





Rapid Analysis and Management of Large Spatiotemporal Data

Advanced Modeling & Simulation (AMS) Seminar Series
NASA Ames Research Center, November 30th, 2023

Motivation and Vision

Problem:

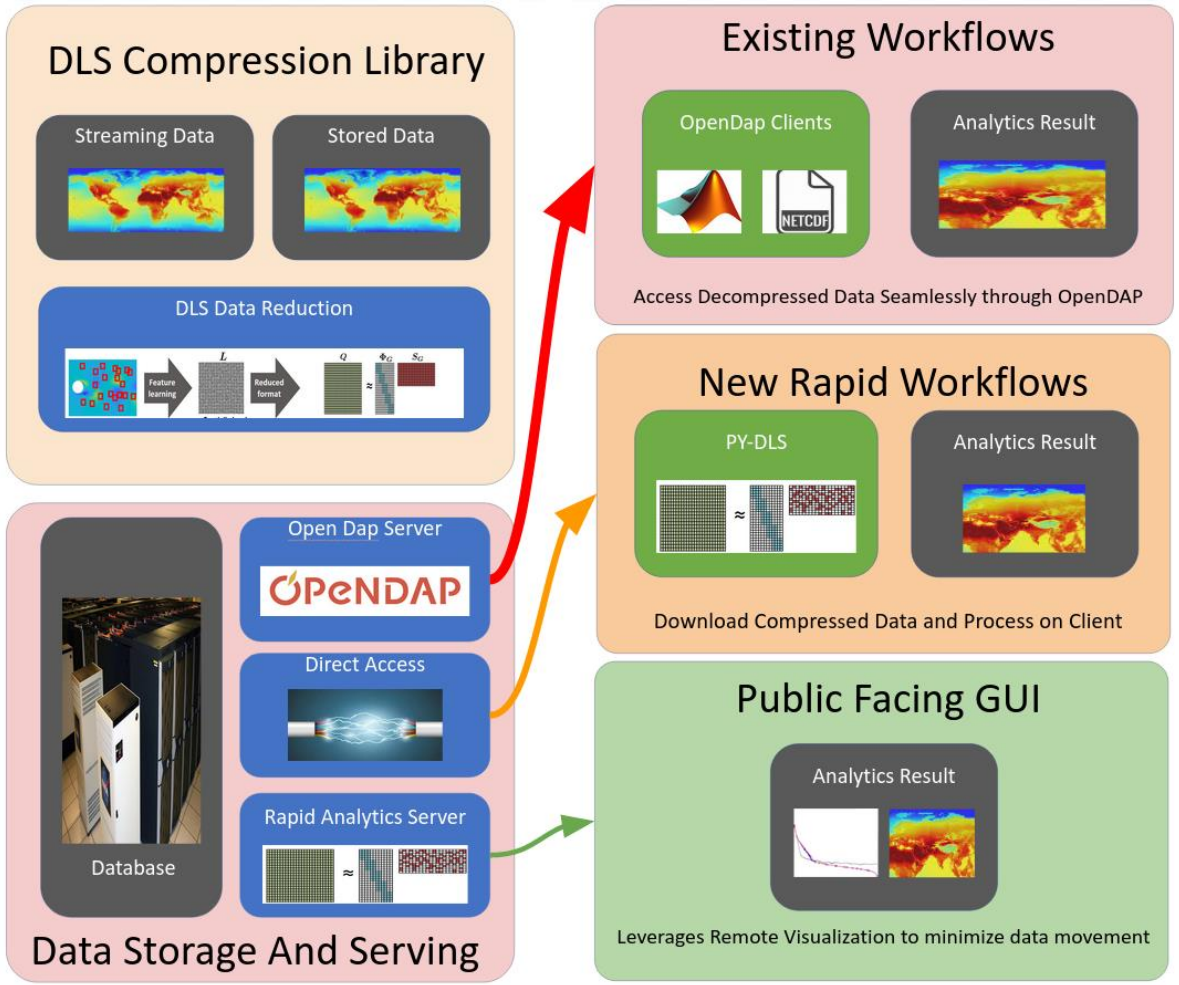
- High Fidelity Scientific Data is expensive to store
- Existing lossy compression algorithms are inaccurate beyond 32x.
- Analyzing high fidelity data remotely is slow.
- Not everyone is an expert!

Solution:

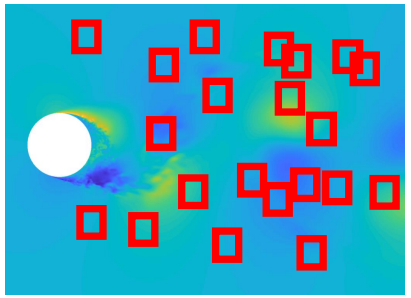
- Data driven Spatiotemporal data reduction for high compression ratios with user defined error bounds.
- Minimal data movement
- Leverage separation of variables for rapid analytics
- Include Interfaces for Novice, Intermediate and Expert users

Benefits:

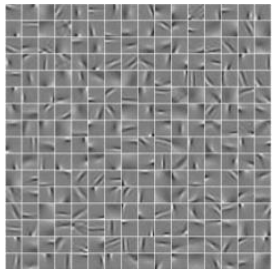
- Lowers hosting and storage costs for data producers
- Decreased Bandwidth usage for data consumers
- Faster Data Analytics for everyone



Methodology: Reduce-then-Analyze

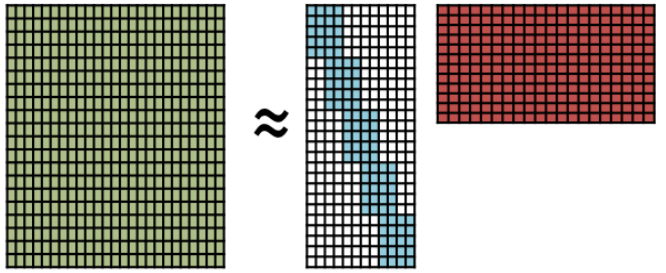


Feature learning



Spatially local features

Reduced format

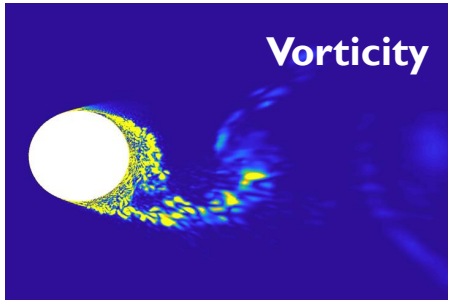


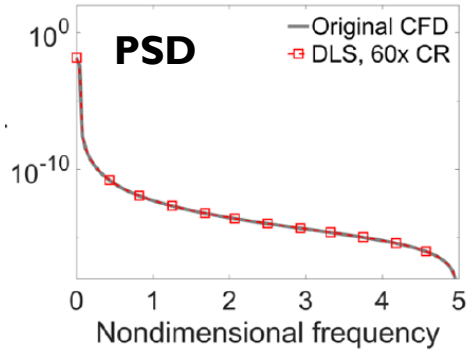
Highly sparse, generalized, space-time decomposition: Data-informed Local Subspaces (DLS)

Query/Analysis

- The reduced data format consists of a spatial and temporal separation of variables

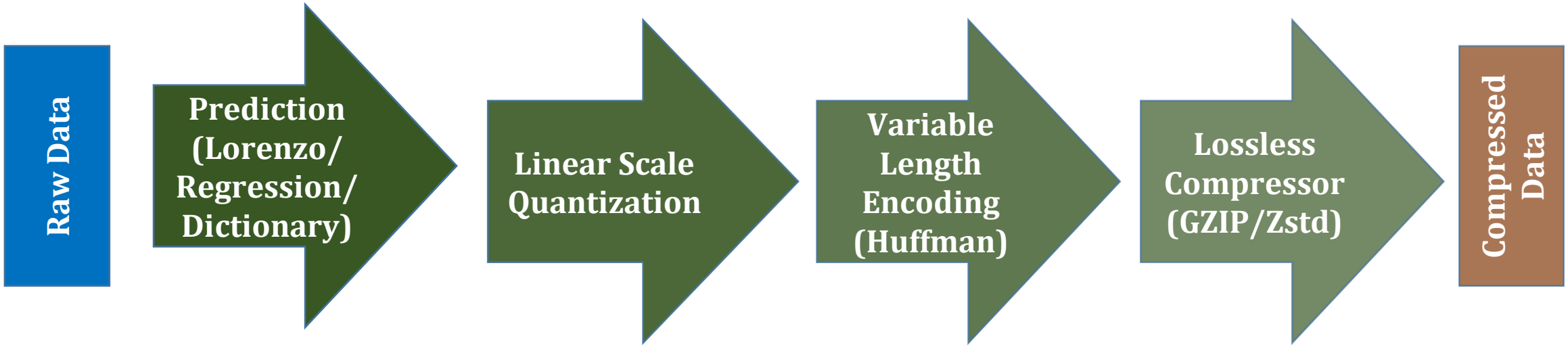
- Efficiency from:
  - Reduced data sizes (20-1000x)
  - Batch processing of basis and coeffs.
  - Ease of accessing data subsets





Efficient Compression of Spatiotemporal Data

Typical Lossy Compression Workflow

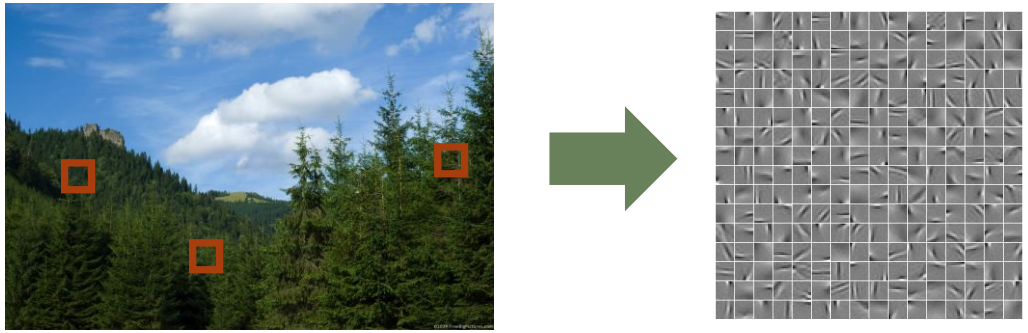


- Typical compression ratios from 5x to 20x for highly multiscale data
- The data prediction step in many compressors (such as SZ) is data-agnostic
- **Our Solution:** **A data-driven prediction step**
- **Hypothesis:** **Data-specific prediction will give better compression**

