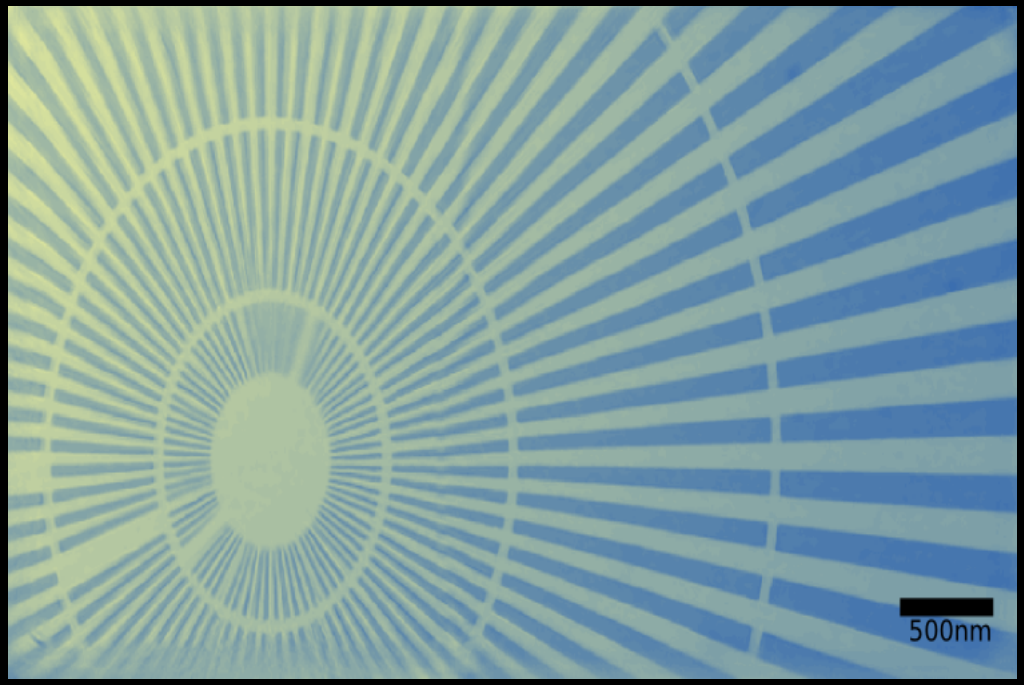
U.S. DEPARTMENT OF ENERGY

Argonne NATIONAL LABORATORY

"I have had my results for a long time, but I do not yet know how I am to arrive at them."

–Carl Friedrich Gauss, 1777-1855

Imaging Meets HPC Through Scalable Data Analysis



Reconstructed image of gold nanostructure with 30nm features *[courtesy Junjing Deng (NU) and Youssef Nashed (ANL)]*

