Infrastructure for Topological (and Statistical) Analysis of Extreme-Scale Data

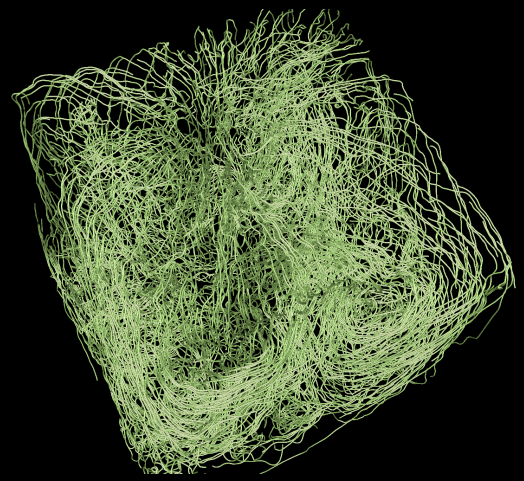
Tom Peterka

tpeterka@mcs.anl.gov

Mathematics and Computer Science Division

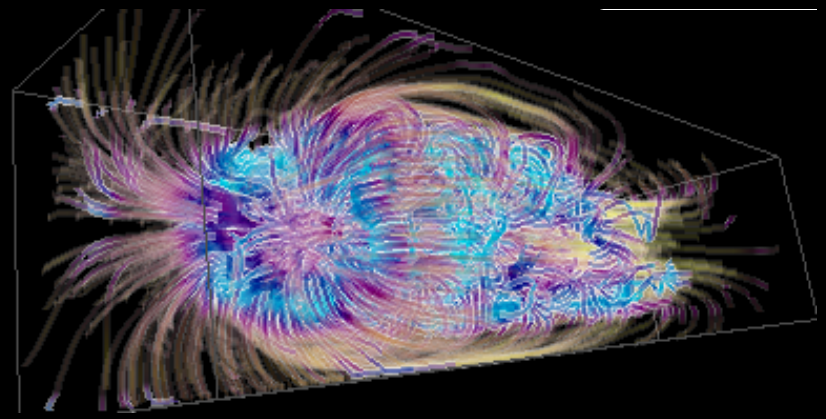
Workshop on Large Scale Data Analysis Using Topology and Statistics 6/14/13

Data Analysis Comes in Many Flavors



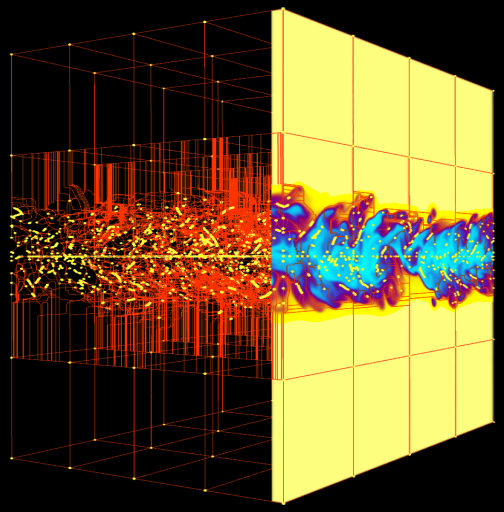
Visual

Particle tracing of thermal hydraulics flow



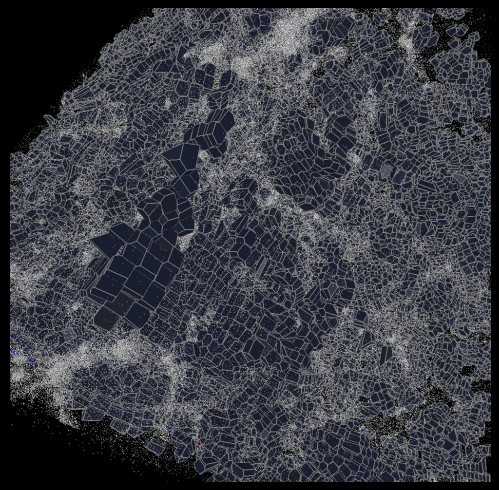
Statistical

Information entropy analysis of astrophysics



Topological

Morse-Smale complex of combustion



Geometric

Voronoi tessellation of cosmology

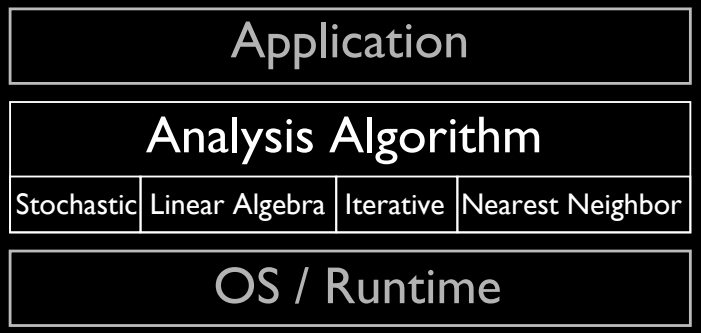
Fortunately, Their Data Movement Patterns Do Not

	Analysis	Communication	
Regular	Sort-Last Rendering	Swap-Based Reduction	Homogeneous Data
	Morse-Smale Complex	Merge-Based Reduction	Heterogeneous Data
	Information Entropy	Merge-Based Reduction	
Semi Regular	Particle Tracing	Nearest Neighbor	
	Voronoi Tessellation	Nearest Neighbor	
	Region Growing	Nearest Neighbor	
Irregular	Graph layout	Send-Receive	

Many different analysis operations share a small set of communication patterns. These communication kernels together with supporting utilities for decomposition and I/O can be encapsulated, optimized, and reused.

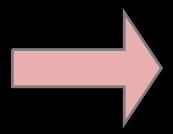
How to Parallelize Data Analysis: 2 Ways

By hand

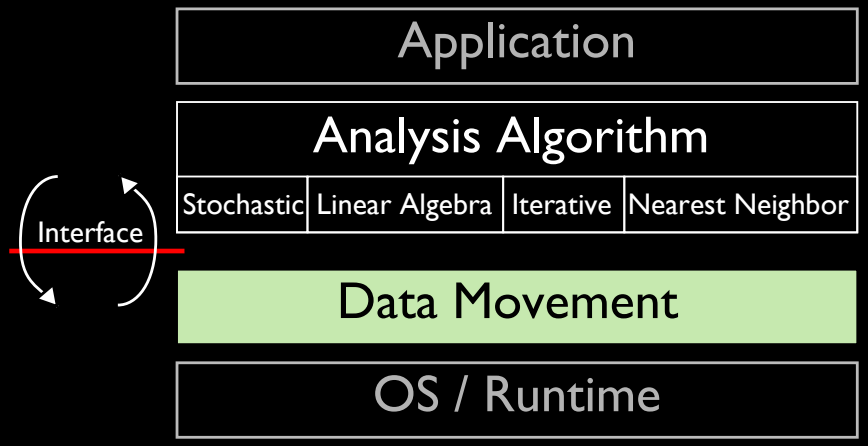



```
void ParallelAlgorithm() {
   …
   MPI_Send();
   …
   MPI_Recv();
   …
   MPI_Barrier();
   …
   MPI_File_write();
}
```

or



With tools




```
void ParallelAlgorithm() {
   …
   LocalAlgorithm();
   …
   DIY_Merge_blocks();
   …
   DIY_File_write()
}
```