Data Driven Prediction

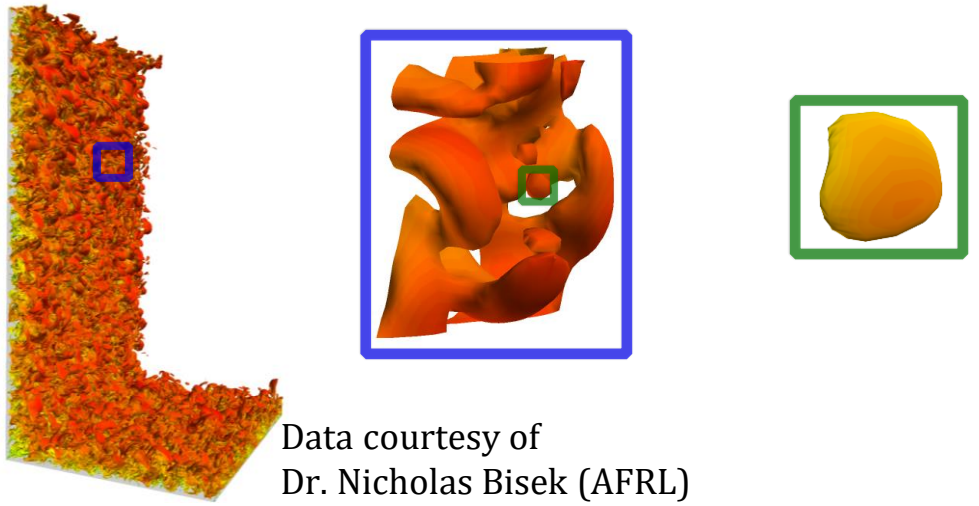
Feature learning from redundant structures in natural images:

- Learn compact set of features that explain large, global, images



Similar statistics found in multiscale scientific data.

- Feature learning from large flowfields
  - learn spatially small, simpler structures →
  - more flexibility



Data courtesy of
Dr. Nicholas Bisek (AFRL)

Data-informed Local Subspaces (DLS)

Generalized Finite Element Method

[Babuska et al., 1994; Duarte and Oden, 1995]

$$\boldsymbol{u_h}(\boldsymbol{x},t) = \left\{ \sum_{\alpha=1}^{n_{FEM}} \hat{\boldsymbol{u}}_\alpha(t)\varphi_\alpha(\boldsymbol{x}) \right\} \quad \textbf{FEM}$$

$$+ \left\{ \sum_{\alpha=1}^{n_{FEM}} \varphi_\alpha(\boldsymbol{x}) \sum_{i=1}^{K} \tilde{\boldsymbol{u}}_{\alpha i}(t)\boldsymbol{L_i}(\boldsymbol{x}) \right\} \quad \textbf{Enrichment of FEM}$$



FEM function:

$\varphi(x)$



Enrichment functions:

$L_1(x)$



$L_2(x)$



GFEM functions:

$\varphi(x)$



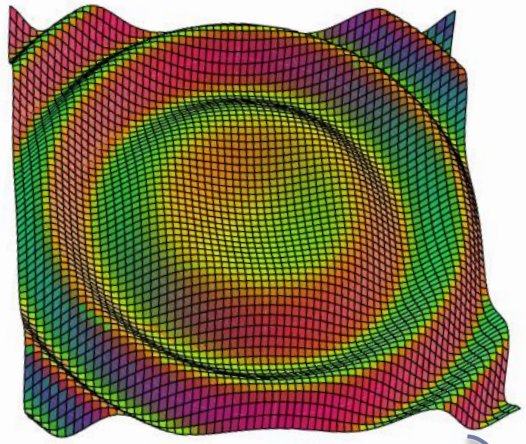
$\varphi(x)L_1(x)$



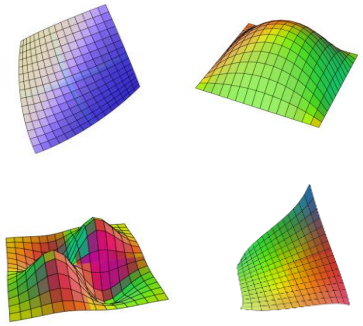
$\varphi(x)L_2(x)$



Data-informed Local Subspaces (DLS)

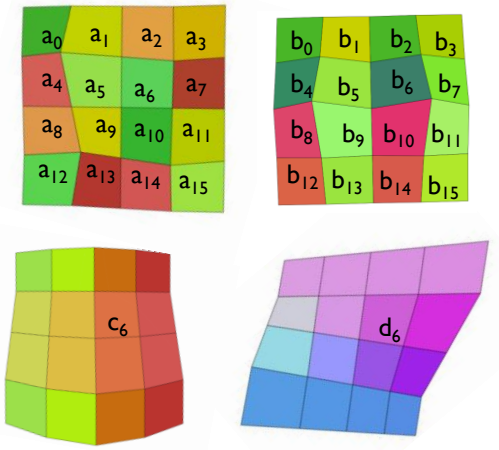


DLS

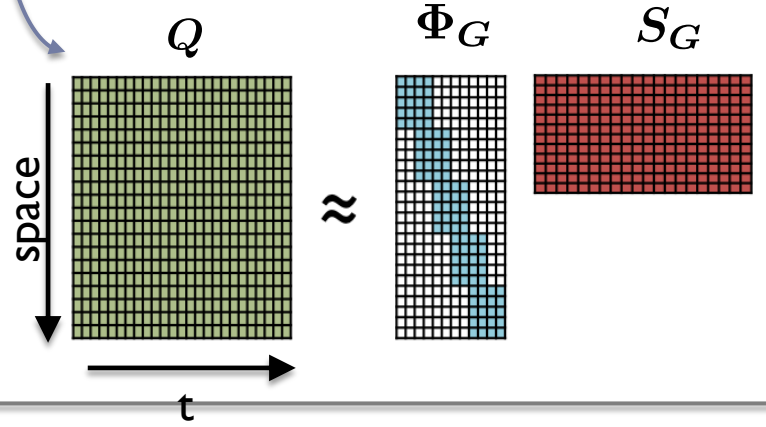


Enrichment Functions
(const for all time)

&



Degrees of Freedom defined on coarse grid



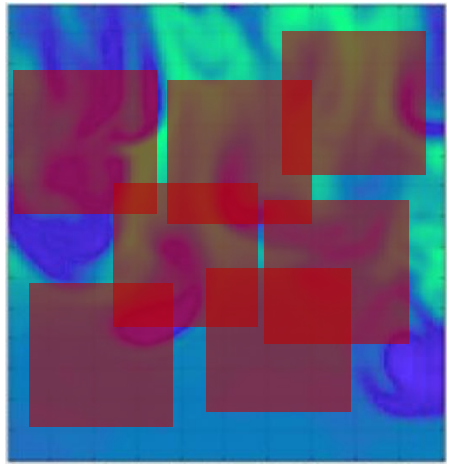
Takeaway: DLS maps the fine grid solution into an enriched finite element space defined on a coarse grid!

Phase I: Enrichment Functions

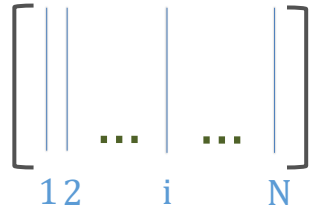
Finite elements + data-derived features

- Training data consist of local patches randomly sampled from **a high-fidelity solution → reduced cost of feature generation**
- Currently using SVD based decomposition
- Sparse coding option is also available
- Deep learning is being explored

High Fidelity Training Solution



Local Snapshot Matrix (Q_{local})



Decompose

$$Q_{local} \approx L\ S$$

$$u_h(x,t) = \left\{ \sum_{\alpha=1}^{n_{FEM}} \hat{u}_\alpha(t)\varphi_\alpha(x) \right\} \text{ FEM}$$

$$\left\{ \sum_{\alpha=1}^{n_{FEM}} \varphi_\alpha(x) \sum_{i=1}^{K} \tilde{u}_{\alpha i}(t) L_i(x) \right\} \text{ Enrichment of FEM}$$

Enrichment functions:

$L_1(x)$ 

$L_2(x)$ 

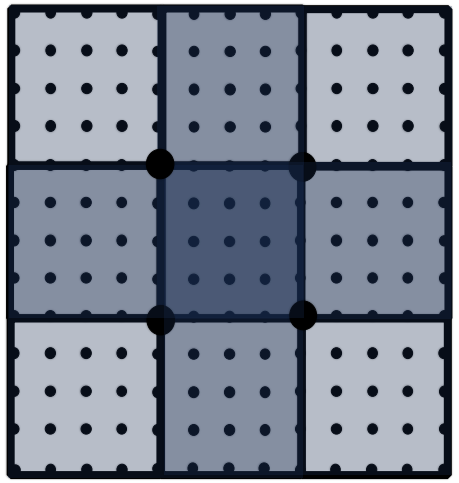
Phase II: Compression Matrix

$$\Phi^T \Phi S = \Phi^T Q$$

- Block Banded matrix.
- One block per coarse grid element
- Symmetric Positive Semidefinite
- Cholesky gives lower triangular B such that

$$\Phi^T \Phi = B B^T$$

- Reuse same decomposition to compress every single time step.



