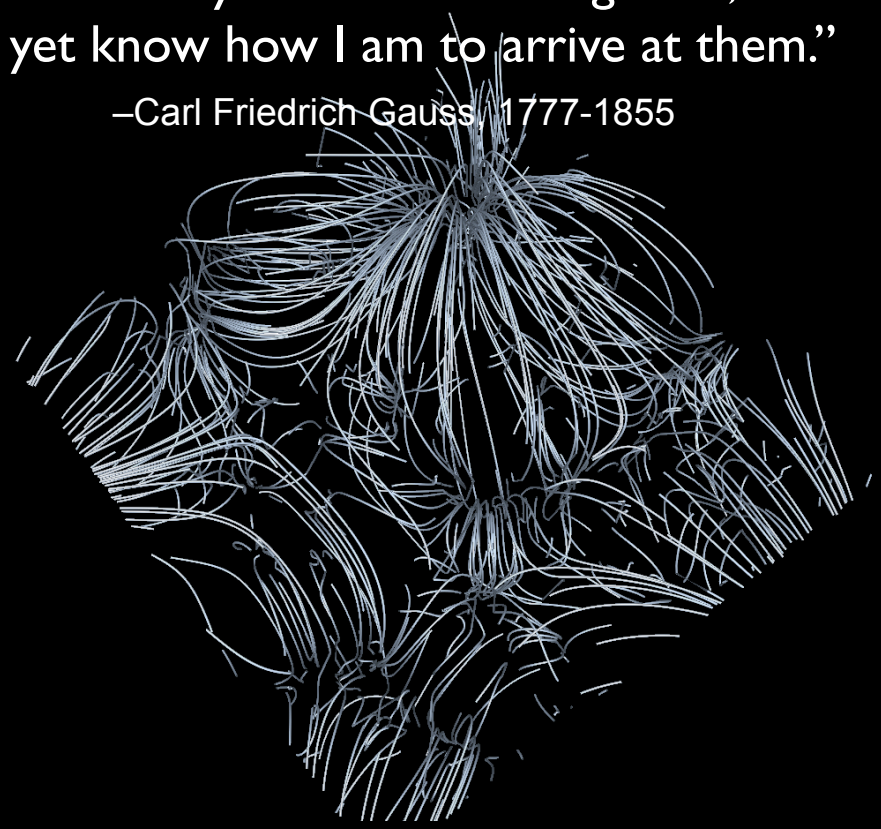
Scalable Parallel Building Blocks for Custom Data Analytics

"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



Early stages of Rayleigh-Taylor Instability flow

Tom Peterka

tpeterka@mcs.anl.gov

Mathematics and Computer Science Division

Exabytes, not Exaflops: Data-Intensive Computing and Analysis

Normalized Storage / Compute Metrics Today

Machine	FLOPS (PF/s)	Storage B/W (GB/s)	Bytes comp. per byte stored
LLNL BG/L	0.6	43	$O(10^3)$
Jaguar XT4	0.3	42	$O(10^3)$
Intrepid BG/P	0.6	50	$O(10^3)$
Roadrunner	1.0	50	$O(10^4)$
Jaguar XT5	1.4	42	$O(10^4)$

In 2001, Flops per bytes stored was approximately 500, John May, 2001.

Future Architecture Design Points

Specification	2010	2018	X Change
FLOPS	2 PF/s	1 EF/s	500
Memory size	0.3 PB	10 PB	33
Memory BW	25 GB/s	400 GB/s	16
Network BW	1.5 GB/s	50 GB/s	33
Storage BW	0.2 TB/s	20 TB/s	100

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, 12/09.

Percent Saved of Computed Data

Code	Domain	% Saved	PI
FLASH	Astrophysics	10	Ricker
Nek5000	CFD	1	Fischer
CCSM	Climate	1	Jacob
GCRM	Climate	10	Cram
S3D	Combustion	1-5	Bennett

CScADS Sci. Data Analysis & Visualization Workshop '09

"Analysis and visualization will be limiting factors in gaining insight from exascale data."

–Dongarra et al., International Exascale Software Project Draft Road Map, 2009.

Analysis and its Impact on Future Hardware and Software: Codesign

Analysis Characteristics:

-Various kernels need to be considered

-eg. Ray casting, image compositing, particle tracing, topology

-Not Compute or Memory bound

-Network, Storage (data movement) bound

-Global reductions

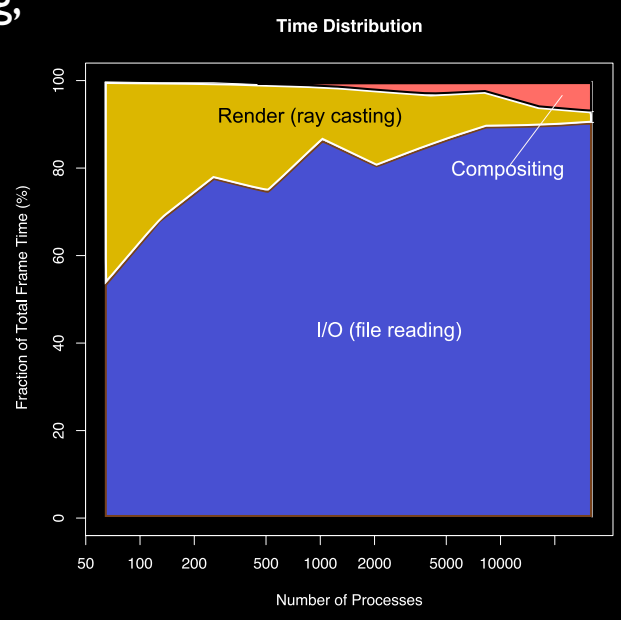
-Local nearest neighbor communication

-Asynchronous and synchronous communication

-Highly imbalanced workloads

-Memory capacity can be limiting

-Short run time at scale



Analysis is dominated by data movement: I/O and communication.