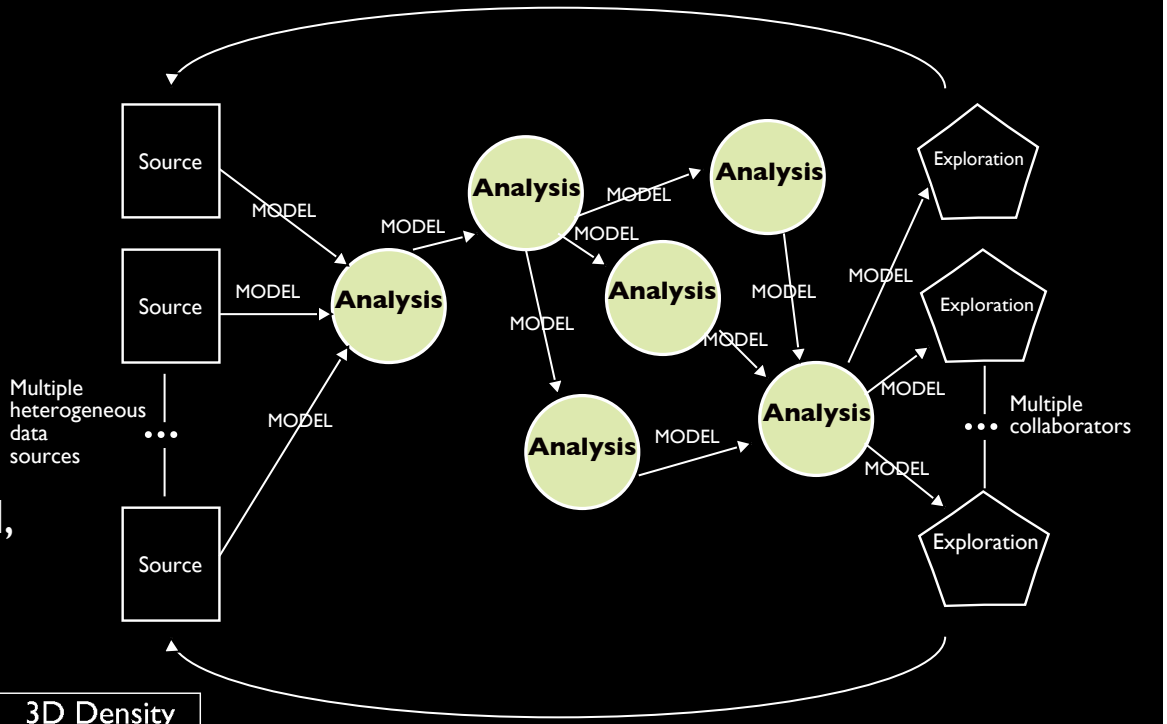
Tom Peterka

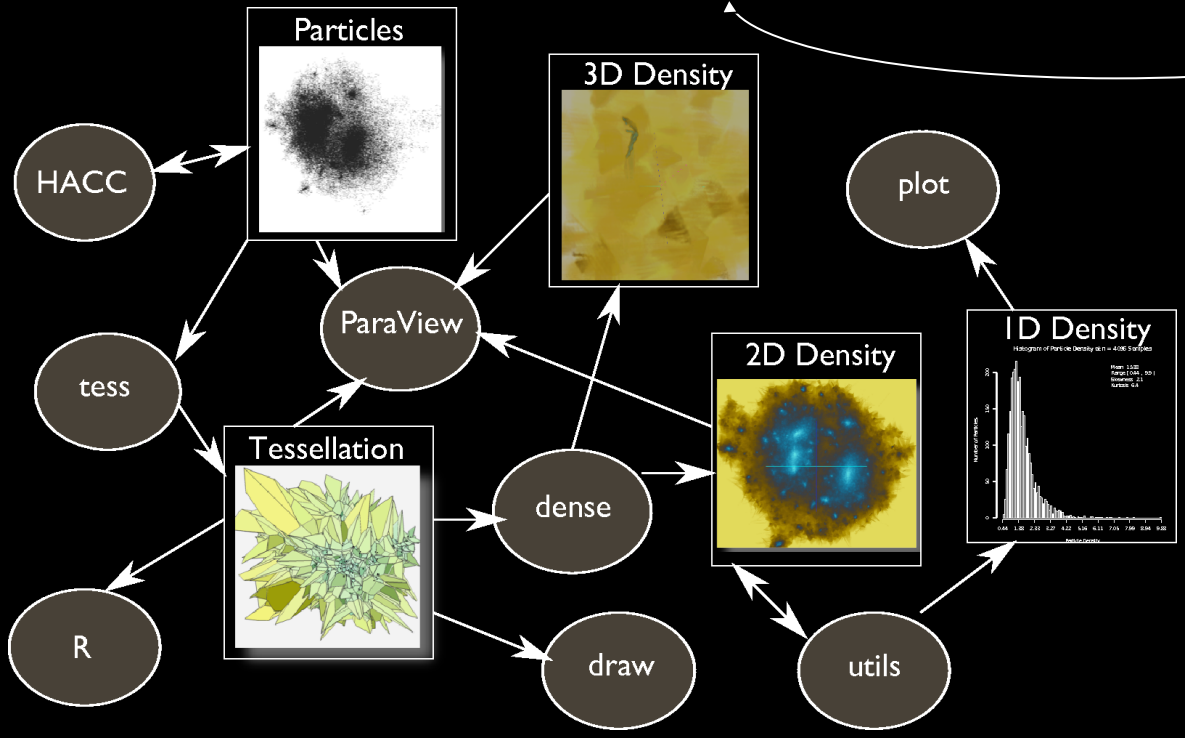
tpeterka@mcs.anl.gov

Mathematics and Computer Science Division

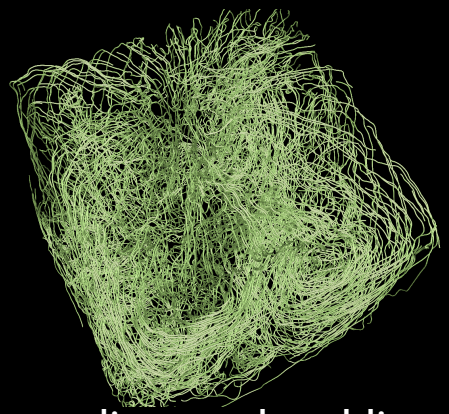
Definition of Data Analysis

- Any data transformation, or a network or transformations.
- Anything done to original data beyond its original generation.
- Can be visual, analytical, statistical, or data management.



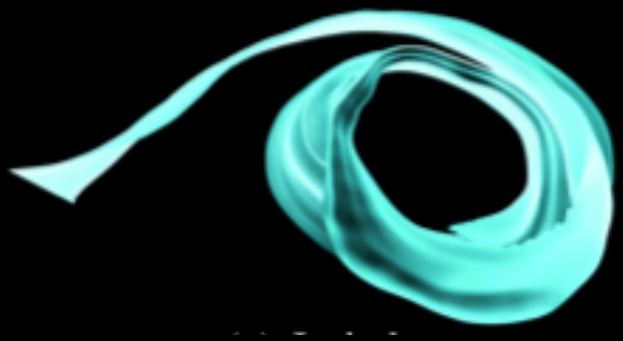


Examples



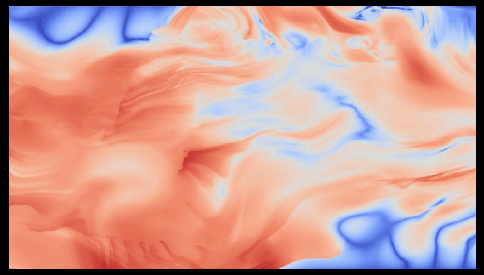
Streamlines and pathlines

[Peterka et al. IPDPS'11]



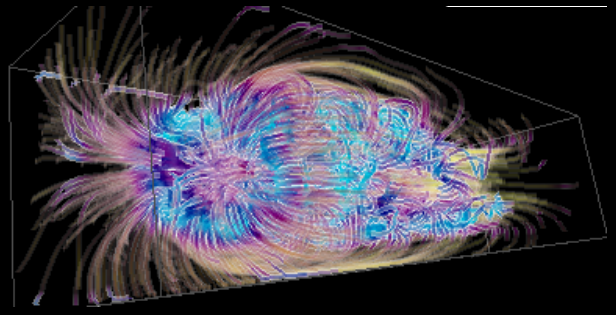
Stream surfaces

[Lu et al. SC14]



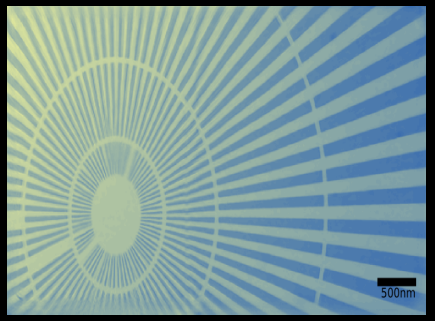
FTLE

[Nouanesengsy et al. SC12]



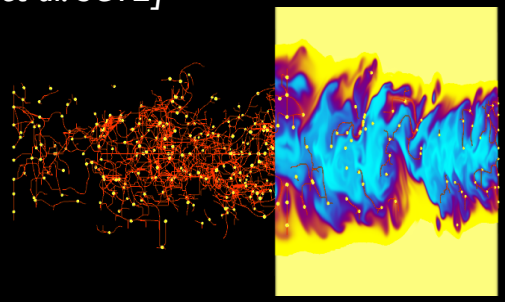
Information entropy

[Chaudhuri et al. LDAV'12]



Ptychography

[Nashed et al. in submission]



Morse-Smale complex

[Gyulassy et al. IPDPS'12]

Voronoi and Delaunay tessellation

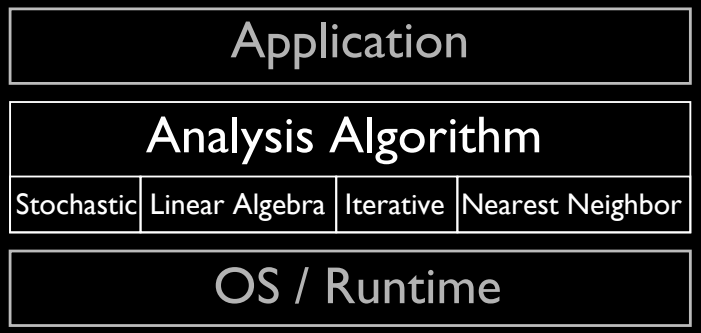
[Peterka et al. SC14]

Common Denominators

- Big science => big data, big machines
- Most analysis algorithms are not up to speed
  - Either serial, or
  - Overheads kill scalability
- Solutions
  - Process data closer to the source
  - Write scalable analysis algorithms
  - Parallelize in various forms
  - Build software stacks of useful and reusable layers
- Usability and workflow
  - Develop libraries rather than tools
  - Users write small main programs and call into libraries