Enrichment functions:

$L_1(x)$ 

$L_2(x)$ 

$$u_h(x,t) = \left\{ \sum_{\alpha=1}^{n_{FEM}} \hat{u}_\alpha(t)\varphi_\alpha(x) \right\} \text{ FEM}$$

$$\left\{ \sum_{\alpha=1}^{n_{FEM}} \varphi_\alpha(x) \sum_{i=1}^{K} \tilde{u}_{\alpha i}(t) L_i(x) \right\} \text{ Enrichment of FEM}$$

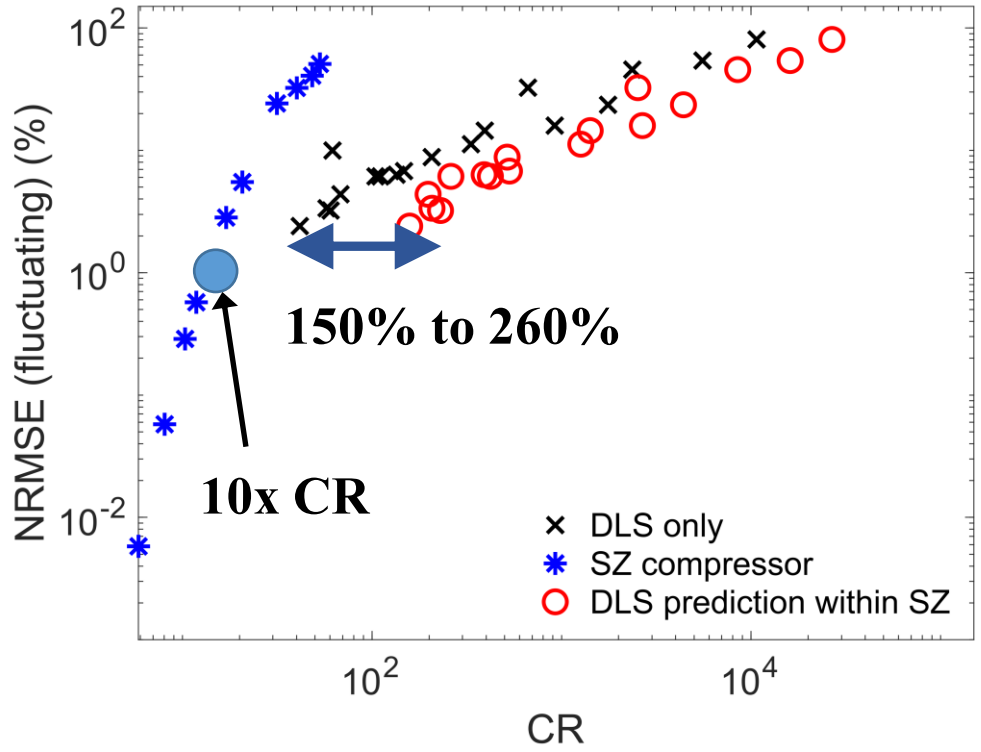
Phase III Compression:

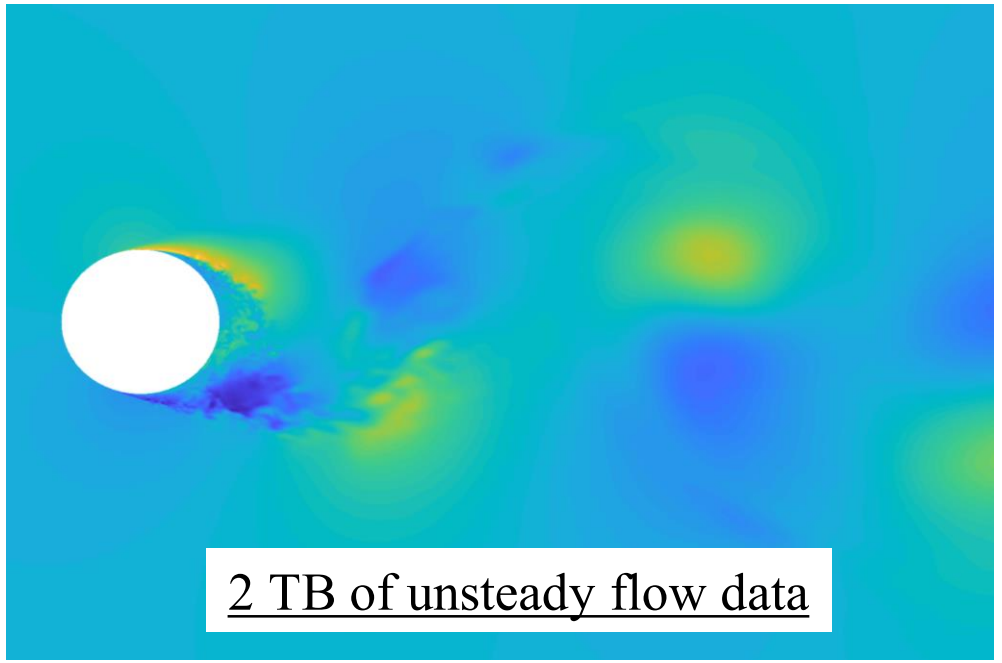
For each fine grid snapshot in time $\mathbf{q_i}$:

Solve:

$$B B^T \mathbf{s_i} = \Phi^T \mathbf{q_i}$$

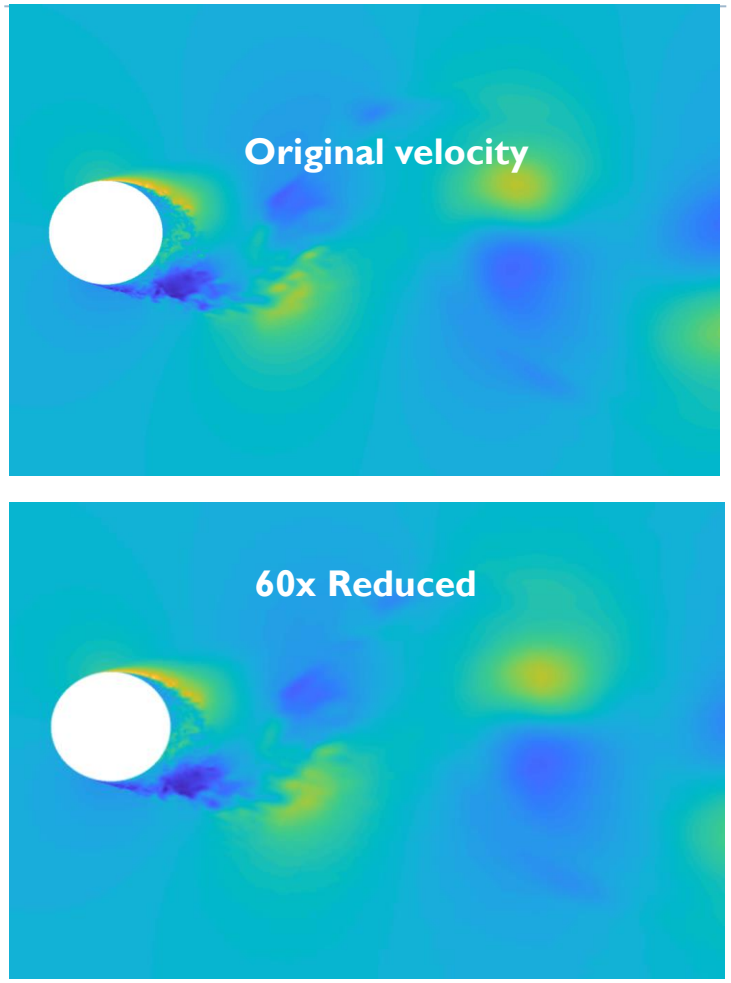
DLS + Quantization + GZIP

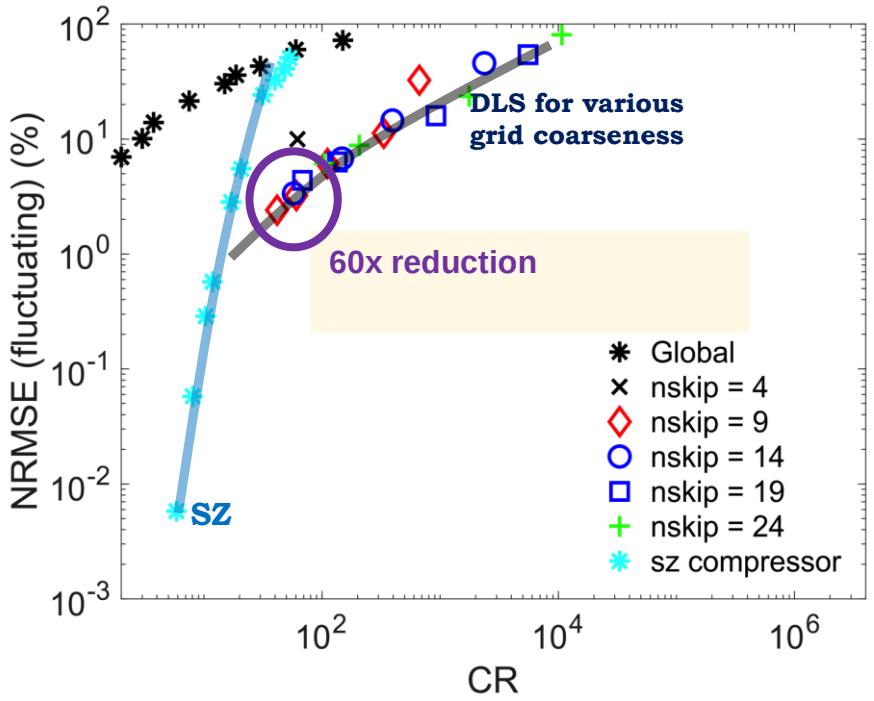




- BitGrooming maxes out at 4x (theoretical, 2-3x in reality)
- The SZ compressor performs well below 10x compression.
- **Using DLS prediction within SZ we can push this compression to over 100x!**

