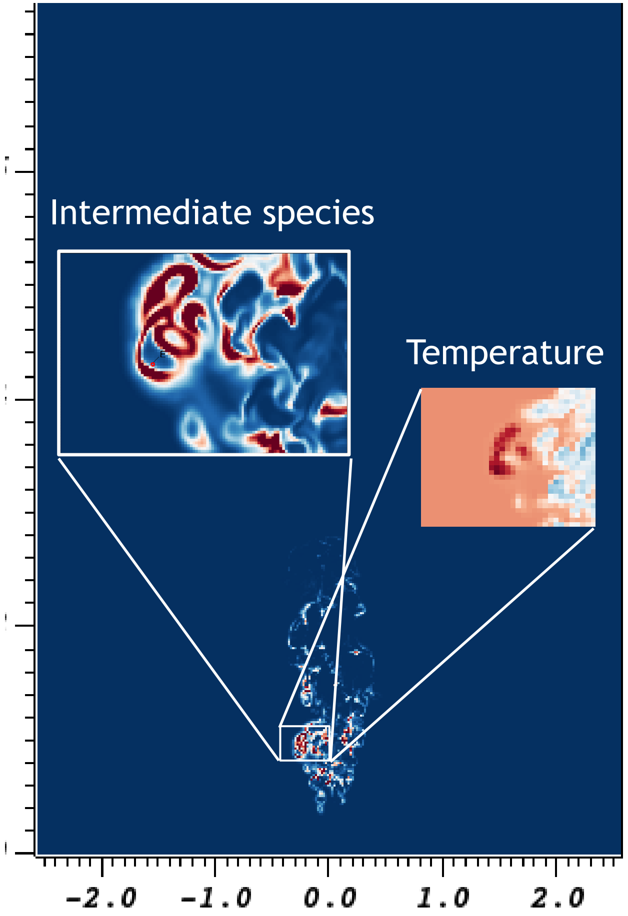


"[Anomaly detection in scientific data using joint statistical moments](#)." K. Aditya, H. Kolla, W.P. Kegelmeyer, T.M. Shead, J. Ling, W.L. Davis IV, Journal of Computational Physics, 2019.

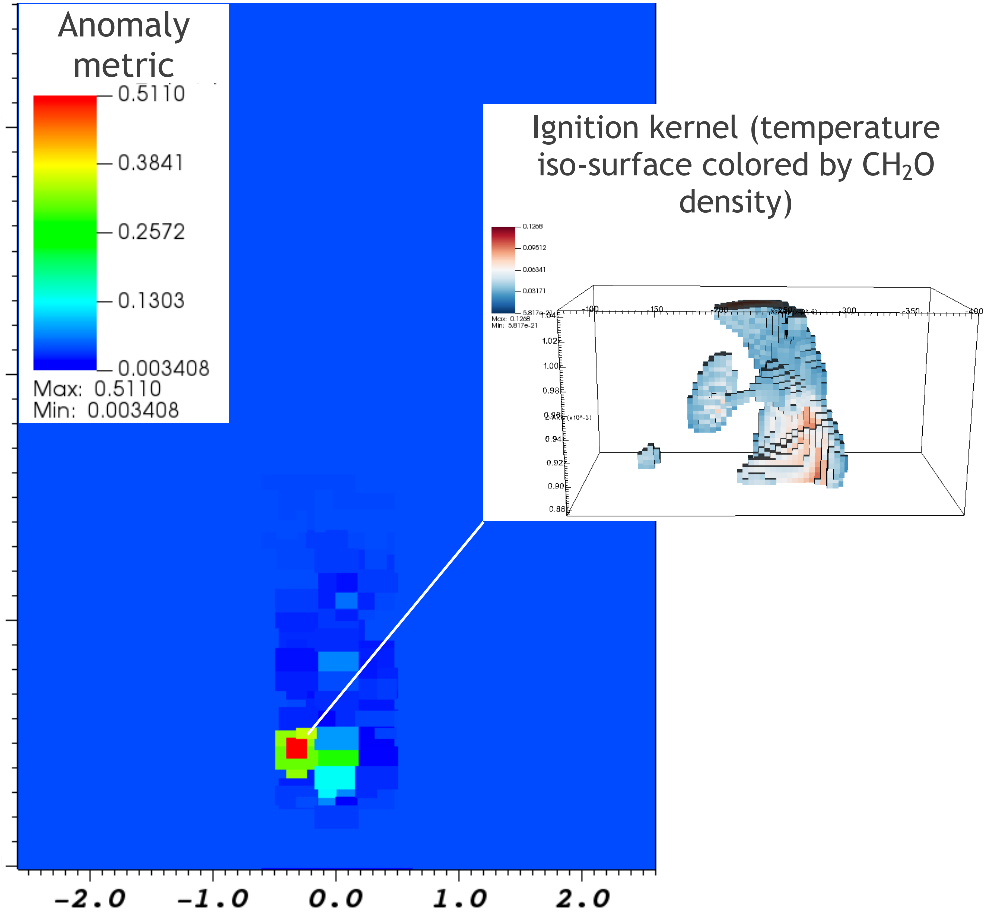
Ignition detection in exascale combustion code

<u>Pele</u>: PeleLM, adaptive-mesh low Mach number hydrodynamics code for reacting flows

Identification currently based on ad-hoc thresholds



Validation: co-kurtosis tensor-based unsupervised anomaly detection



Contributors: Martin Rieth, Jackie Chen, Marco Arienti, Janine Bennett (Sandia Natl. Labs), Matt Larsen (formerly at LLNL)