Abstractions Matter

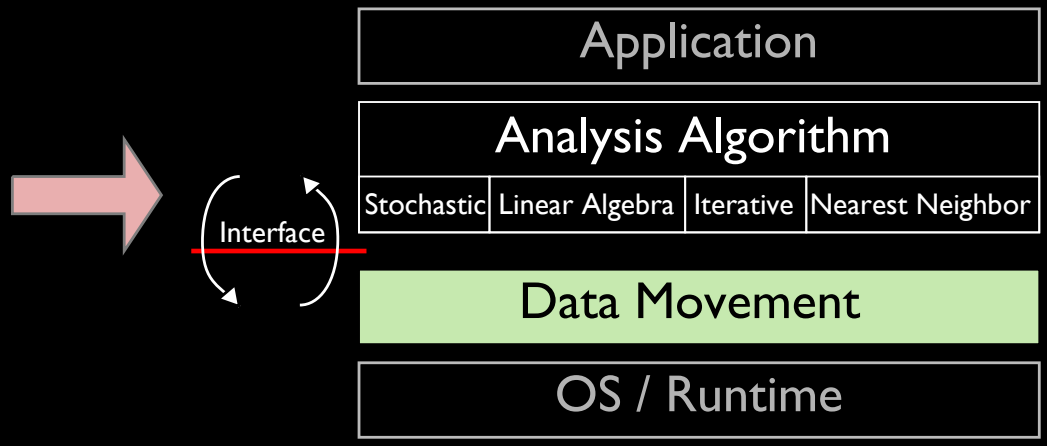
Parallelizing by hand




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

or

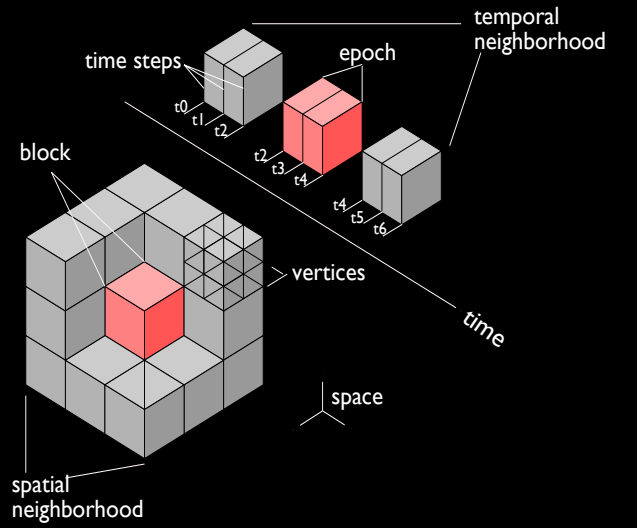
With a data movement layer

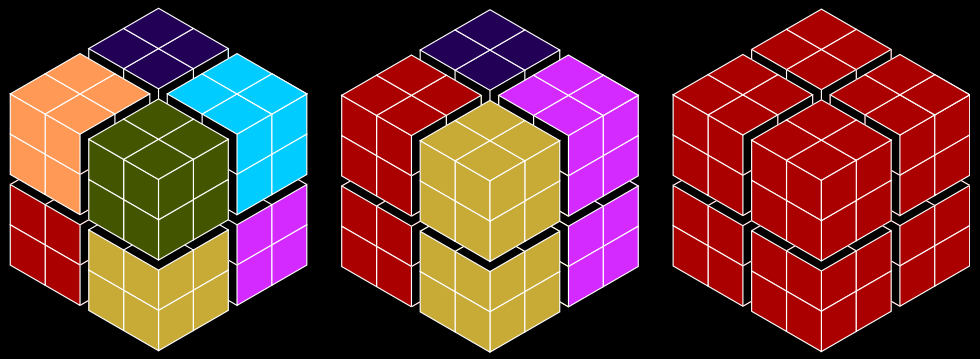



```
void ParallelAlgorithm() {
   …
   LocalAlgorithm();
   …
   DIY_Merge_blocks();
   …
   DIY_File_write()
}
```

Features of a Data Movement Layer

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

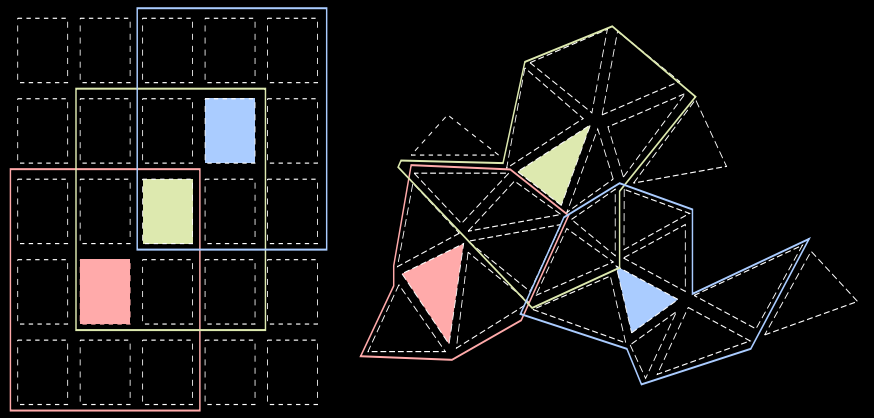




8 processes

4 processes

1 process



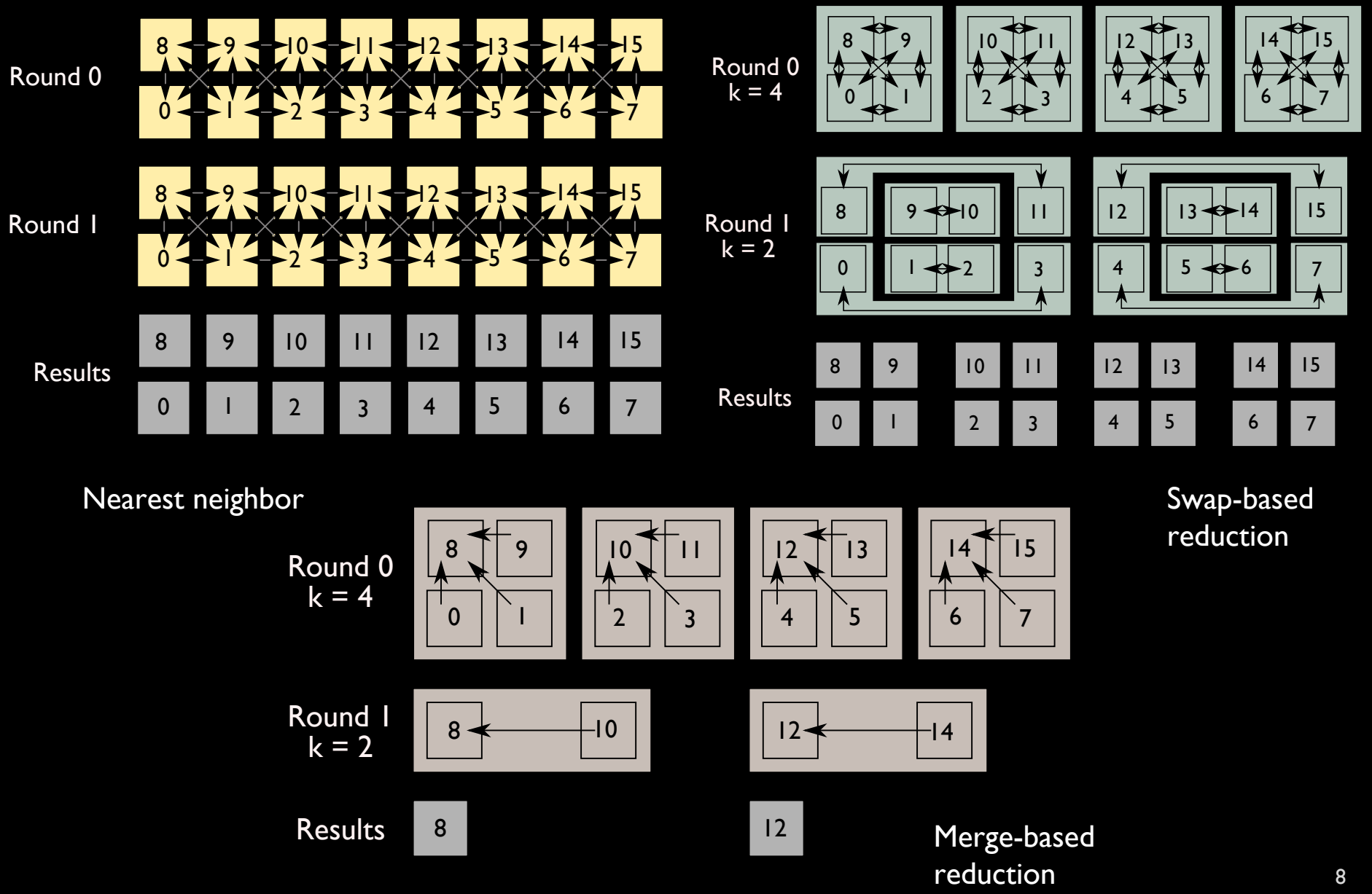
Two examples of 3 out of a total of 25 neighborhoods

Data Movement Patterns

	Analysis	Communication	
Regular	Sort-Last Rendering	Swap-Based Reduction	Homogeneous Data
Regular	Morse-Smale Complex	Merge-Based Reduction	Heterogeneous Data
Regular	Information Entropy	Merge-Based Reduction	Heterogeneous Data
Semi Regular	Particle Tracing	Neighborhood Exchange	
Semi Regular	Voronoi Tessellation	Neighborhood Exchange	
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.