Compression: Flow past Cylinder

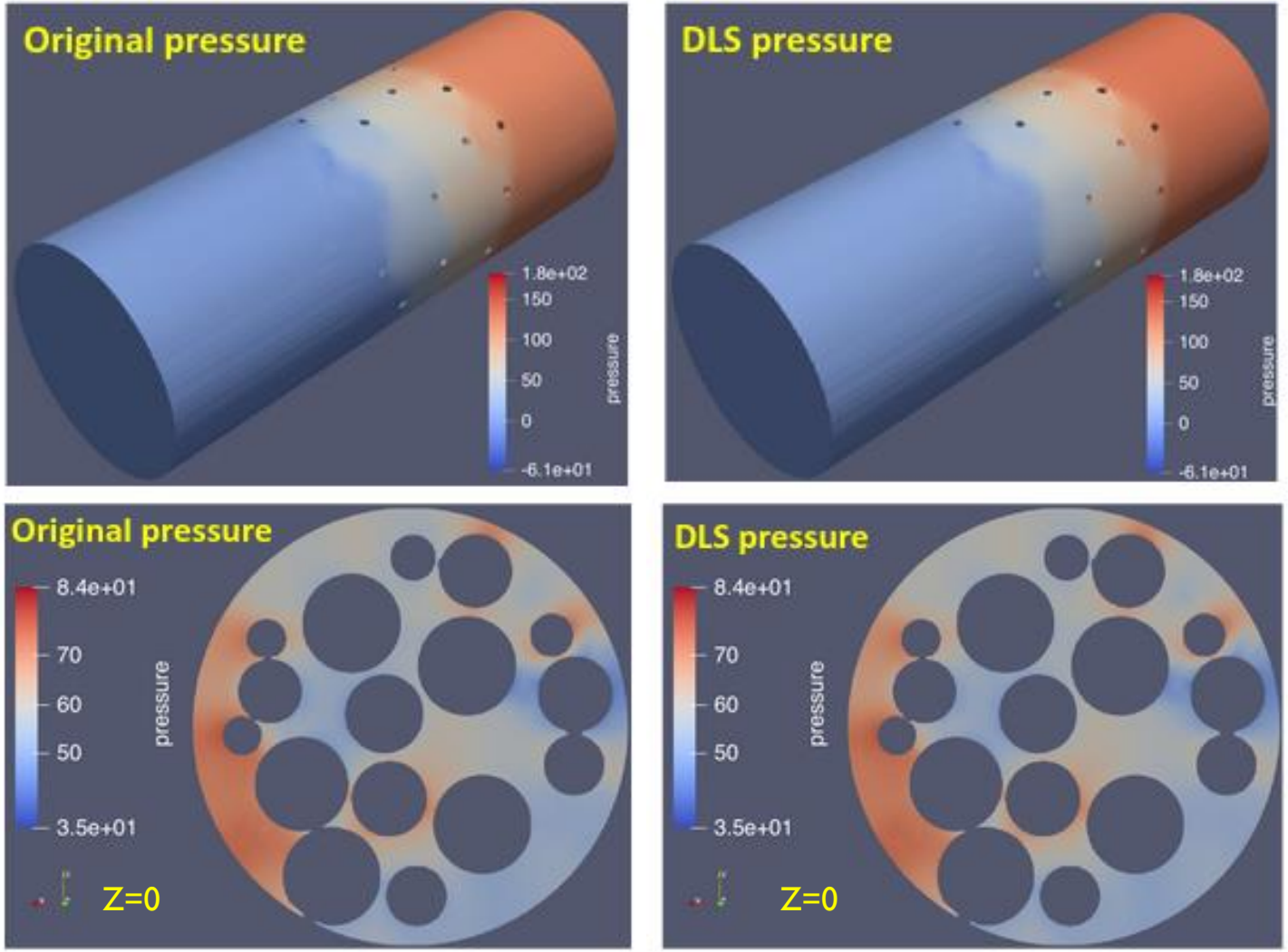




A range of significant data reduction and cost savings can be obtained. The level of savings depends on the user's tolerance for error and analysis type.

RANS Flow Through A Pebble Bed

- Achieved **19x compression**.
- 1.93% NRMSE Over Entire Volume
- 2.40% NRMSE Over the Z=0 slice.

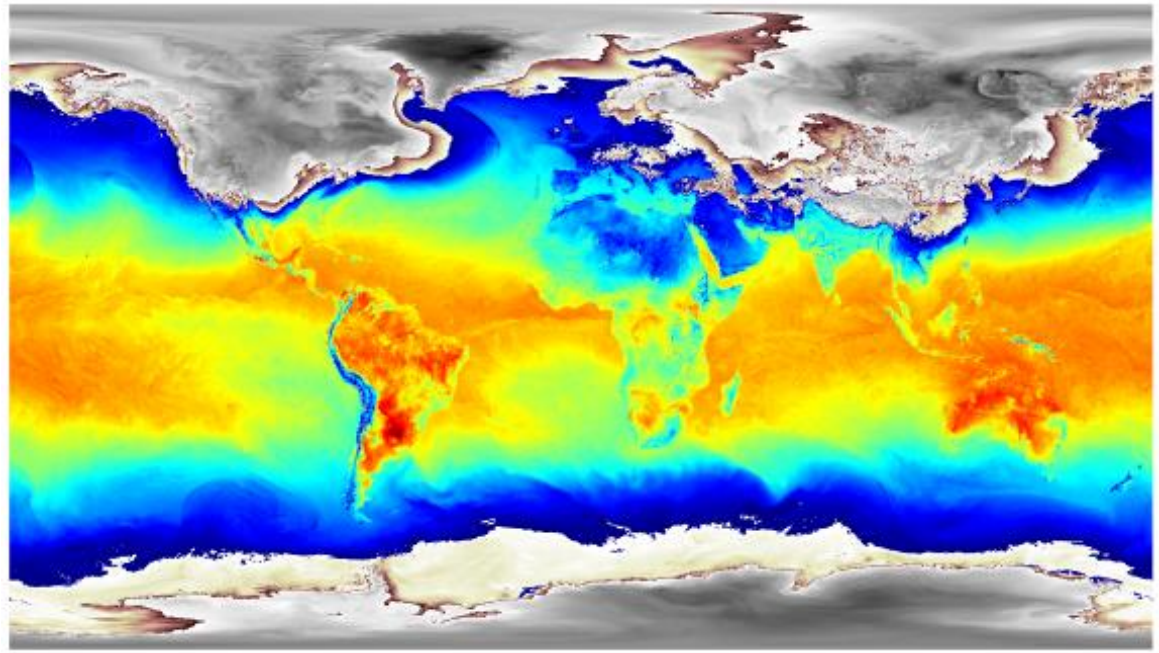


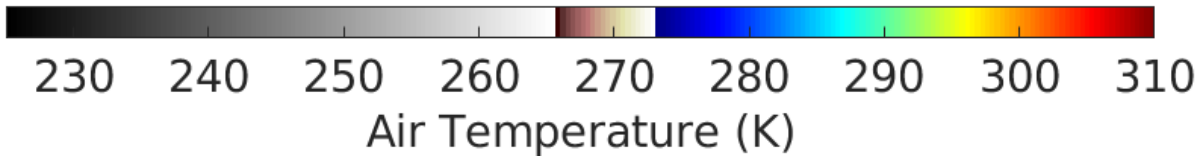
Data courtesy of Dr. Dillon Shaver (Argonne National Lab)

Compression: GEOS-5 NR (Lossy):

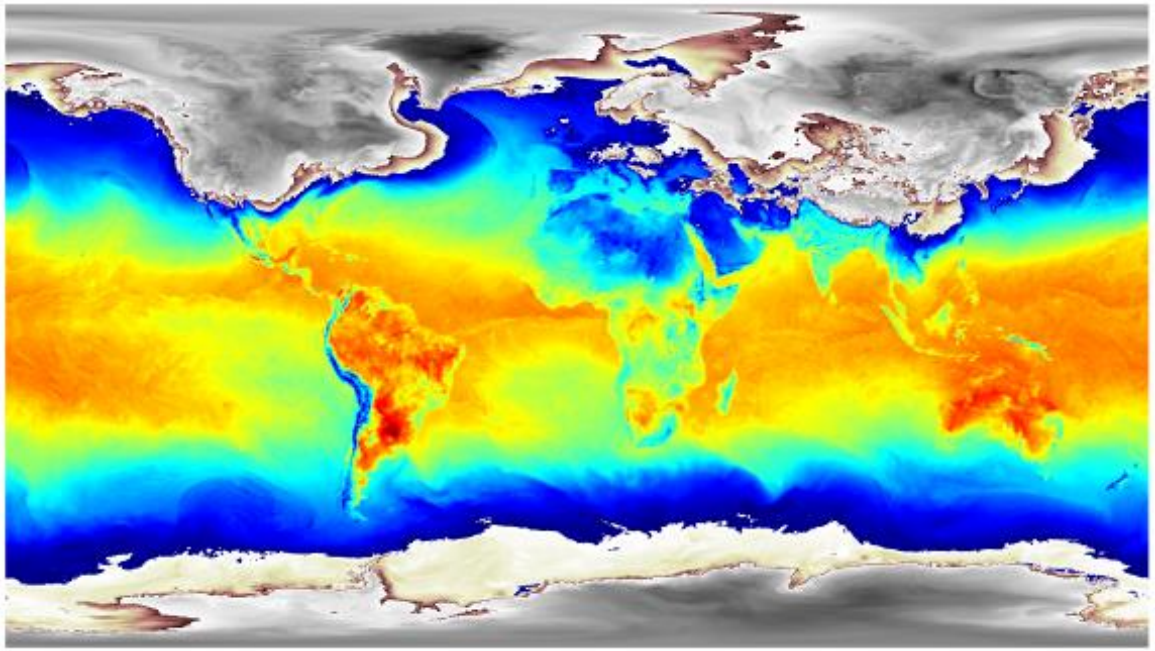
- GEOS-5 Nature Run. Instantaneous 2m Air Temperature **83.97x CR**