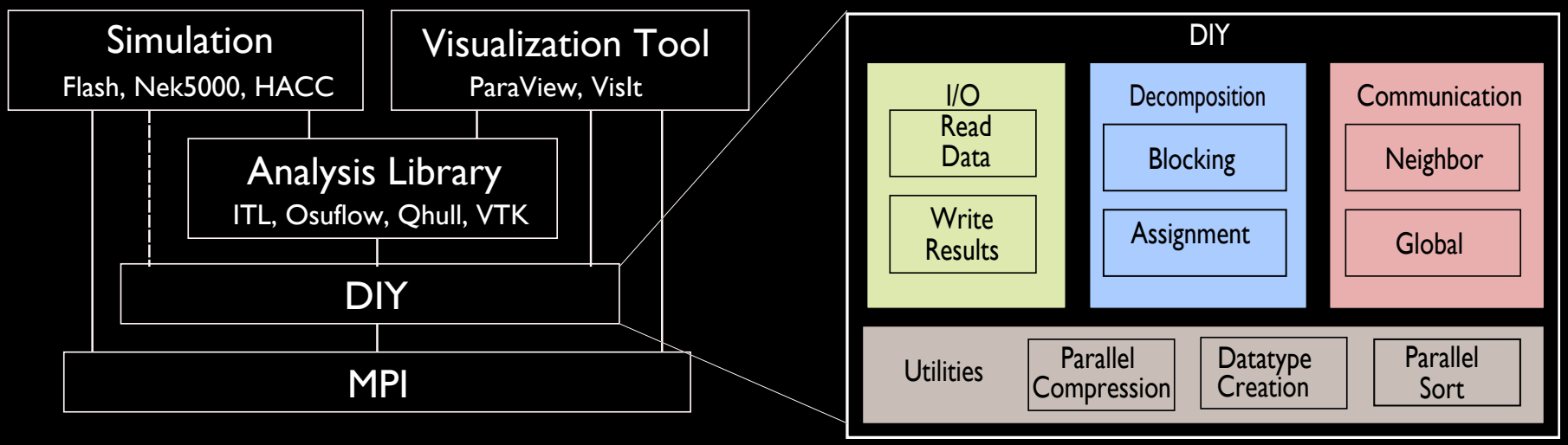
DIY is a Library for Option #2.

Features

- Parallel I/O to/from storage
- Domain decomposition
- Network communication
- Utilities

Library

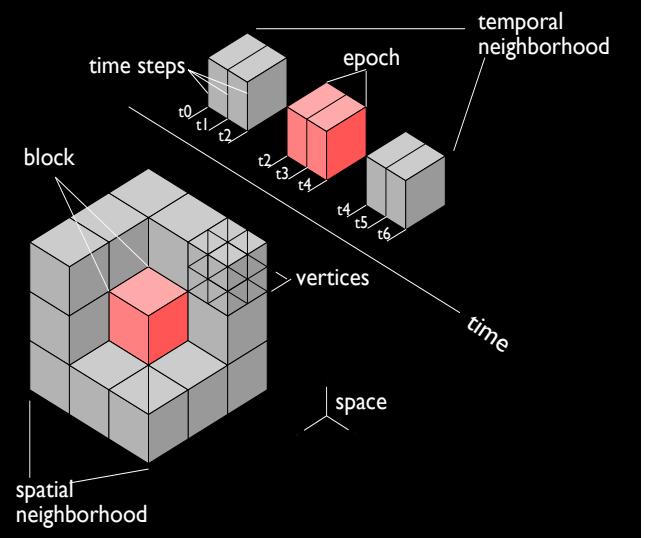
- Written in C++ with C bindings
- Autoconf build system (configure, make, make install)
- Lightweight: libdiy.a 800KB
- Maintainable: ~15K lines of code, including examples

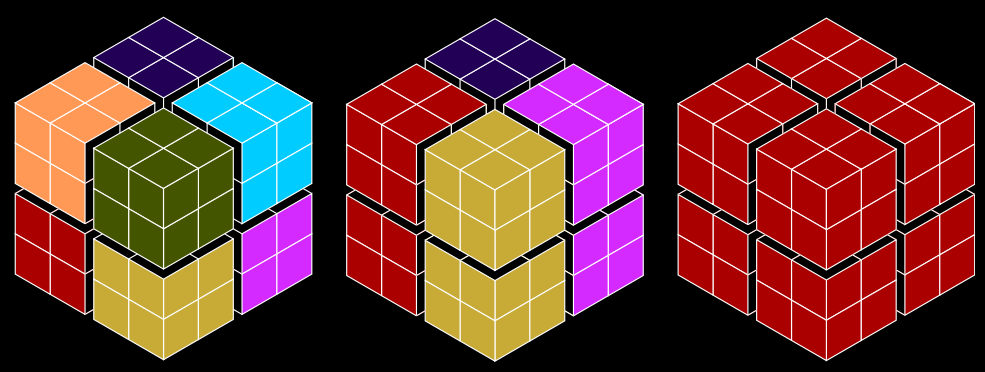


DIY usage and library organization

Nine Things That DIY Does

1. Separate analysis ops from data ops
2. Group data items into blocks
3. Assign blocks to processes
4. Group blocks into neighborhoods
5. Support multiple multiple instances of 2, 3, and 4
6. Handle time
7. Communicate between blocks in various ways
8. Read data and write results
9. Integrate with other libraries and tools

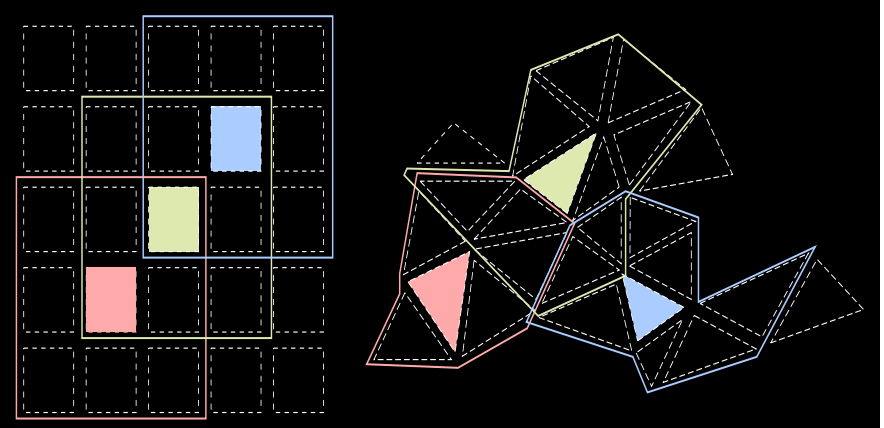




8 processes

4 processes

1 process



Two examples of 3 out of a total of 25 neighborhoods

Parallel Information-Theoretic Analysis

Collaboration with the Ohio State University and New York University Polytechnic Institute

Objective

-Decide what data are the most essential for analysis

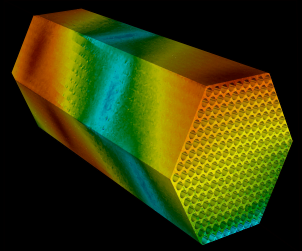
-Minimize the information losses and maximize the quality of analysis

-Steer the analysis of data based on information saliency

Information-theoretic approach

-Quantify Information content based on Shannon's entropy

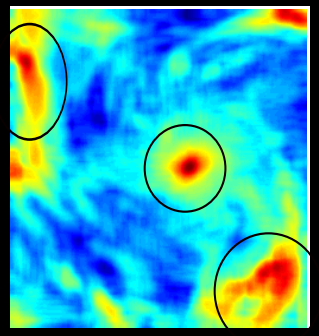
-Use this model to design new analysis data structures and algorithms



Nek5000
CFD model

Information-theoretic algorithms

Areas of high information entropy--turbulent regions in original data--are the interesting regions in simulating coolant flow in a nuclear reactor.