3 Communication Patterns

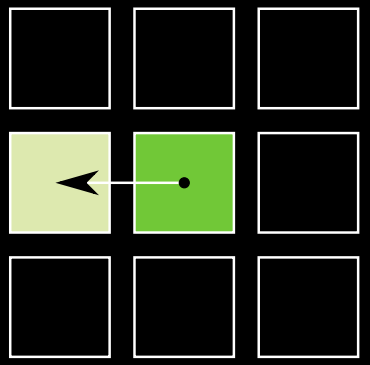


Different Neighborhood Communication Patterns

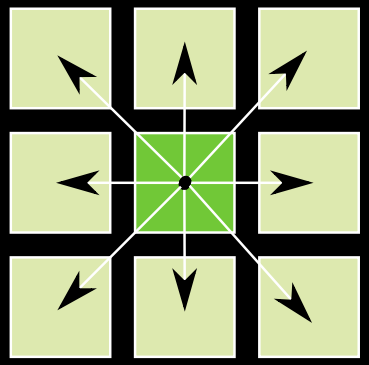
DIY provides point to point and different varieties of collectives within a neighborhood via its enqueue_item mechanism. Items are enqueued are subsequently exchanged (2 steps).

How to enqueue items for neighbor exchange

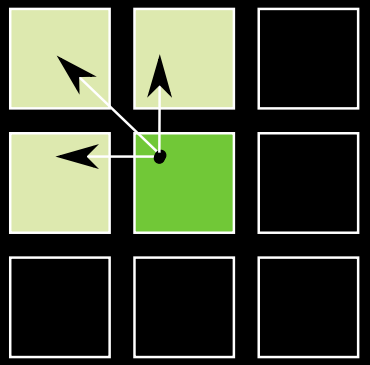
- DIY offers several options
- Send to a particular neighbor or neighbors, send to all nearby neighbors, send to all neighbors
- Support for periodic boundary conditions involves tagging which neighbors are periodic and calling user-defined transform on objects being sent to them



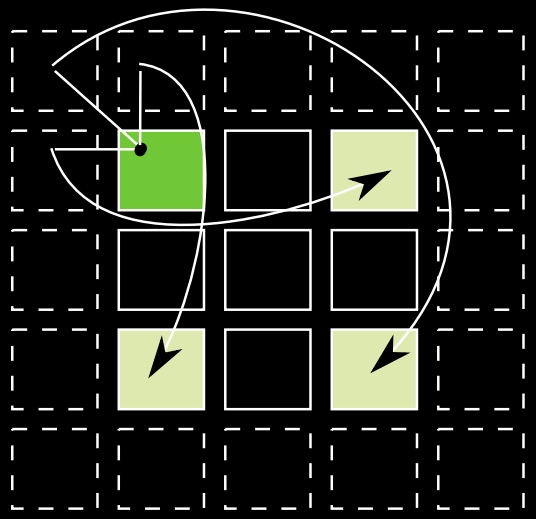
Send to only specific neighbors, indicated in various ways



Send to all neighbors



Send to all neighbors near enough to a target point



Support for wraparound neighbors (periodic boundary conditions)

DIY

helps the user write data-parallel analysis algorithms by decomposing a problem into blocks and communicating items between blocks.