Original Solution





Reduced Solution

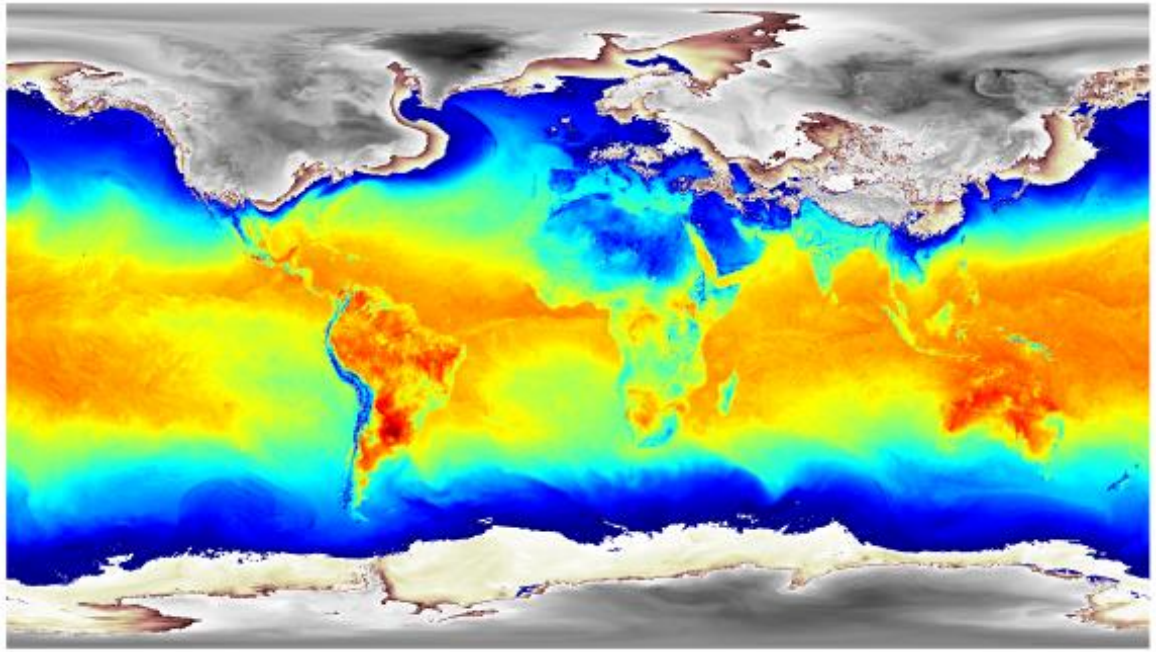


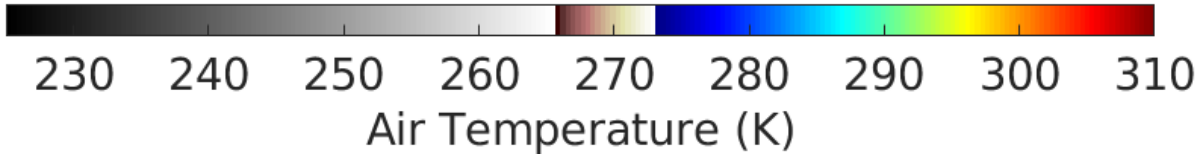
- CR: 83.97x over original size
- NRMSE: 0.2234 %
- Spatial mean temperature diff : -1.56 e-5 K

Compression: GEOS-5 NR (Lossy):

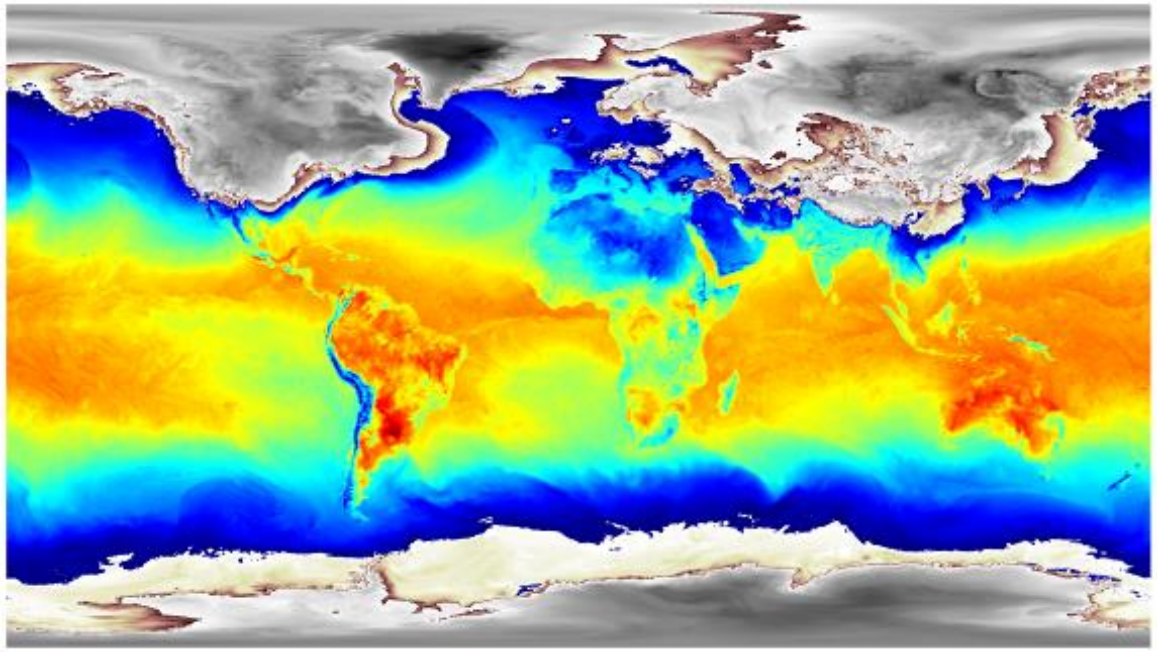
- GEOS-5 Nature Run. Instantaneous 2m Air Temperature **55.98x CR**

Original Solution





Reduced Solution

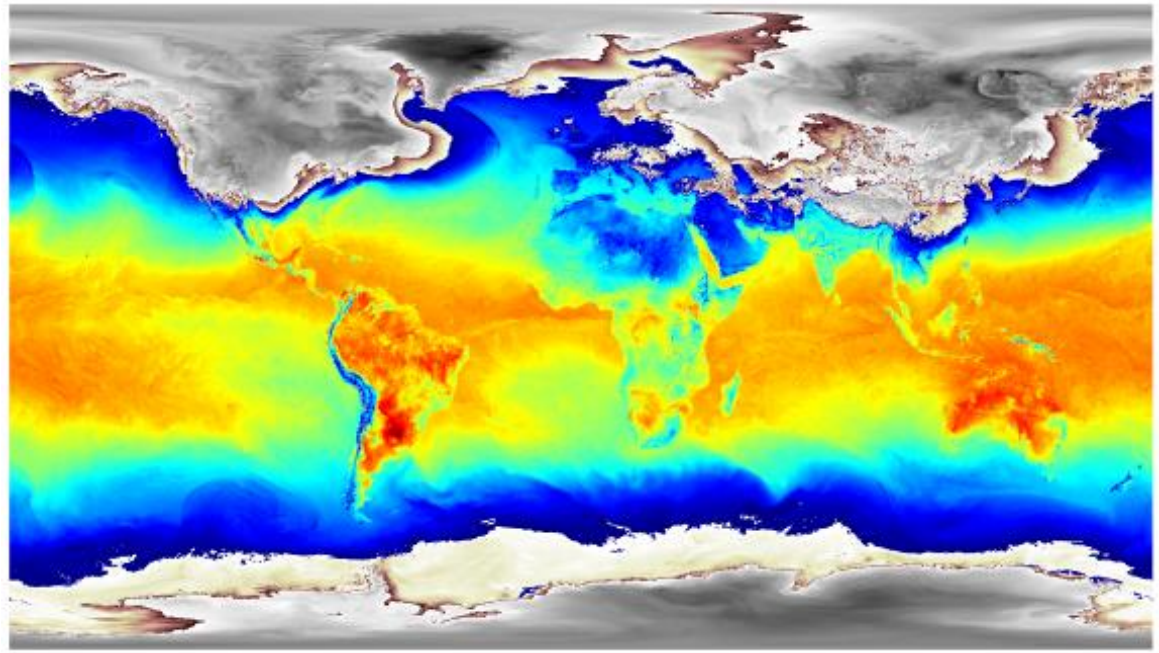


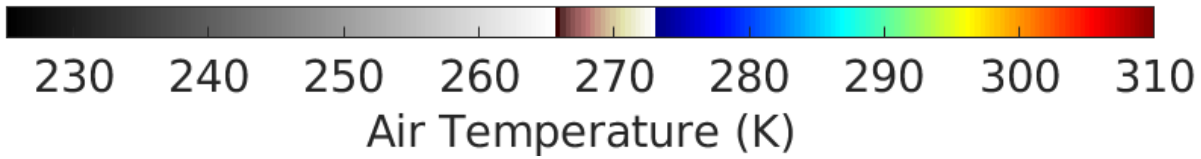
- CR: 55.98x over original size
- NRMSE: 0.1824 %
- Spatial mean temperature diff : -1.31 e-5 K

Compression: GEOS-5 NR (Lossy):

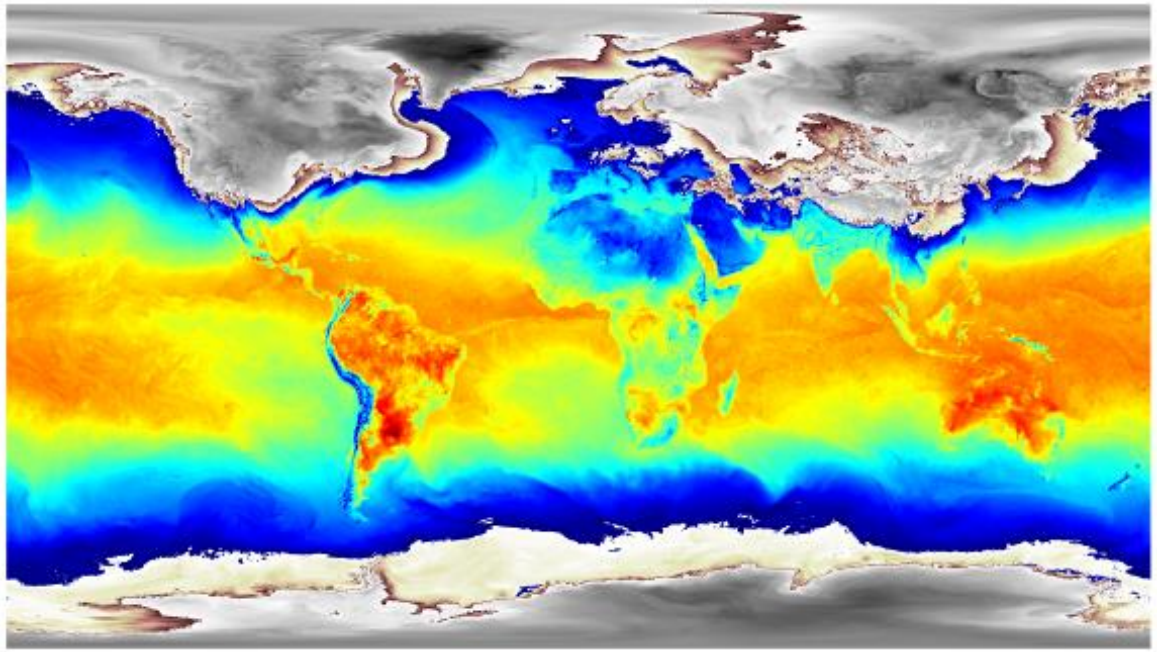
- GEOS-5 Nature Run. Instantaneous 2m Air Temperature **33.59x CR**