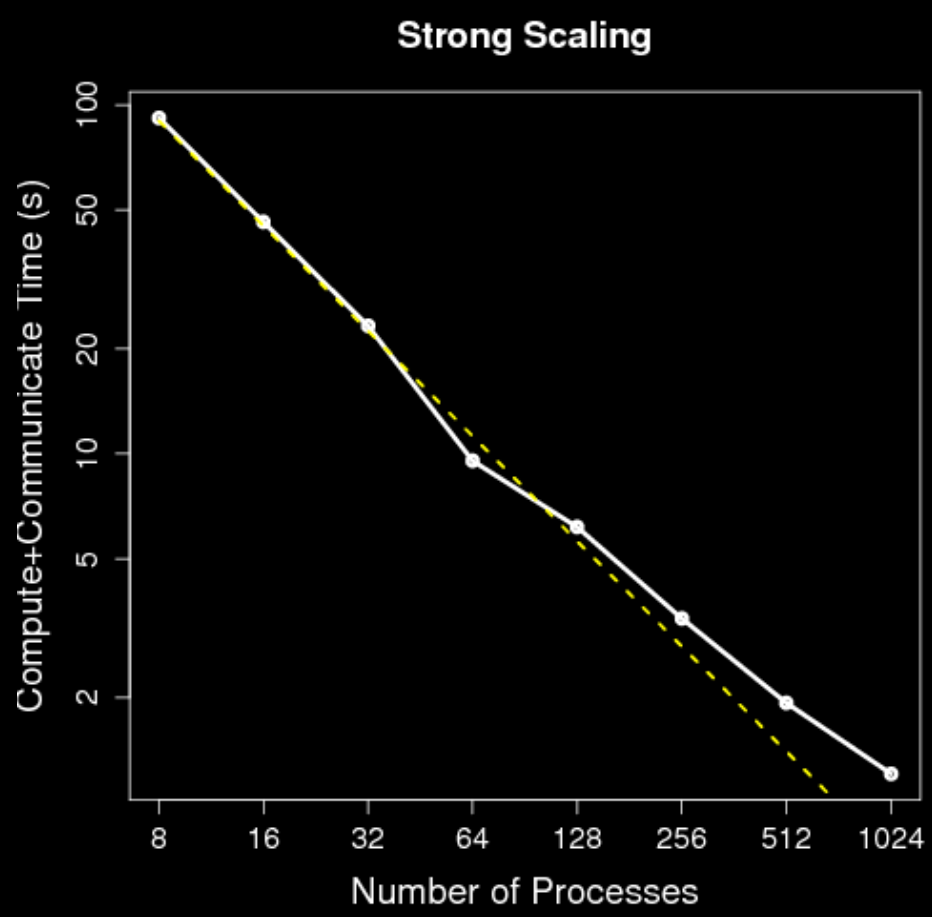
Section of information entropy field

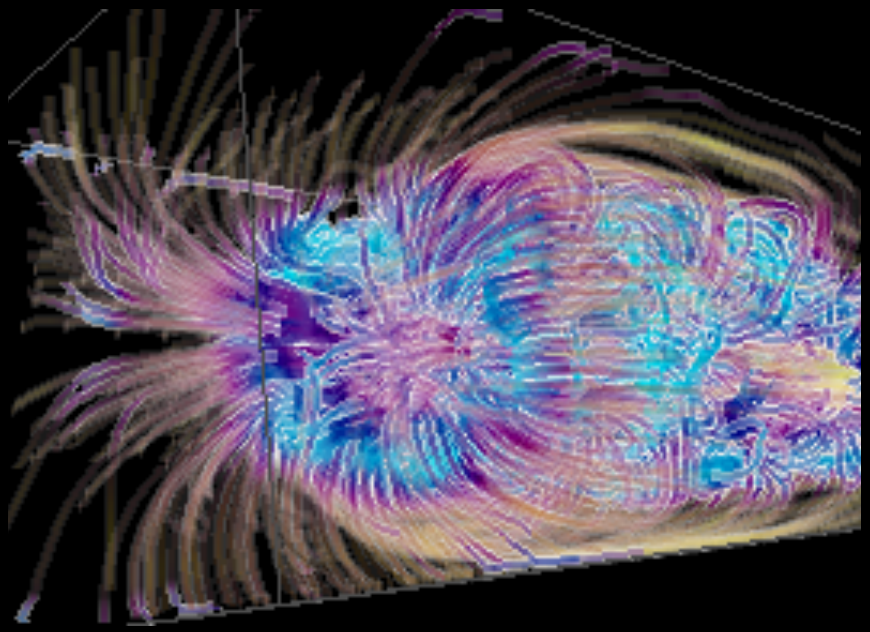
Shannon's Entropy

The average amount of information expressed by the random variable is

$$H(x) = -\sum_{i=1}^{n} p_i \log p_i$$

Information Entropy





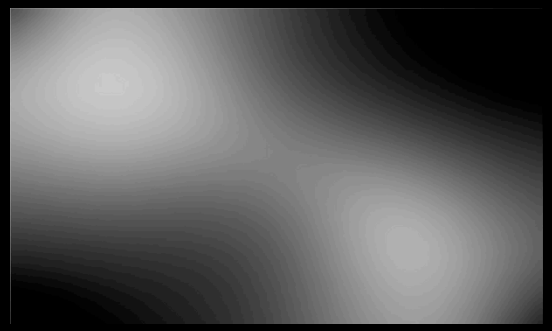
Computation of information entropy in 126x126x512 solar plume dataset shows 59% strong scaling efficiency.

Chaudhuri et al., Scalable Computation of Distributions from Large Scale Data Sets, LDAV '12

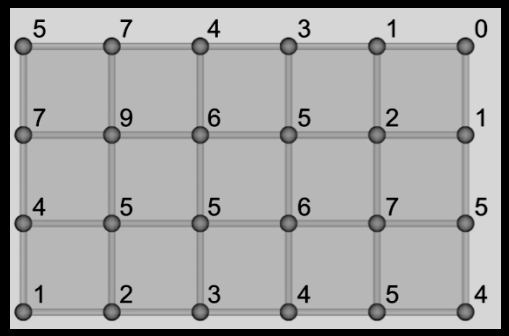
Parallel Topological Analysis

Collaboration with SCI Institute, University of Utah

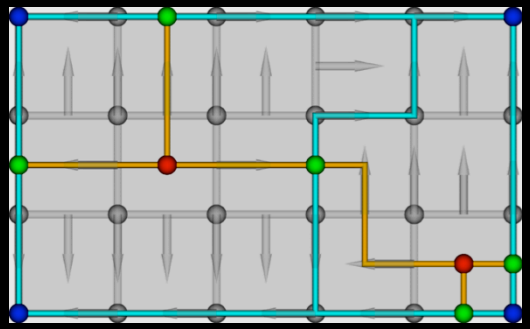
-Transform discrete scalar field into Morse-Smale complex
-Nodes are minima, maxima, saddle points of scalar values
-Arcs represent constant-sign gradient flow
-Used to quickly see topological structure