Features

Parallel I/O to/from storage

Domain decomposition

Network communication

Utilities

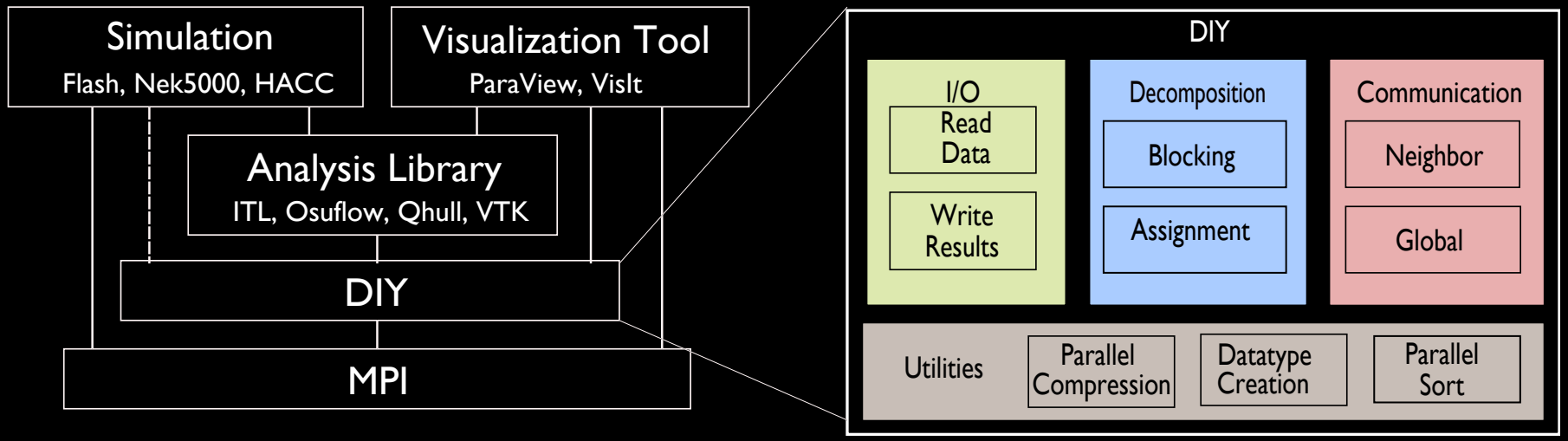
Library

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

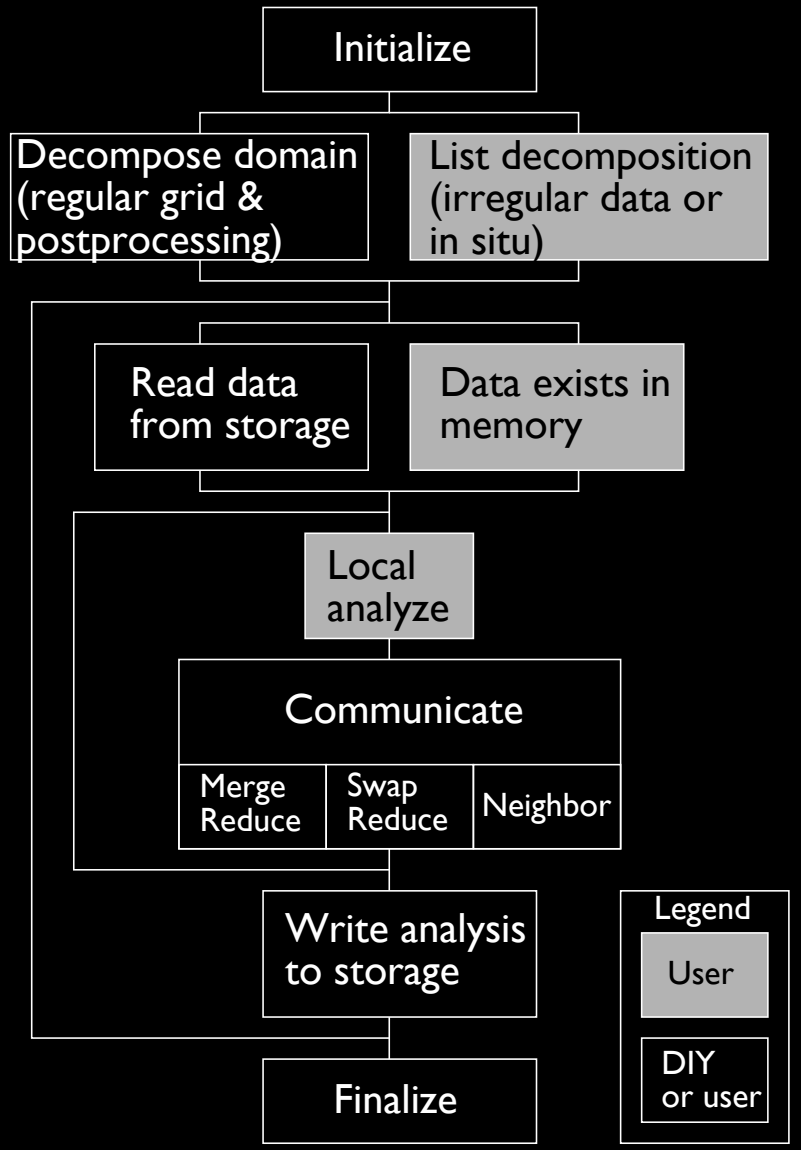
Writing a DIY Program

Documentation

- README for installation
- User's manual with description, examples of custom datatypes, complete API reference

Tutorial Examples

- Block I/O: Reading data, writing analysis results
- Static: Merge-based, Swap-based reduction, Neighborhood exchange
- Time-varying: Neighborhood exchange
- Spare thread: Simulation and analysis overlap
- MOAB: Unstructured mesh data model
- VTK: Integrating DIY communication with VTK filters
- R: Integrating DIY communication with R stats algorithms
- Multimodel: multiple domains and communicating between them

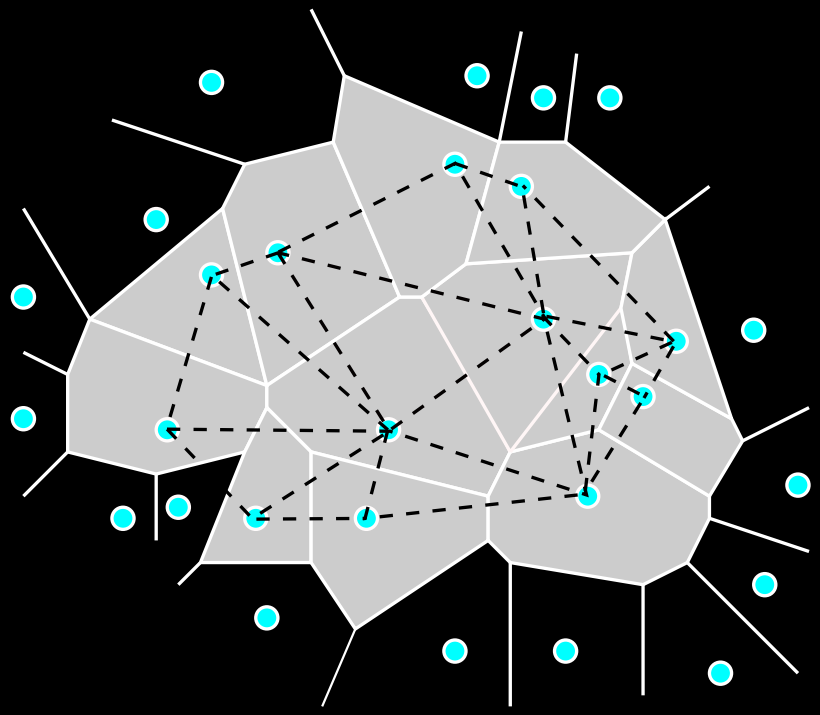


Parallel Tessellation

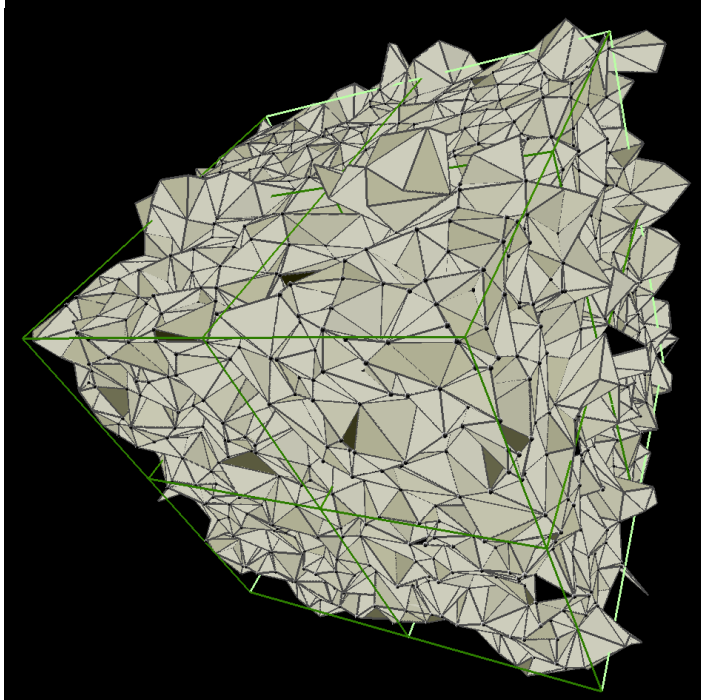
We developed a prototype library for computing in situ Voronoi and Delaunay tessellations from particle data and applied it to cosmology, molecular dynamics, and plasma fusion.