Original Solution





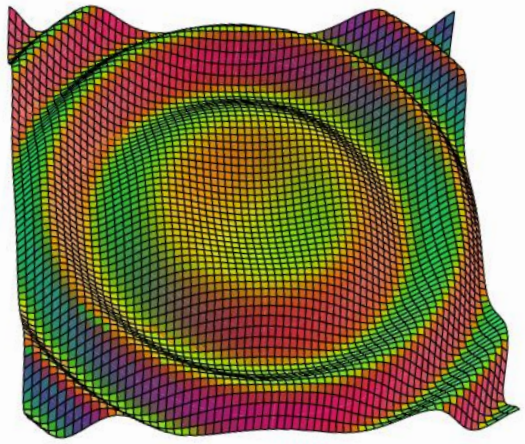
Reduced Solution



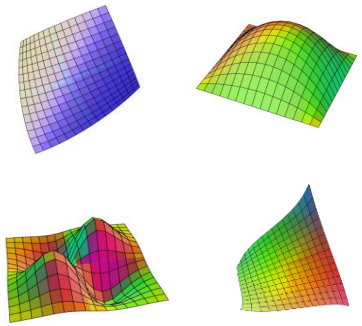
- CR: 33.59x over original size
- NRMSE: 0.1385 %
- Spatial mean temperature diff : -3.81 e-6 K

Accelerated Analytics Over Compressed Data

Rapid Analytics

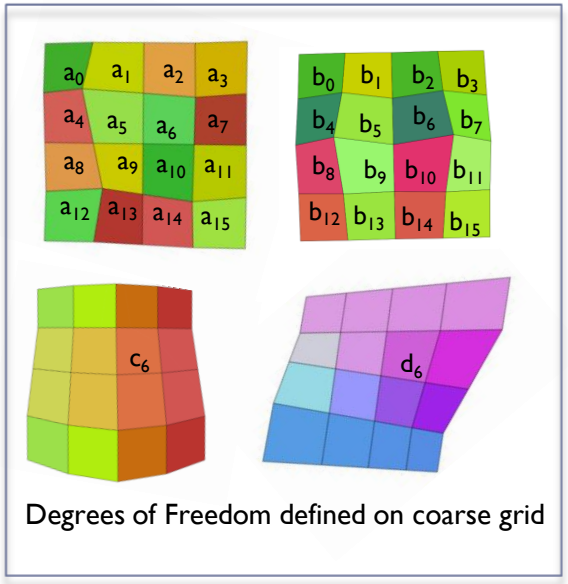
N x 

Compress



Enrichment Functions
(const for all time)

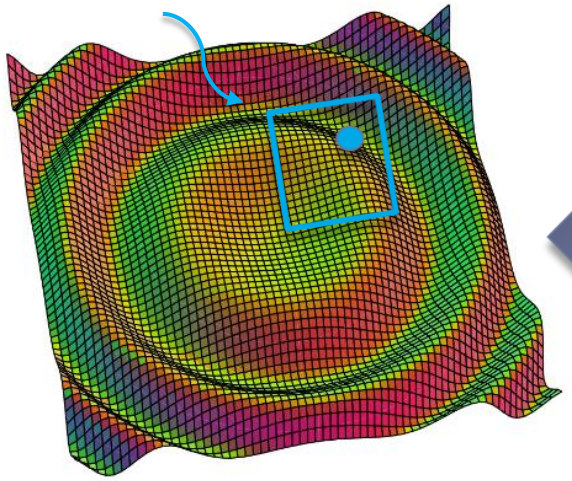
+



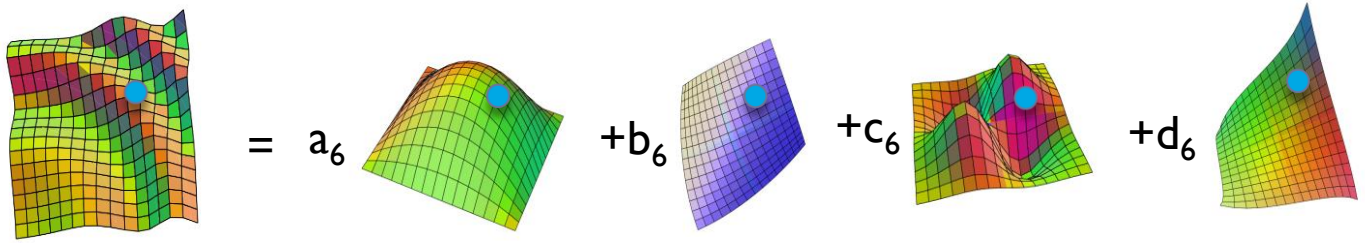
Degrees of Freedom defined on coarse grid

x N

Elm #6

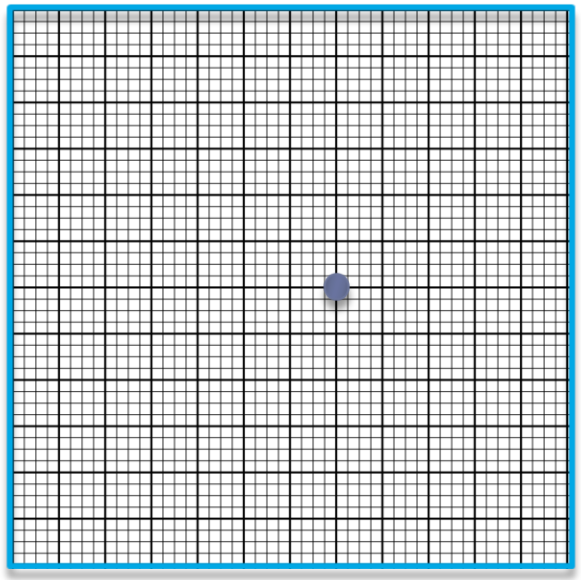


Decompress



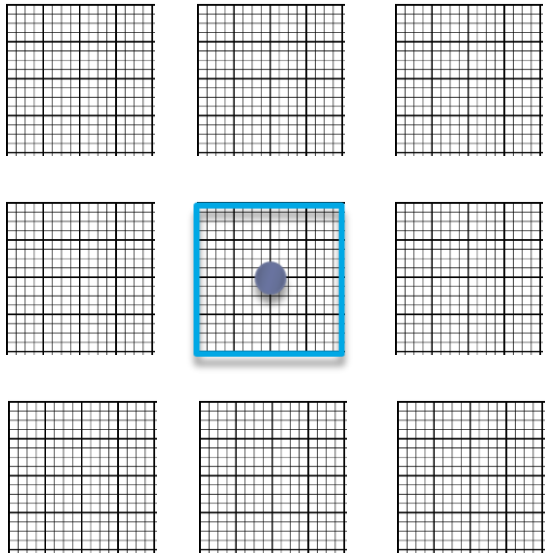
RA 1: Subset Extraction

Basic



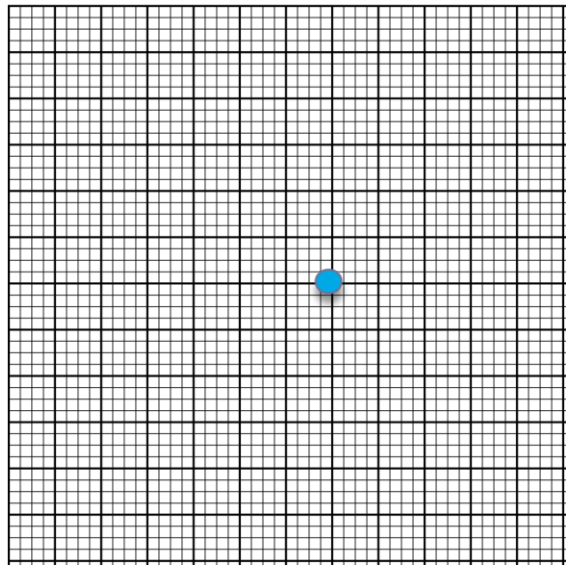
- Decompress the entire solution
- Full dataset must fit in memory
- Not feasible for high fidelity data

HDF5



- Decompress the "chunk"
- Chunk size is defined at compression.
- Chunk cache maximizes performance