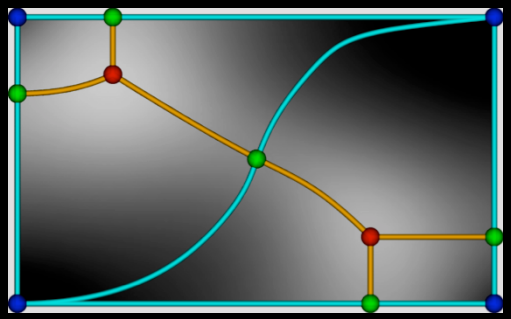
1



2

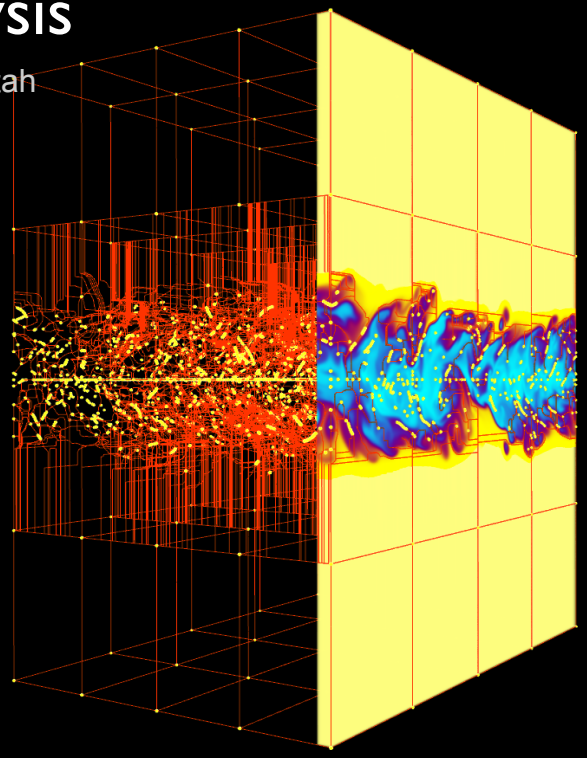


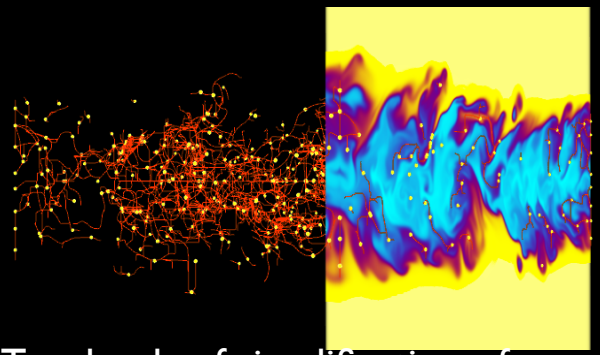
3



4

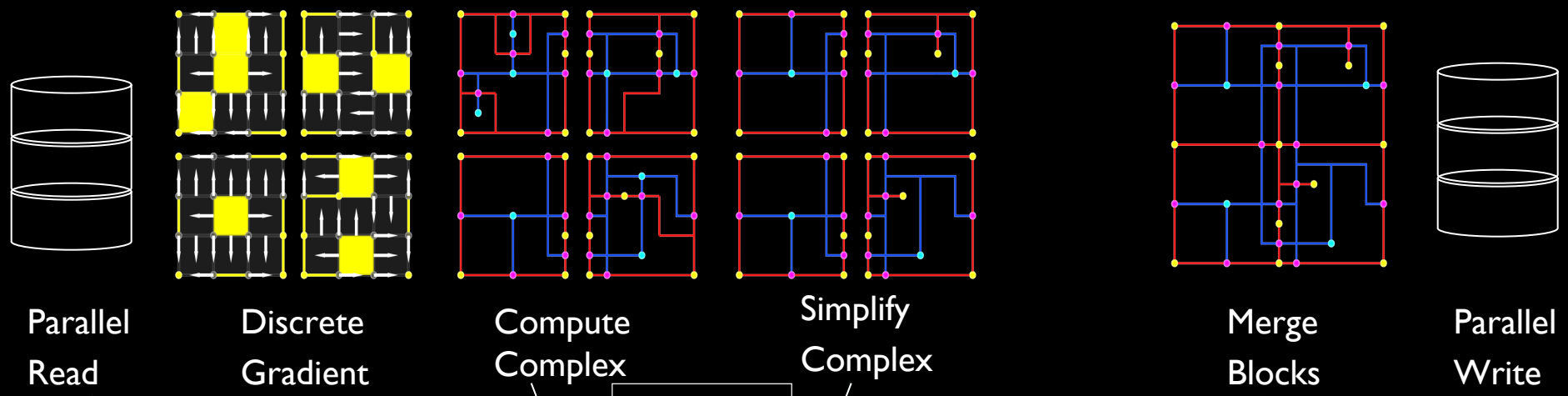
Example of computing discrete gradient and Morse-Smale Complex



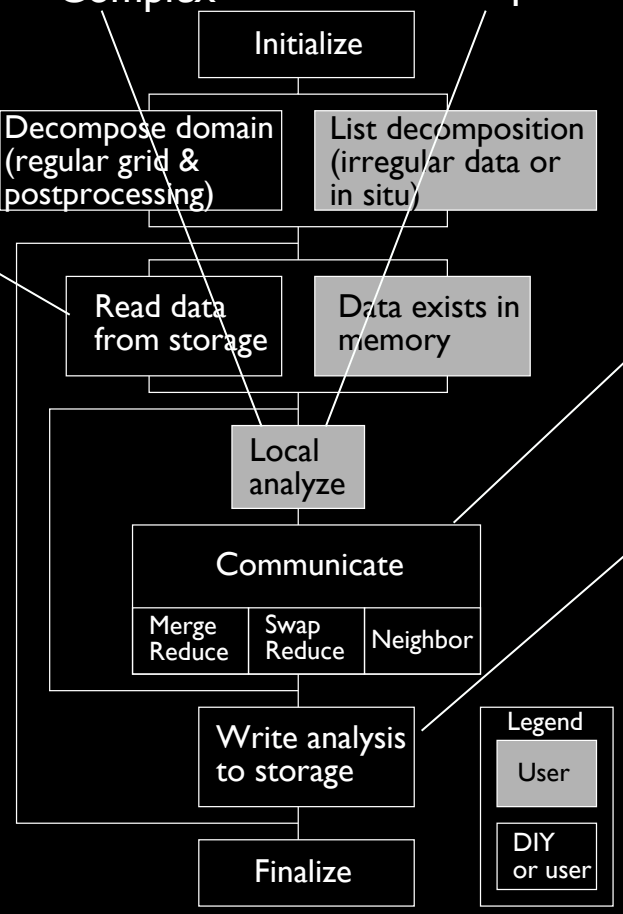


Two levels of simplification of the Morse-Smale complex for jet mixture fraction.

Gyulassy et al., The Parallel Computation of Morse-Smale Complexes, IPDPS '12



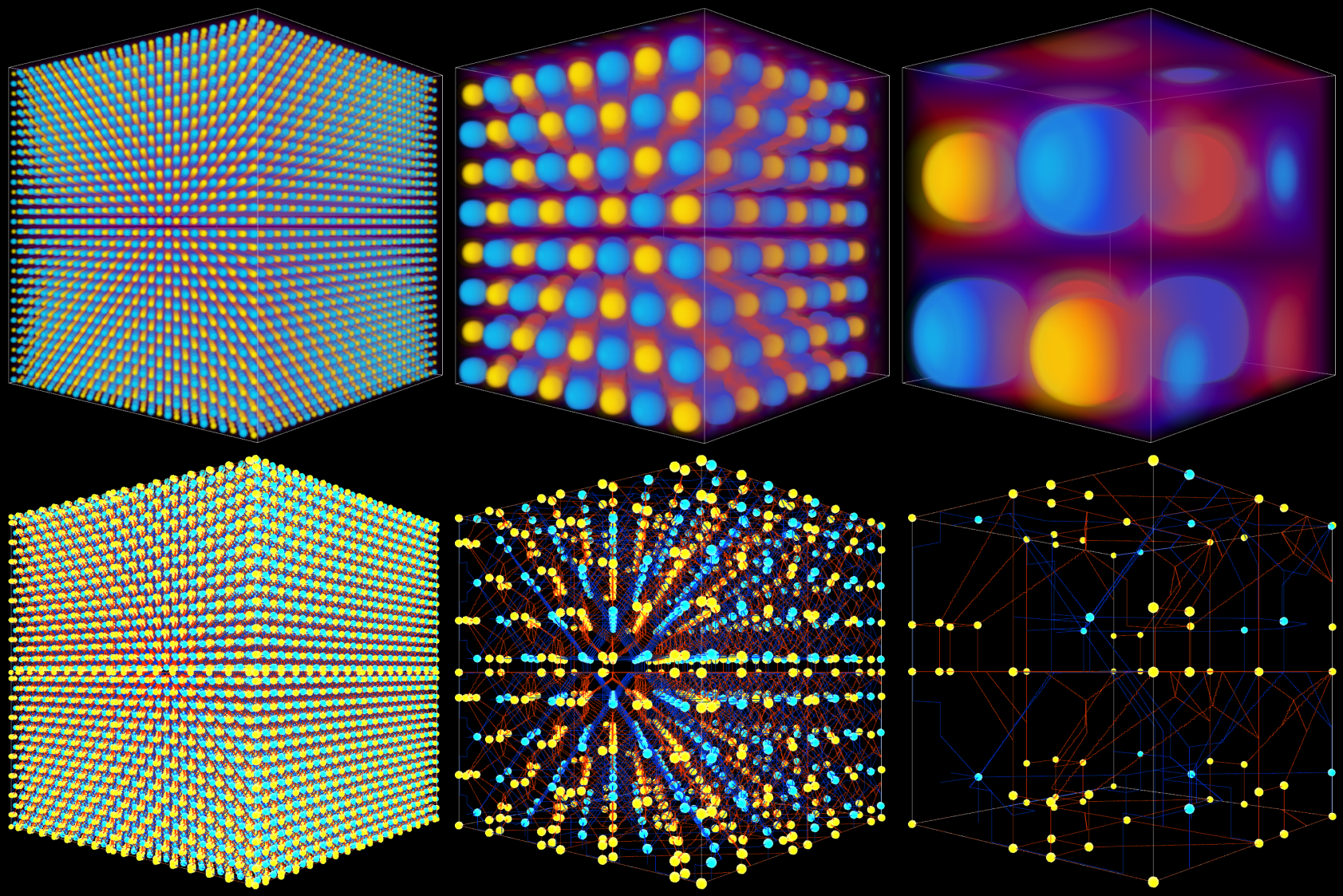
Top: overview of our algorithm, the arrows and circled component indicate the sequence of operations performed by a single process: (a) parallel read, (b) local gradient computation, (c) local MS complex computation, (d) simplification, (e) preparing data structures for communication, (f) merging complexes, (g) parallel write.