Key Ideas

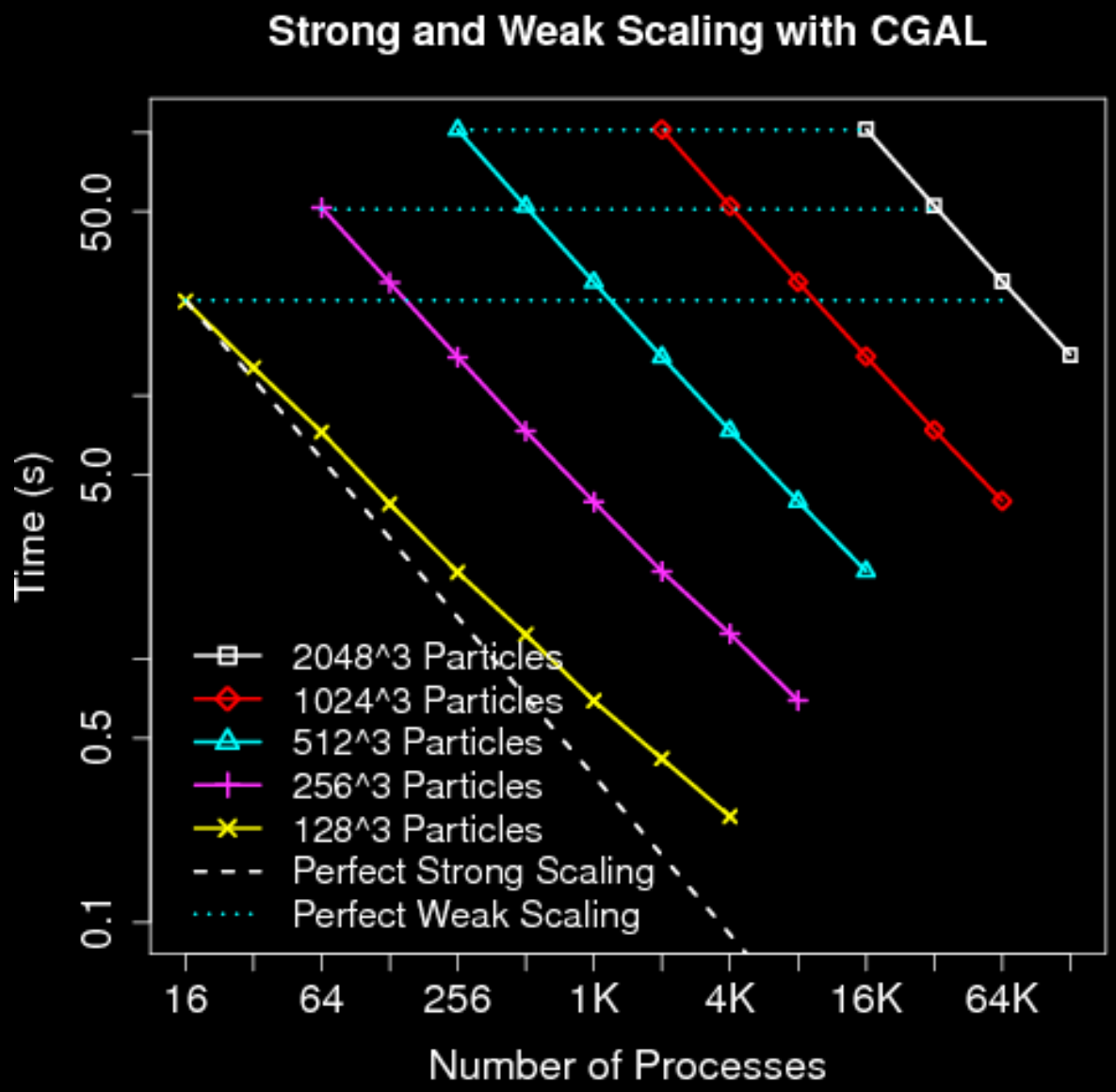
- Mesh tessellations convert sparse point data into continuous dense field data.
- Meshing output of simulations is data-intensive and requires supercomputing resources
- No large-scale data-parallel tessellation tools exist.
- We developed such a library, tess.
- We achieved good parallel performance and scalability.
- Widespread GIS applicability in addition to the datasets we tested.



Scalability



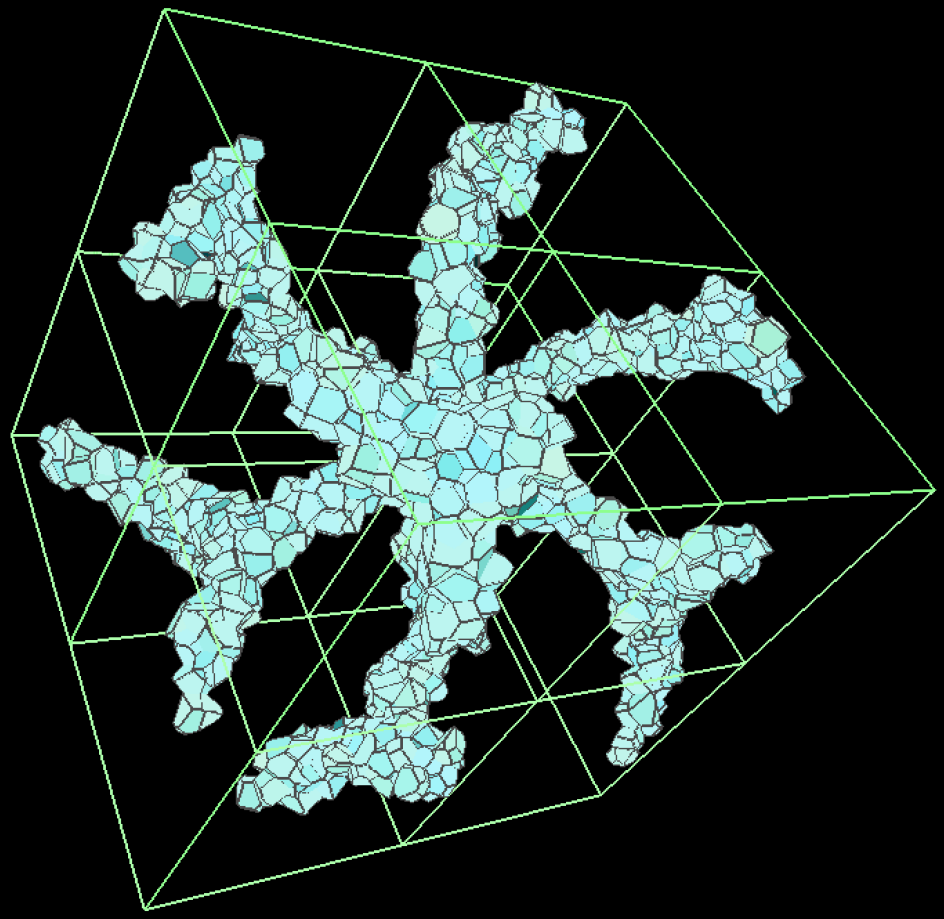
Strong and weak scaling for up to 2048^3 synthetic particles and up to 128K processes (excluding I/O) shows up to 90% strong scaling and up to 98% weak scaling.



Applications in Molecular Dynamics

[Courtesy of Carolyn Phillips, ANL]

In simulations of microphase separated soft matter systems, populations of molecules self-organize to form two or more domains. Each domain contains only one type of bead. These domains can form complicated geometries such as the double gyroid.