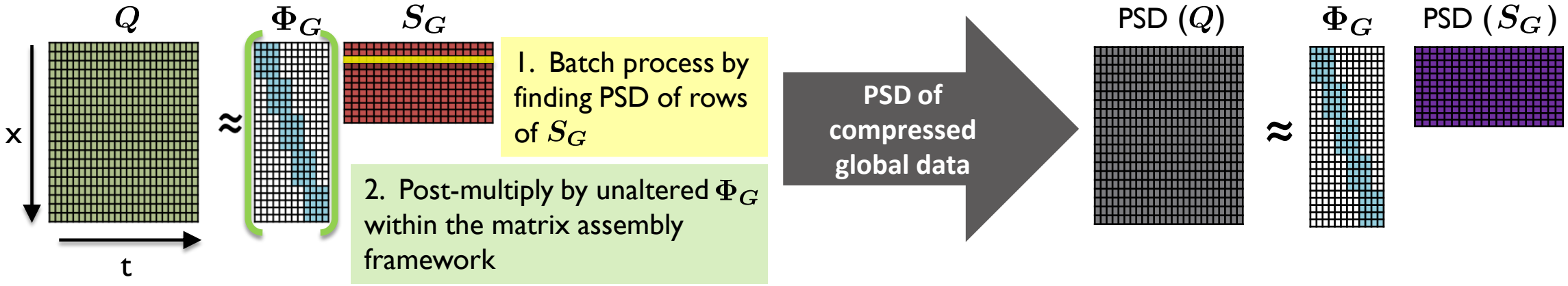
DLS

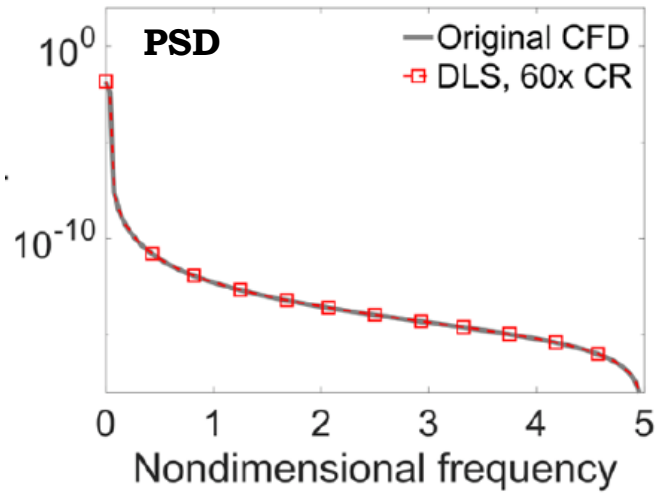


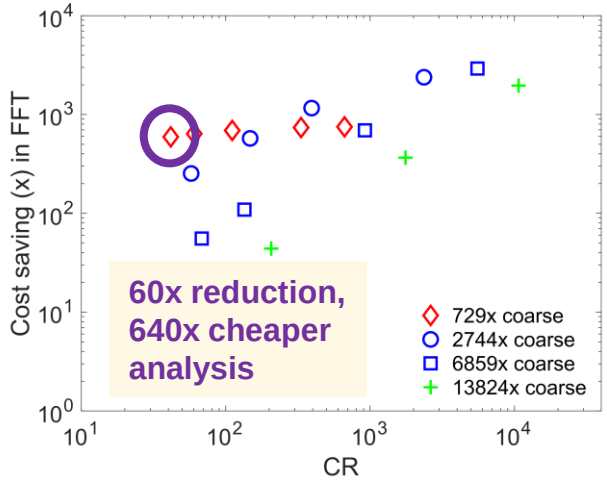
- Decompress the Singular Point
- Low memory footprint
- Can leverage HDF5 for more compression.

RA 2: Leverage Spatial Invariance

- Batch processing temporal component. Example: **Power Spectral Density**



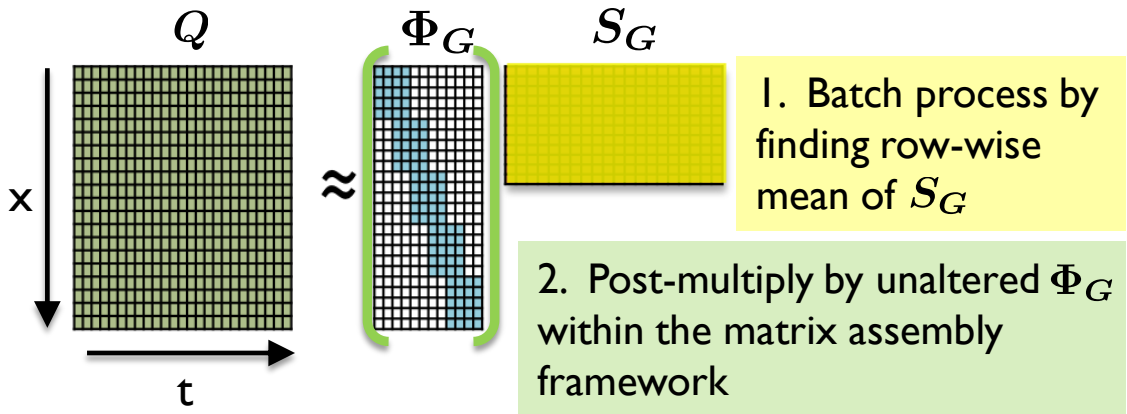




- Analysis on 60x compressed data
- Roughly 640x speedup in PSD calculation over 3000 instances and all the spatial locations

RA 1: Leverage Spatial Invariance

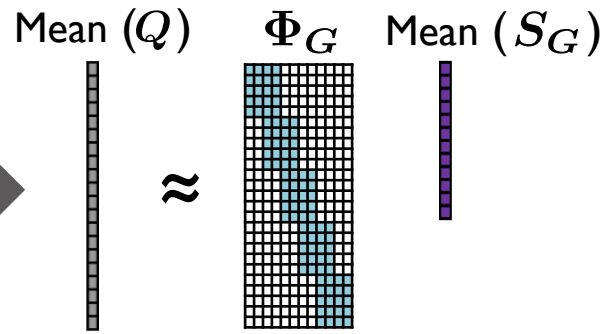
- Batch processing on the temporal component. Example: **Time Average**

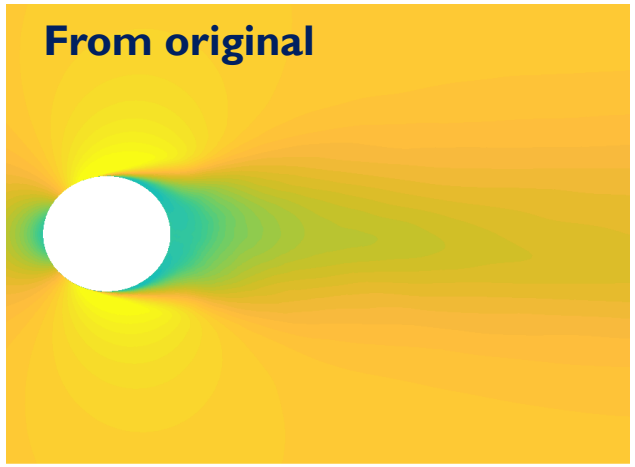


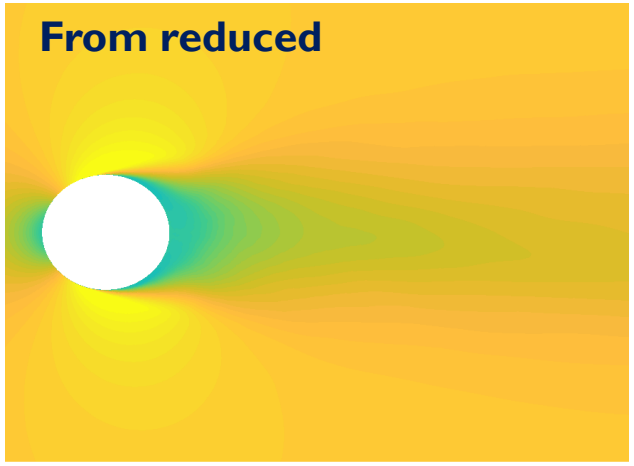
1. Batch process by finding row-wise mean of S_G

2. Post-multiply by unaltered Φ_G within the matrix assembly framework

Time average of compressed data



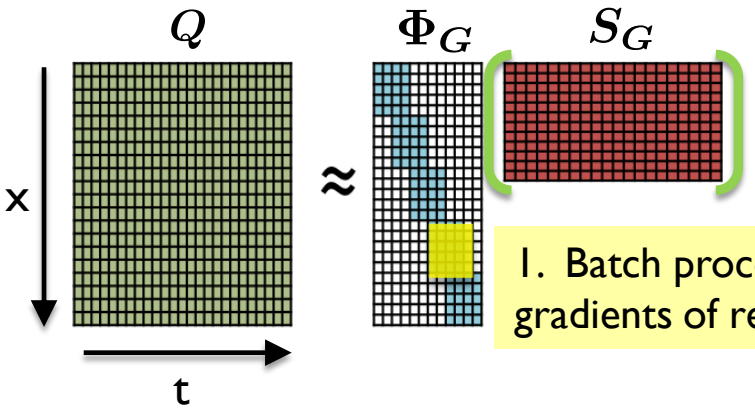




- Analysis on 60x compressed data
- Roughly 25x speedup in time-average calculation over 3000 instances
- Less than 0.1% NRMSE

RA 2: Leverage Temporal Invariance

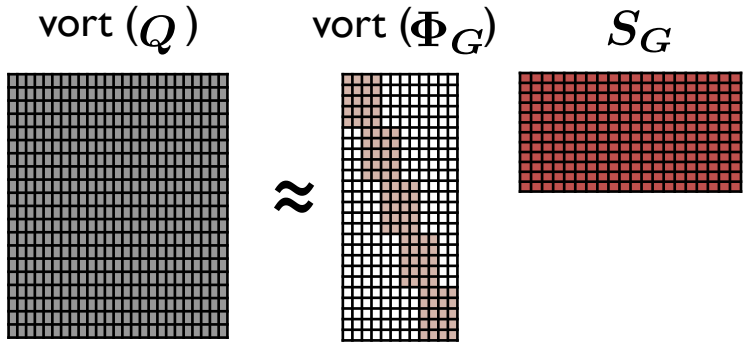
- Batch processing on the spatial component

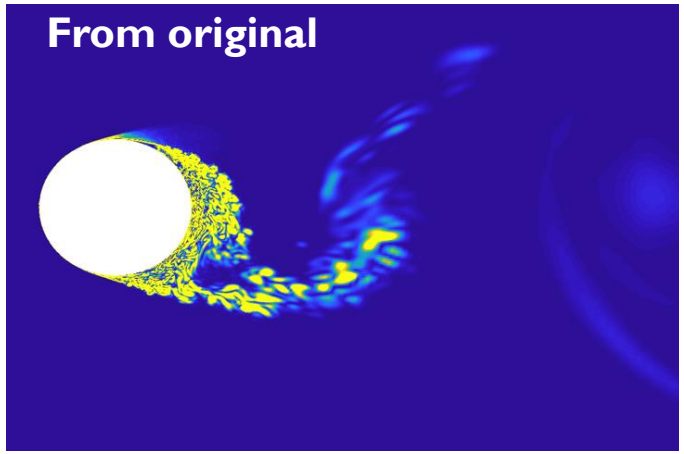


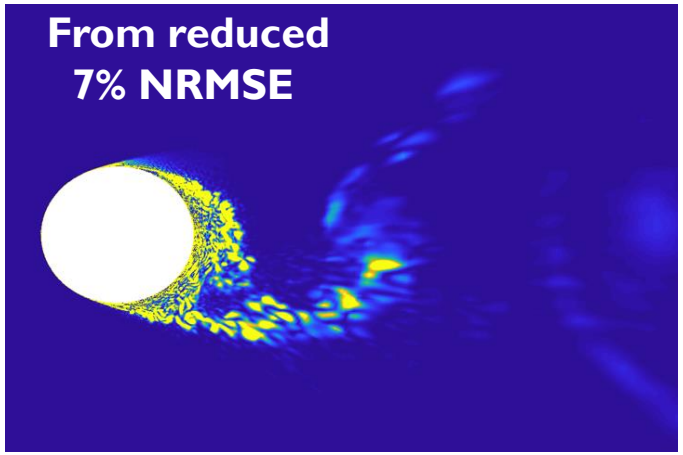
2. Post-multiply by unaltered S_G

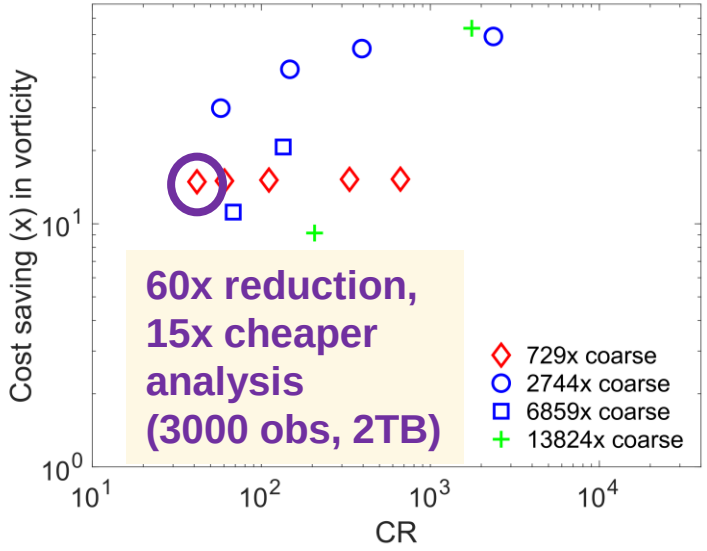
1. Batch process by finding gradients of repeating Φ_L block