Left: Typical flow chart of a DIY parallel analysis program. Lines to upper diagram map these to the specific parallel Morse-Smale Complex construction algorithm.

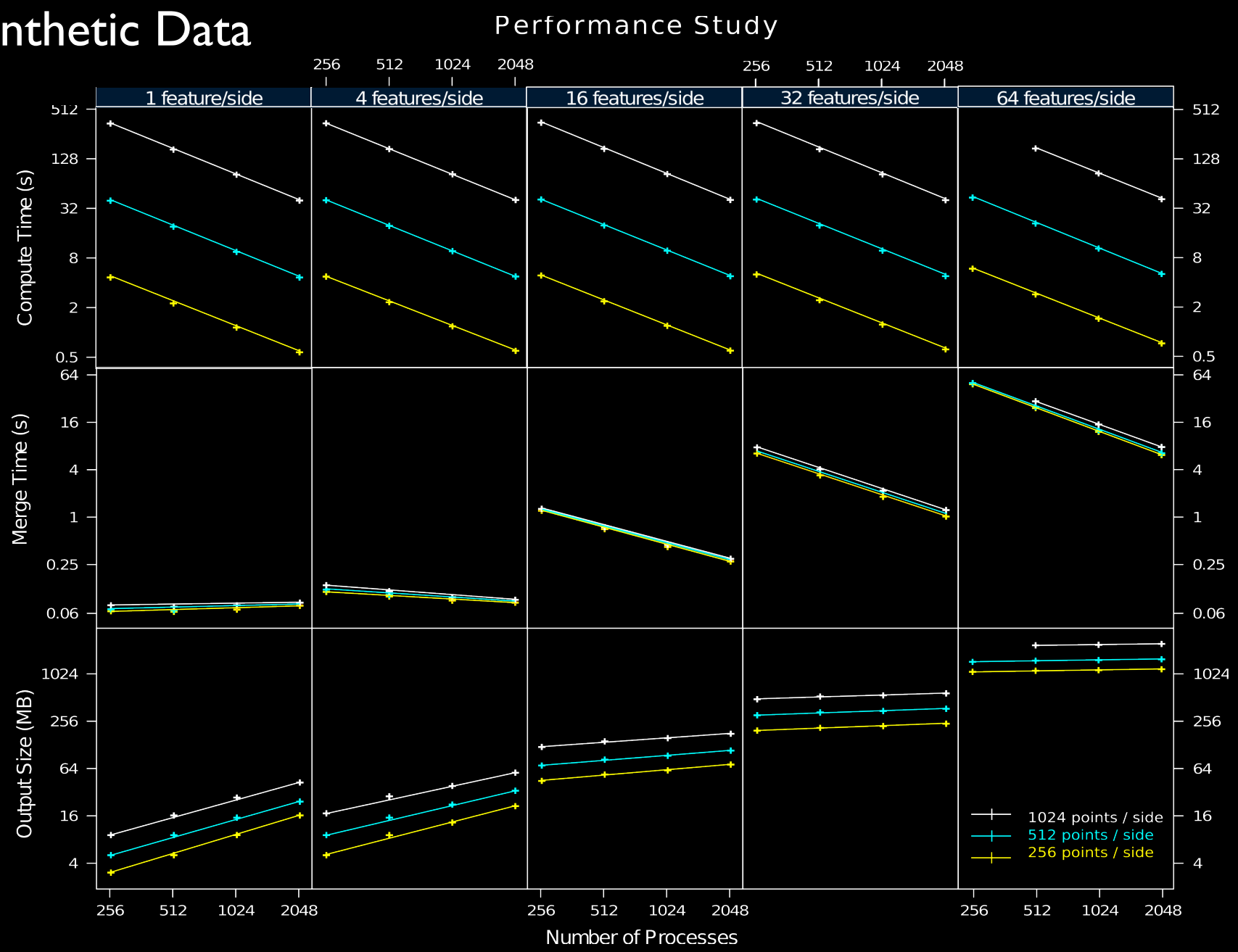
Parallel Approach

Synthetic Data



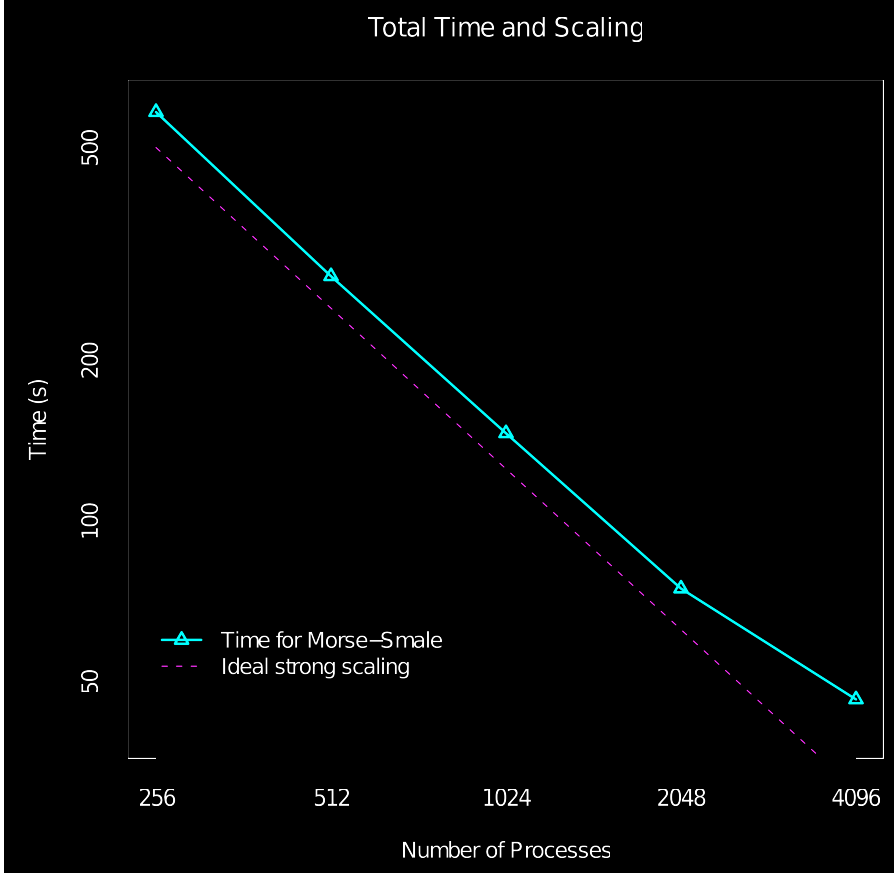
An artificially generated 256^3 dataset is volume rendered (top row), and the corresponding Morse-Smale Complex is illustrated (bottom row) for varying feature counts.