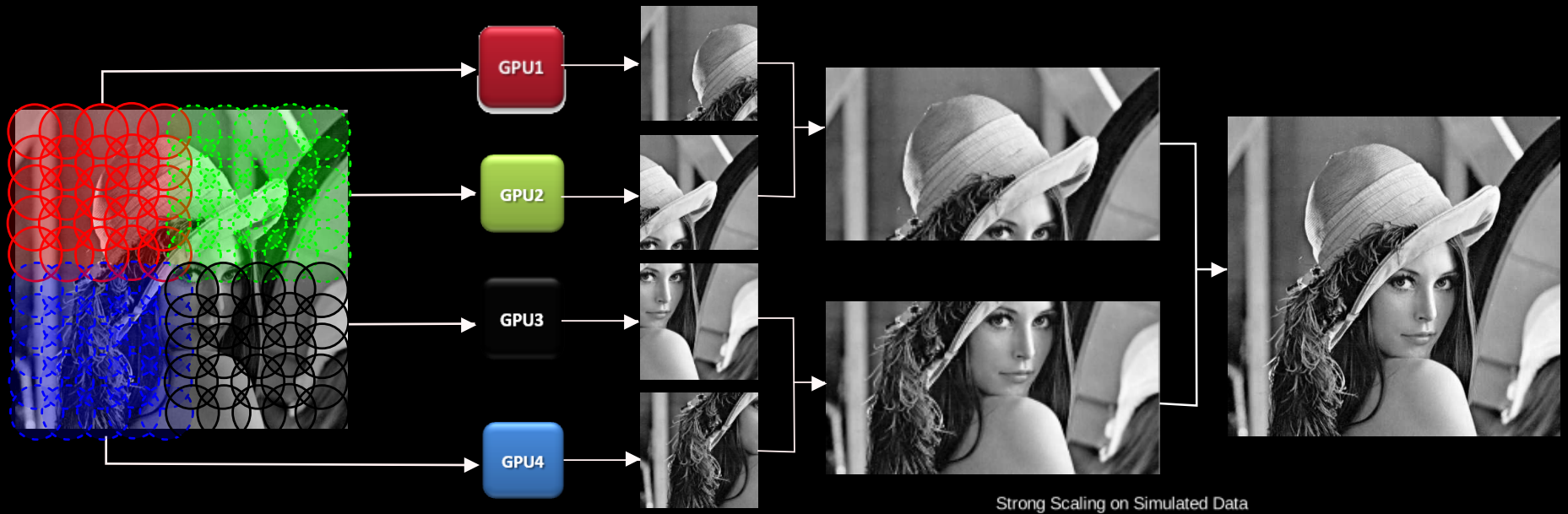


We tessellated over 300 time steps of a system of 2,765 chains of an A-B-A triblock copolymer (176,960 total beads) found in an alternating gyroid morphology. Summing the Voronoi cells of the A and B species suggests that because the B part has to stretch or fold, the B domain dilates relative to the A domain.

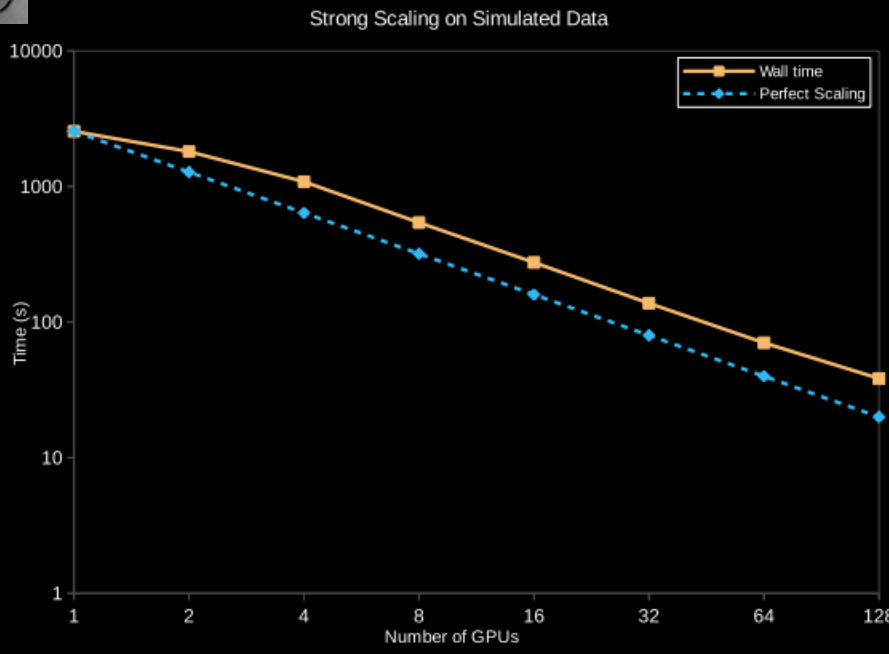
The figure above shows the Voronoi tessellation of 1,000 A-B-C ``telechelics'' composed of two nanospheres (A and C) connected by a polymer tether beads (B) for a total of 8,000 beads in a double gyroid morphology. Only the Voronoi cells associated with the A species are shown. Such surfaces are usually constructed by using isosurface methods, which require averaging over many time steps; whereas by using the tessellation, such surfaces can be constructed for every time step.

Ptychographic Image Reconstruction

[Courtesy of Youssef Nashed, ANL]



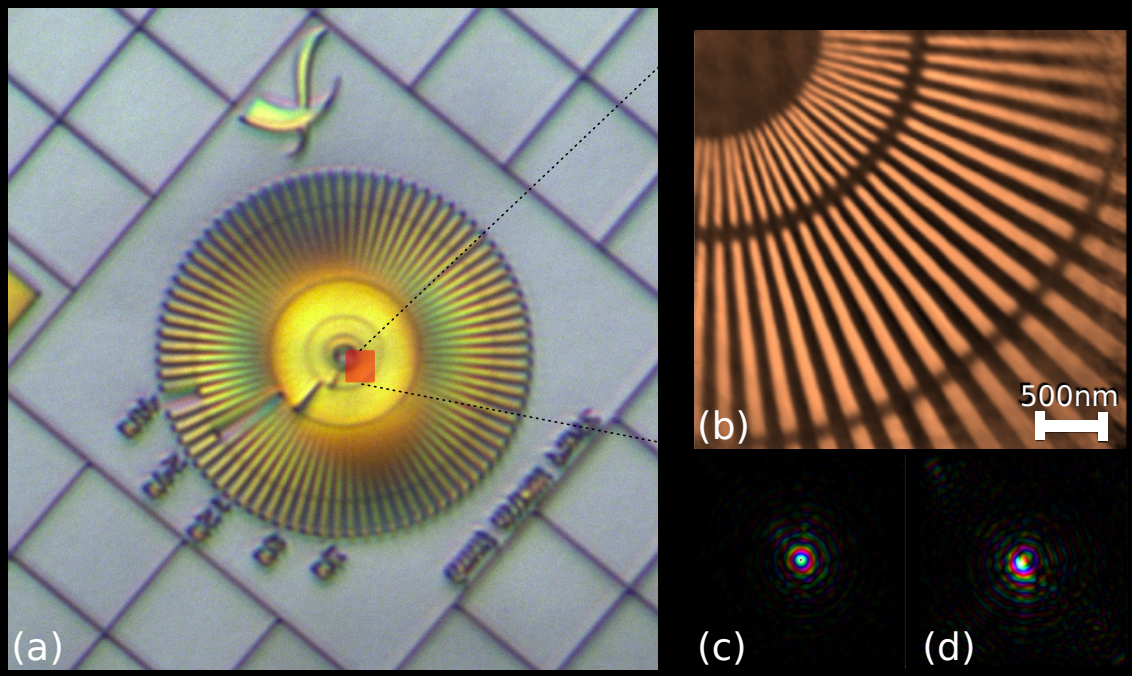
An illustration of multi-GPU phase retrieval. The diffraction patterns are subdivided and distributed among available GPUs. Pairwise stitching is performed on the partial reconstructions attained by phase retrieval. The initial diffraction patterns are subdivided into DIY blocks merged together using DIY's merge-based reduction communication pattern. Strong scaling efficiency on synthetic data is approximately 55% on 21,000 256x256 images.