Vorticity of compressed time-history data



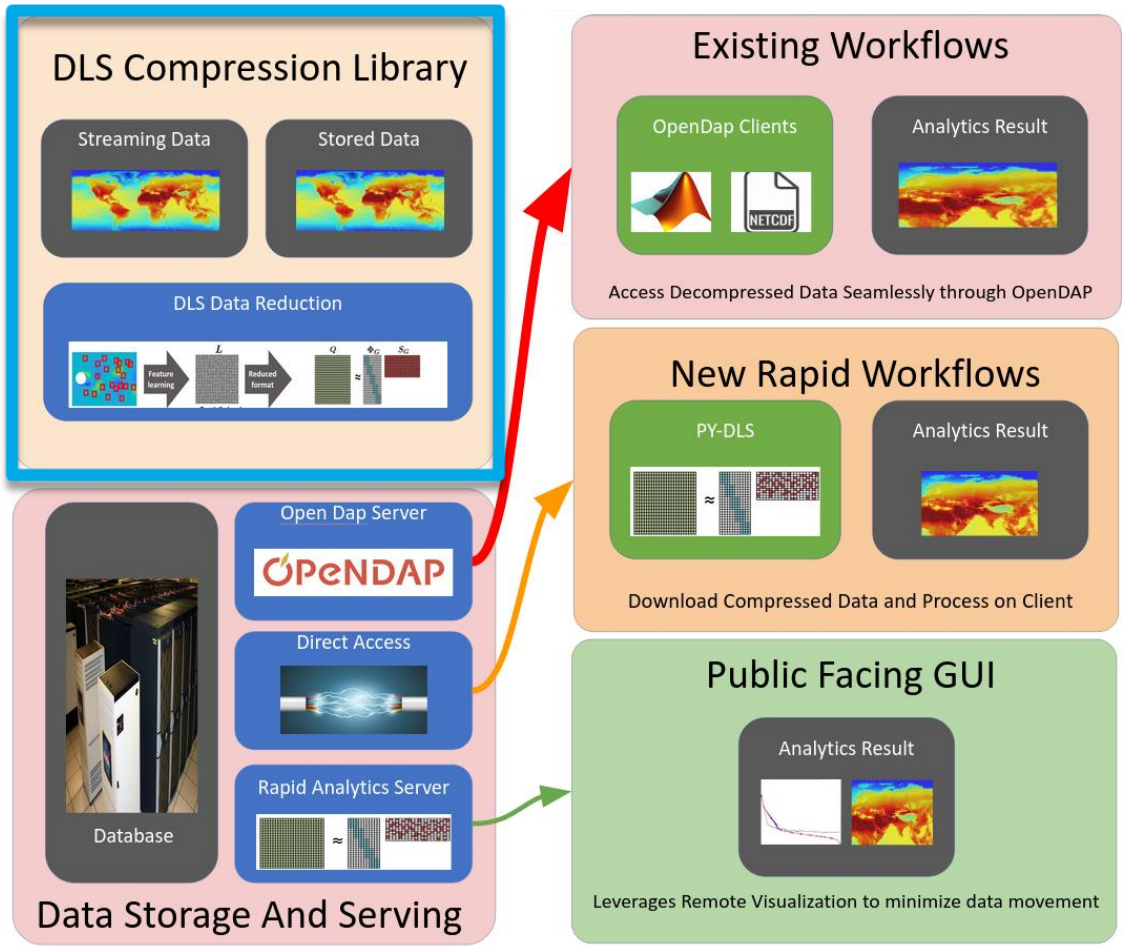






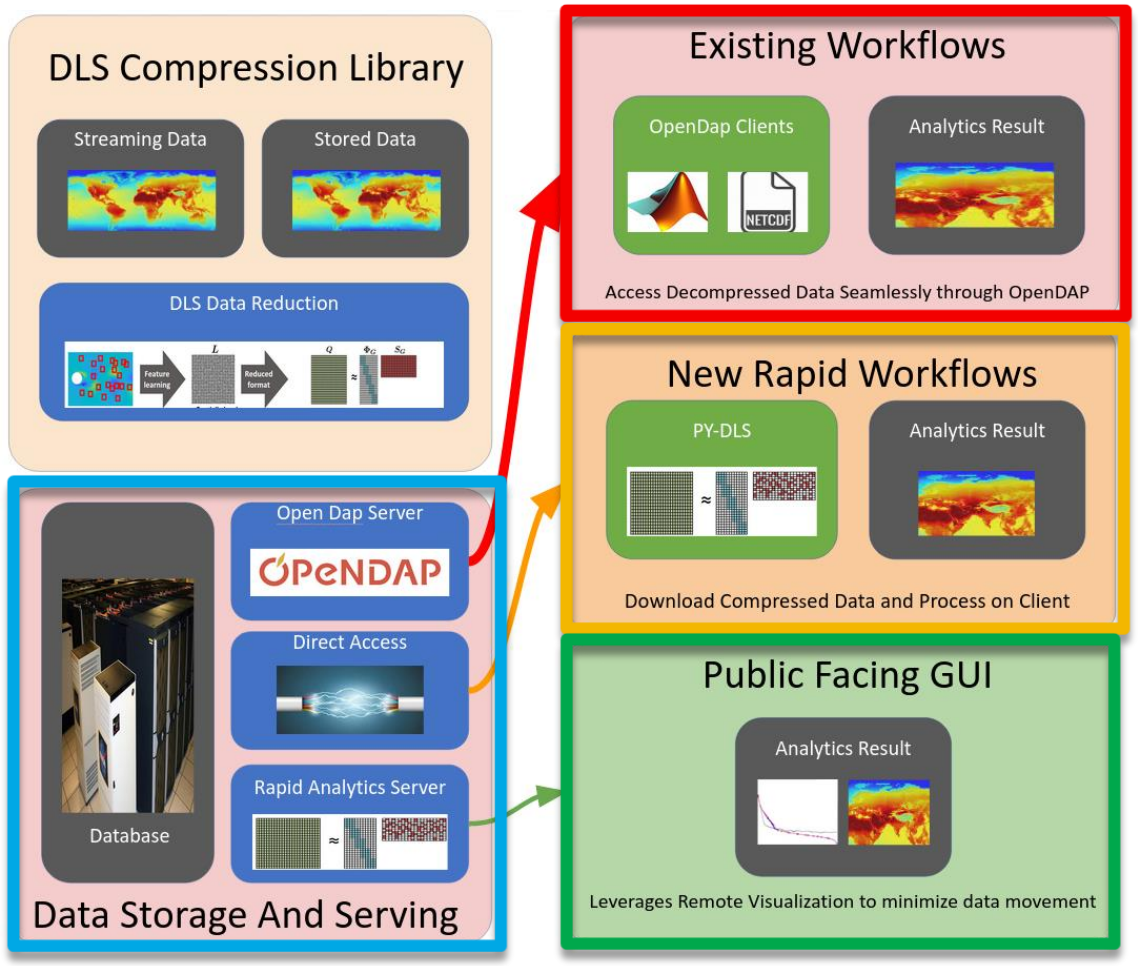
Product:
A Platform for Rapid Data Analytics

- DLS Data Compression Library
  - Open-source C++ Library + CL Interface.
  - Uses MPI for Spatial Domain Decomposition
  - Supports quadratic structured grids in 1D, 2D and 3D
  - Offline Data Compression
    - Supports Plot3D, NetCDF + more
  - Online Data Compression
    - Leverage simulations Domain Decomposition
    - Optional "fast" compression approach available
  - Coming Soon!
    - Unstructured and tetrahedral grids
    - Adaptive Enrichment.



Product:
A Platform for Rapid Data Analytics

- Data Storage and Server
  - OpenDap Client/Server:
    - Seamless integration into existing NetCDF workflows.
  - Direct Access through Python API
    - C,C++ interface with Python bindings.
    - Supports "Serverless" data deployment (e.g. S3)
    - Direct access to the enrichment functions and DOFS
    - Built in collection of rapid analytics algorithms.
  - Public Facing GUI
    - Intuitive graphical interface for cloud based data analytics
    - Leverages Plotly and Paraview for advanced visualization
    - Allows everyday users to quickly analyze complex data.



SBIR Project Information

- **Project Title:** Rapid Data Analytics Using Machine Learning
- **SBIR Topic:** NASA 2023, Topic S17 (01), Technologies for Large Scale Numerical Simulation (Exascale Computing)
- **Contract Number:** 80NSSC23PB560
- **Prime:** RNET Technologies, Inc.
  - PI: Ben O'Neill (BONeill@RNET-Tech.com)
  - Contract: V. "Nagu" Nagarajan (VNagarajan@RNET-Tech.com)
- **Subcontractor: Florida Atlantic University**
  - PI: Rohit Deshmukh (rdeshmukh@fau.edu)
- **NASA Technical Monitor**
  - Sharad Gavali