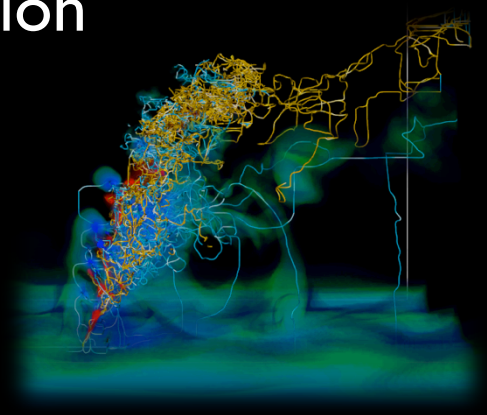
Synthetic Data

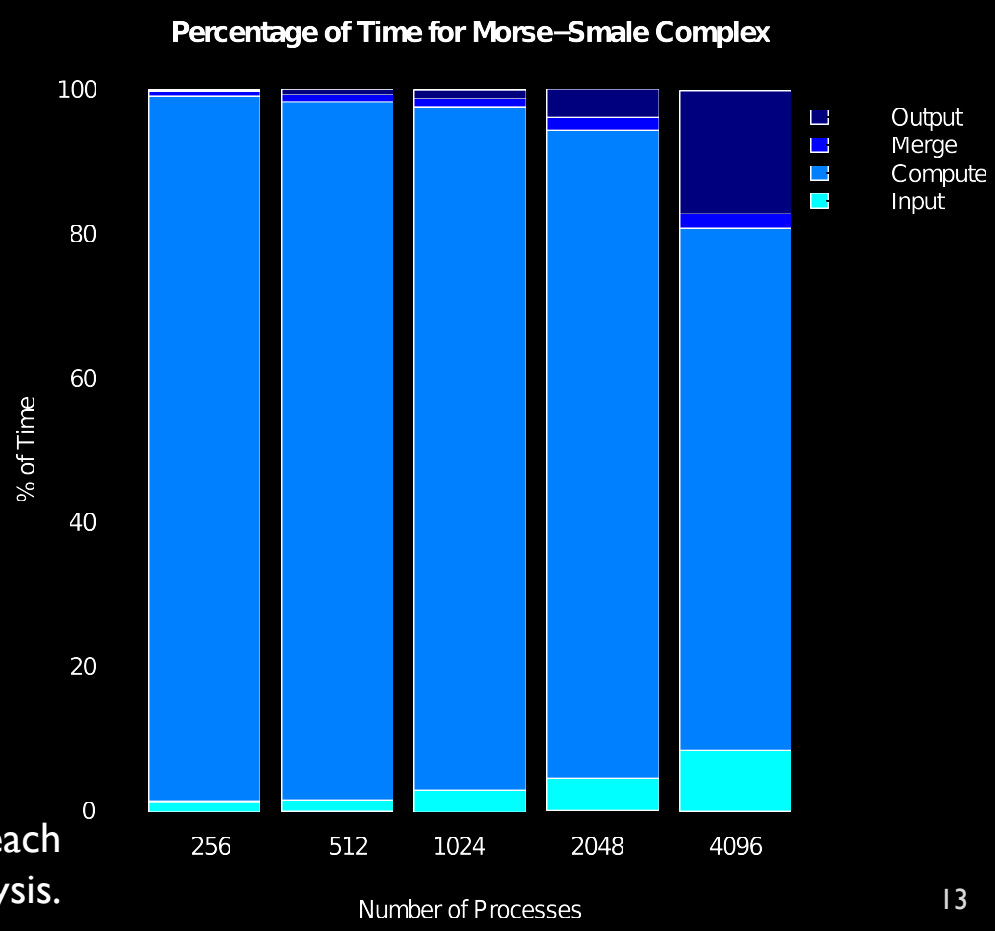


Compute time, merge time, and output size as a function of number of processes, data size, and data complexity.