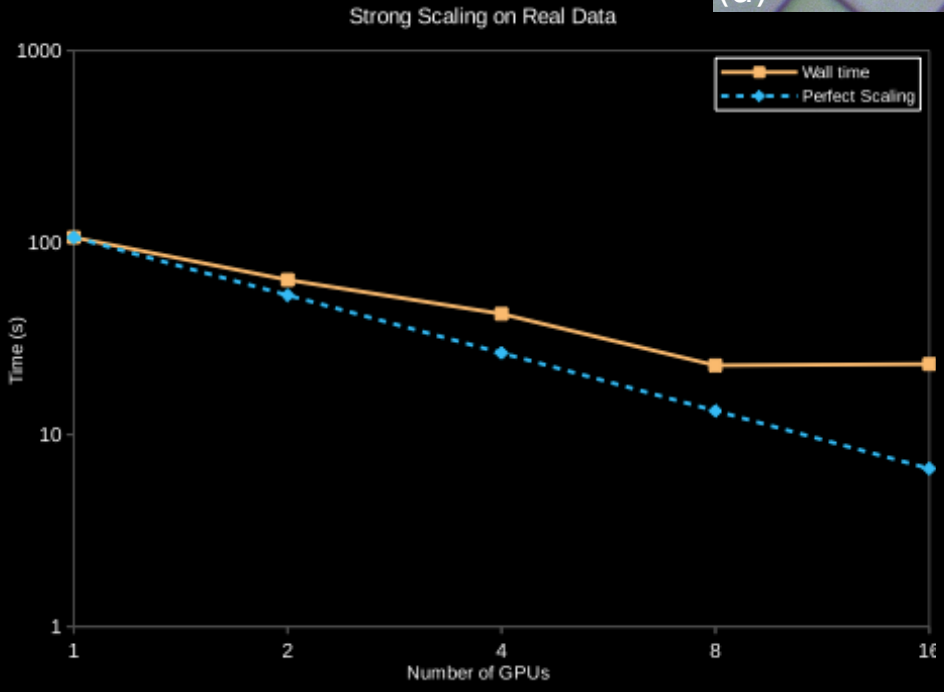


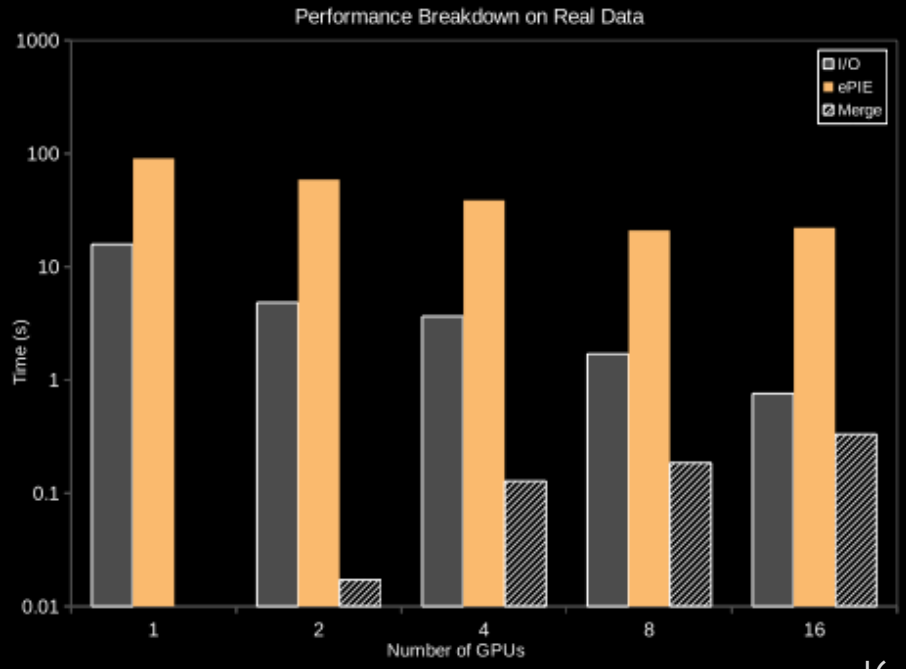
Parallel Reconstruction Performance

[Courtesy of Youssef Nashed, ANL]

A gold Siemens star test pattern, with 30 nm smallest feature size, was raster scanned through a 26×26 grid using a step size of 40nm and an exposure time of 0.6s per scan point, using a 5.2 keV X-ray beam. The total scanning time was about 20 minutes.







Looking Ahead

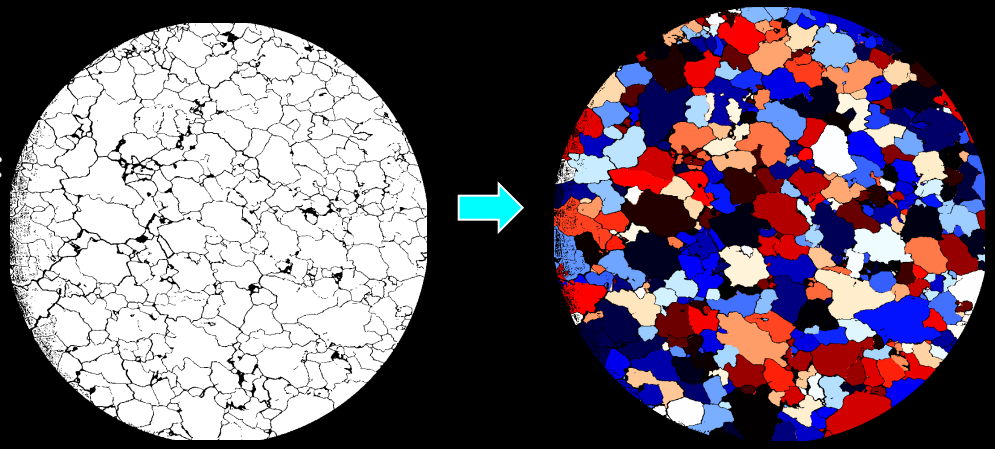
A collaboration between ANL and LBL (Tom Peterka and Dmitriy Morozov) resulted both in new applications of DIY and a new prototype implementation of DIY2 featuring multithreaded in/out of core block management. Includes shared repositories (diy, diy2, tess, tess2) and publications [SC14].

Ongoing discussions between other ANL and LBL staff. (ANL: Peterka, Nashed, Jacobsen, Vine, McNulty; LBL: Ushizima, Krishnan, Shapiro, Marchesini, Maia, Sethian) regarding ptychographic reconstruction approaches.

[Courtesy of Dmitriy Morozov, LBL]

LBL (Dmitriy Morozov and Patrick O'Neil) developed tools for segmentation and connectivity analysis of granular and porous media using diy2.

Left: 3D image of a granular material (flexible sandstone) acquired at ALS by Michael Manga and Dula Parkinson. (Data: 2560 × 2560 × 1276). Right: Watershed segmentation of the material identifies individual grains (run on Edison @ NERSC) [courtesy Morozov, O'Neil (LBL)].



Further Reading

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. Proceedings 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.
- Peterka, T., Morozov, D., Phillips, C.: High-Performance Computation of Distributed-Memory Parallel 3D Voronoi and Delaunay Tessellation. Proceedings of SC14, New Orleans, LA, 2014.
- Lu, K., Shen, H.-W., Peterka, T.: Scalable Computation of Stream Surfaces on Large Scale Vector Fields. Proceedings of SC14, New Orleans, LA, 2014.

U.S. DEPARTMENT OF ENERGY

Argonne NATIONAL LABORATORY

"The purpose of computing is insight, not numbers."

–Richard Hamming, 1962

Acknowledgments:

Facilities

Argonne Leadership Computing Facility (ALCF)

Oak Ridge National Center for Computational Sciences (NCCS)

National Energy Research Scientific Computing Center (NERSC)

Funding

DOE SDMAV Exascale Initiative

DOE Exascale Codesign Center

DOE SciDAC SDAV Institute

People

Dmitriy Morozov (LBL)

https://bitbucket.org/diatomic/diy

Tom Peterka

tpeterka@mcs.anl.gov

Mathematics and Computer Science Division