Cross-Flow Combustion



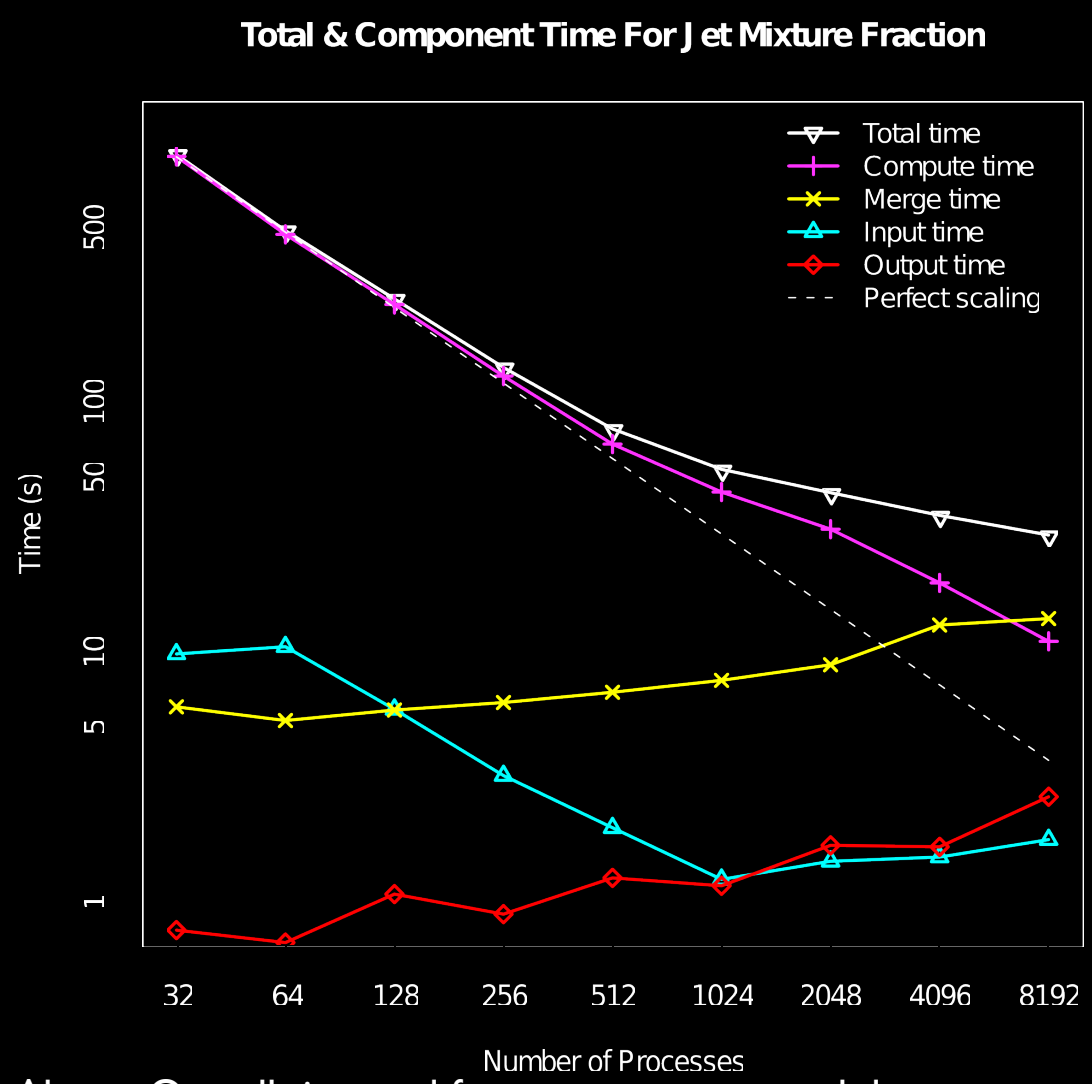


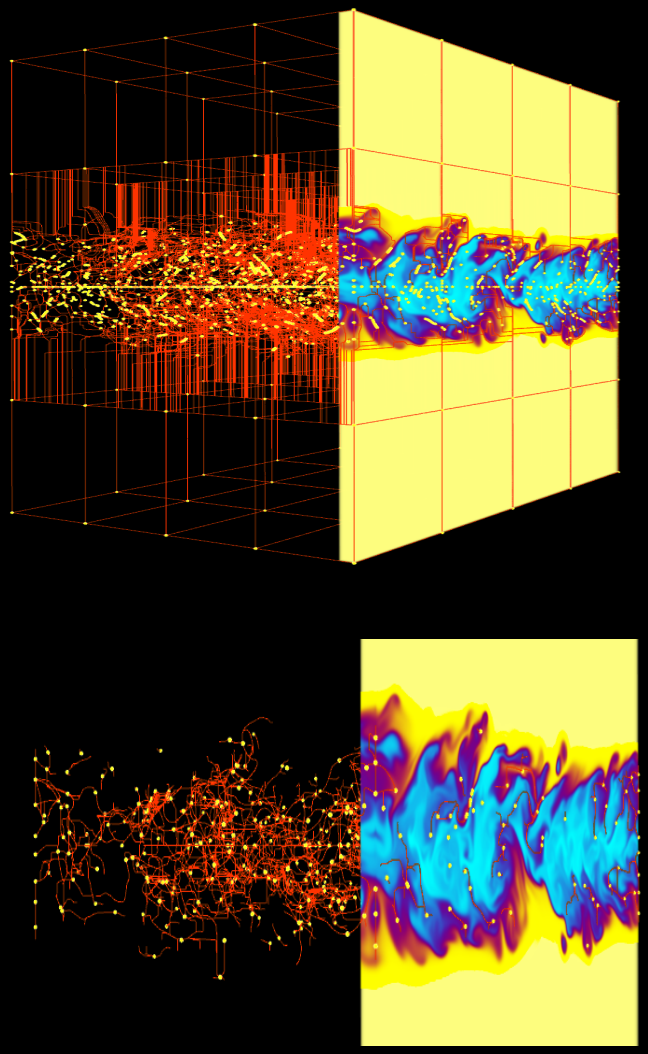


Above: Performance of parallel Morse-Smale analysis in log-log scale. Total time includes reading the dataset from storage, computing the analysis, and writing results to storage. Data size 1 timestep @ 1408x1080x1100.

Right: Percentage of time spent in each component of the Morse-Smale analysis.

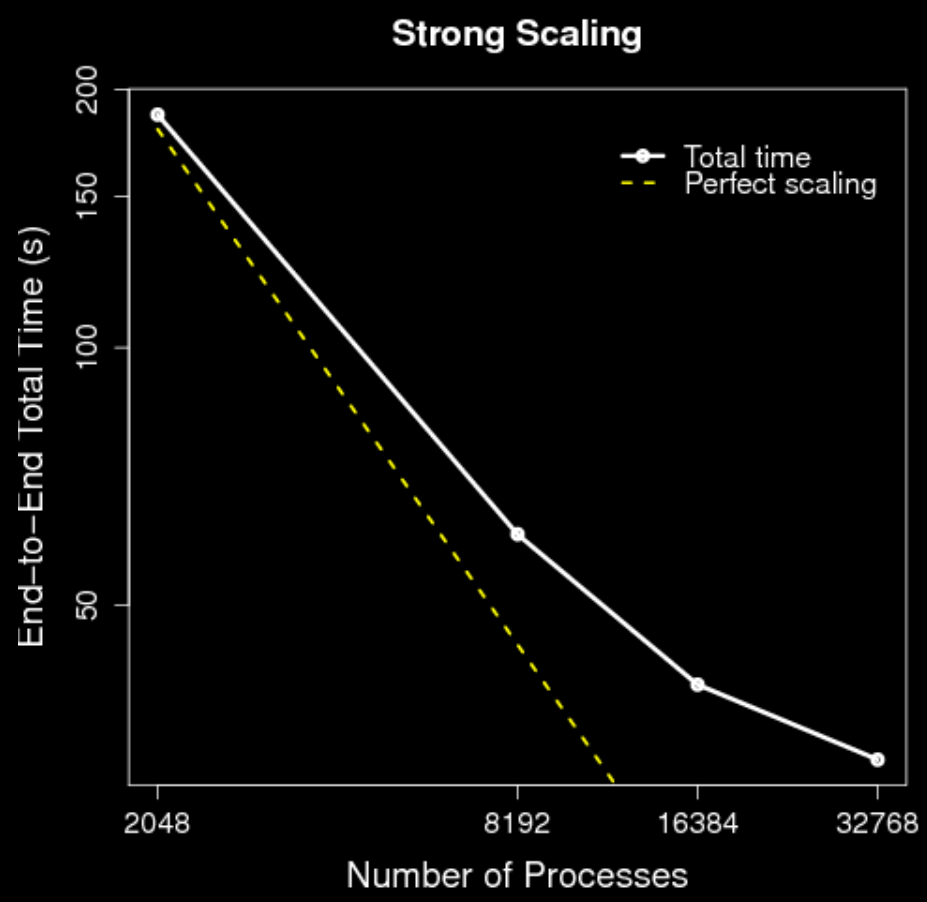
Jet Mixture Fraction

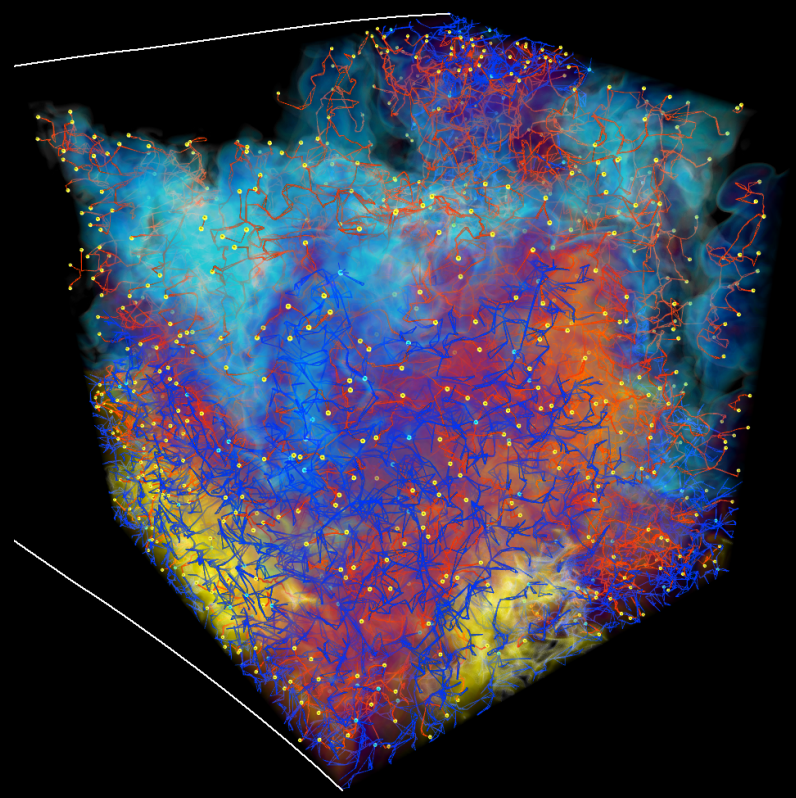




Above: Overall time and four components: read data, compute, merge, and write results, plotted in log-log scale. At small numbers of processes, time is dominated by computing, and at higher numbers of processes by merging. Right: Full complex (top) and simplified complex (bottom). Data size one timestep @ 768x896x512.

Turbulent Mixing





Computation of Morse-Smale complex in 1152^3 Rayleigh-Taylor instability data set results in 35% end-to-end strong scaling efficiency, including I/O. Data size 1 timestep @ 1152 x 1152 x 1152.

Recap and Looking Ahead

Done: Benefits

- Productivity
  - Express complex algorithms flexibly
    - Multiple blocks per process
    - Complete / partial reductions
    - Neighbor inclusion and communication
  - Simplify existing tasks
    - Custom data type creation
    - Compression
- Performance
  - Published scalability
  - Configurable algorithms

To Do: Research Directions

- Advanced decomposition
  - Block groups
- Improved communication algorithms
  - Less synchronous, more overlap with computation
- High-level communication operations
  - Ghost cell exchange, kernel convolution (stencil)
- Load balancing
  - Block overloading, dynamic reassignment
- Programming models
  - MPI + X
- Usability
  - Improved API

References

DIY

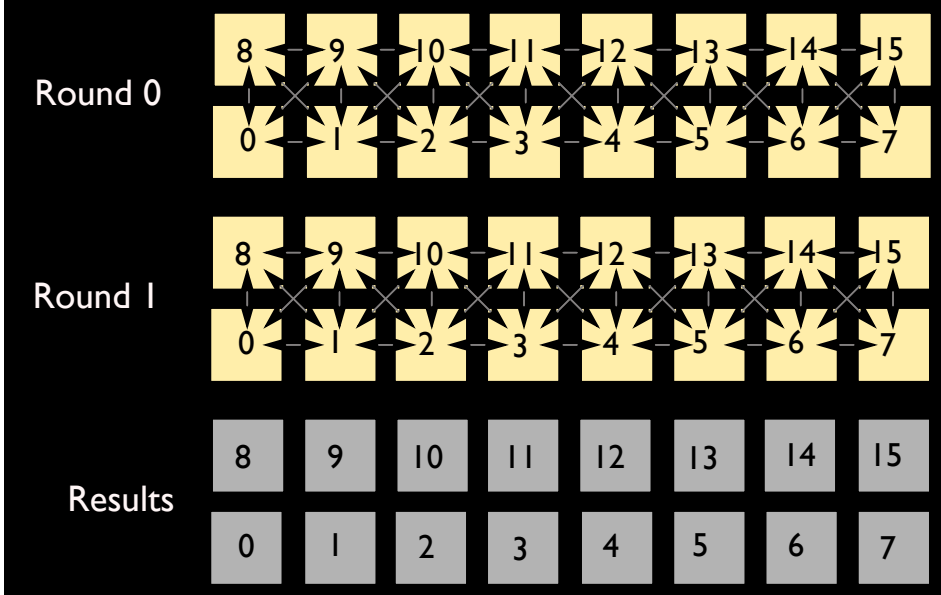
- Peterka, T., Ross, R., Kendall, W., Gyulassy, A., Pascucci, V., Shen, H.-W., Lee, T.-Y., Chaudhuri, A.: Scalable Parallel Building Blocks for Custom Data Analysis. Proceedings of Large Data Analysis and Visualization Symposium (LDAV'11), IEEE Visualization Conference, Providence RI, 2011.
- Peterka, T., Ross, R.: Versatile Communication Algorithms for Data Analysis. 2012 EuroMPI Special Session on Improving MPI User and Developer Interaction IMUDI'12, Vienna, AT.

DIY applications

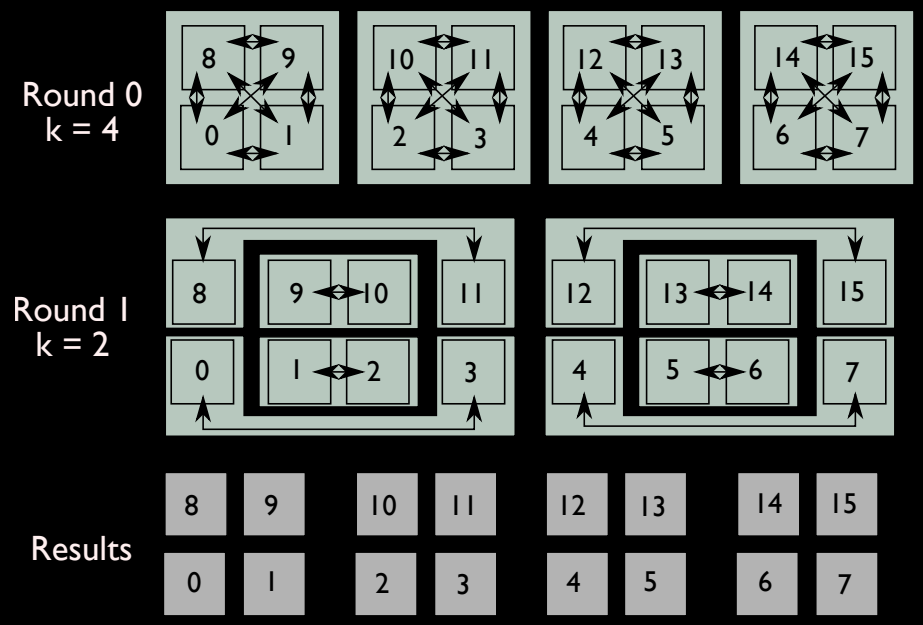
- Peterka, T., Ross, R., Nouanesengsey, B., Lee, T.-Y., Shen, H.-W., Kendall, W., Huang, J.: A Study of Parallel Particle Tracing for Steady-State and Time-Varying Flow Fields. Proceedings IPDPS'11, Anchorage AK, May 2011.
- Gyulassy, A., Peterka, T., Pascucci, V., Ross, R.: The Parallel Computation of Morse-Smale Complexes. Proceedings of IPDPS'12, Shanghai, China, 2012.
- Nouanesengsy, B., Lee, T.-Y., Lu, K., Shen, H.-W., Peterka, T.: Parallel Particle Advection and FTLE Computation for Time-Varying Flow Fields. Proeedings of SC12, Salt Lake, UT.
- Peterka, T., Kwan, J., Pope, A., Finkel, H., Heitmann, K., Habib, S., Wang, J., Zagaris, G.: Meshing the Universe: Integrating Analysis in Cosmological Simulations. Proceedings of the SC12 Ultrascale Visualization Workshop, Salt Lake City, UT.
- Chaudhuri, A., Lee-T.-Y., Zhou, B., Wang, C., Xu, T., Shen, H.-W., Peterka, T., Chiang, Y.-J.: Scalable Computation of Distributions from Large Scale Data Sets. Proceedings of 2012 Symposium on Large Data Analysis and Visualization, LDAV'12, Seattle, WA.

Backup Slides

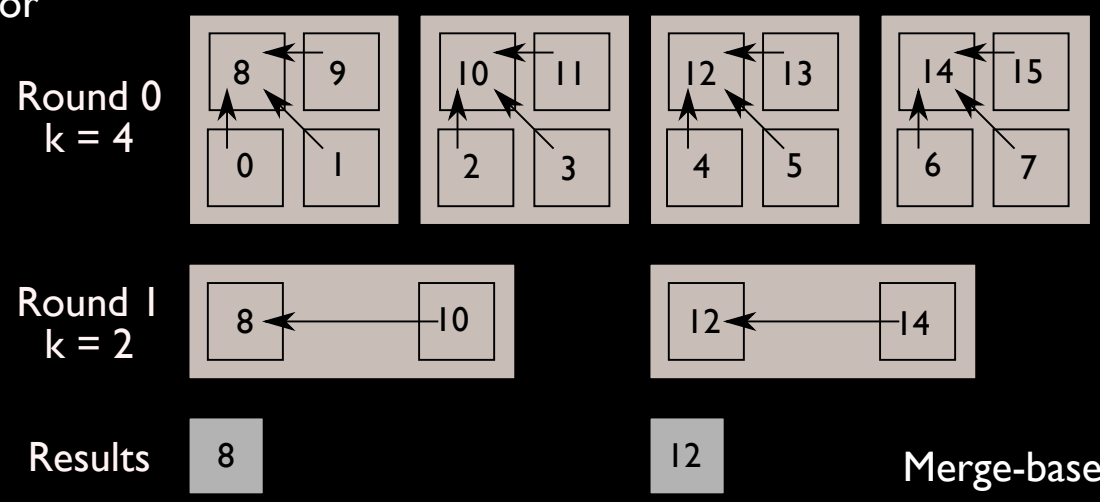
3 Communication Patterns



Nearest neighbor



Swap-based reduction



Merge-based reduction