System software:

-MPI + threads

-Efficient Parallel I/O

-Efficient reduction

-Sparse collectives

-Nonblocking collectives

-One-sided communication

-Load balancing libraries

System hardware:

-More or less powerful CPUS ok

-Separate collective networks

-Node-local storage

-Combined CPU-GPU memory and network

Data Analysis at *scale

Keys to parallel data analysis at scale

-Decompose the domain

-Assign to processors

-Access data

-Combine local and global operations

-Scale efficiently

-Balance load, minimize communication

-Store results

Sounds just like a parallel computation problem

The simulation already does decomposition, processor assignment, data access

Integrate with simulation

Approach

-Analysis driven by scientists themselves and their codes

-RASV:

Reduce: data mining, probabalistic, data transformations, feature identification (data triage)

Analyze: machine learning, statistical, automated approaches

Store: less, more important data

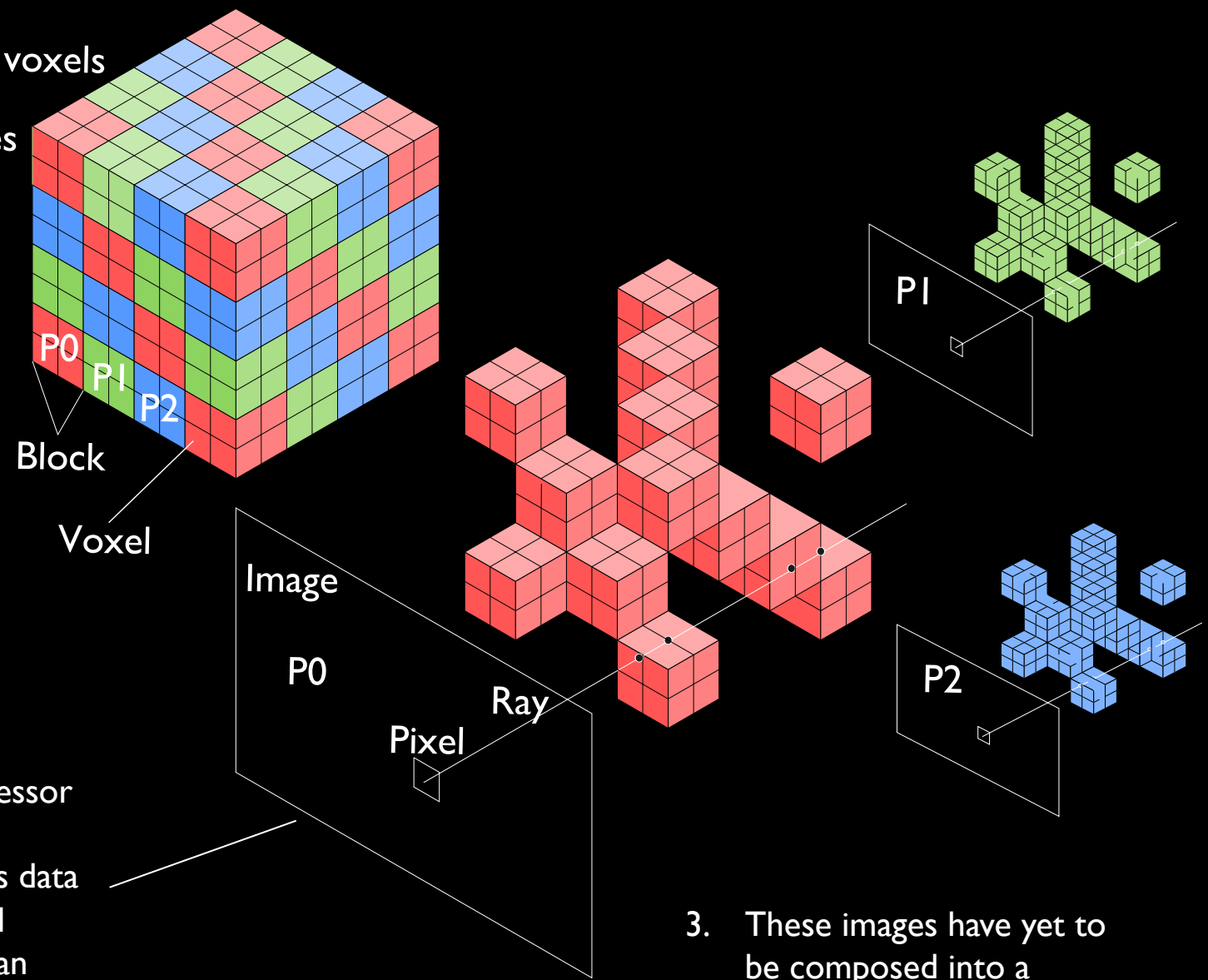
Visualize: in situ, coprocessing, postprocessing

"The combination of massive scale and complexity is such that high performance computers will be needed to analyze data, as well as to generate it through modeling and simulation."

–Lucy Nowell, LAB 10-256, 2010.

Case Study: Parallel Volume Rendering

8^3 = 512 voxels
64 blocks
3 Processes

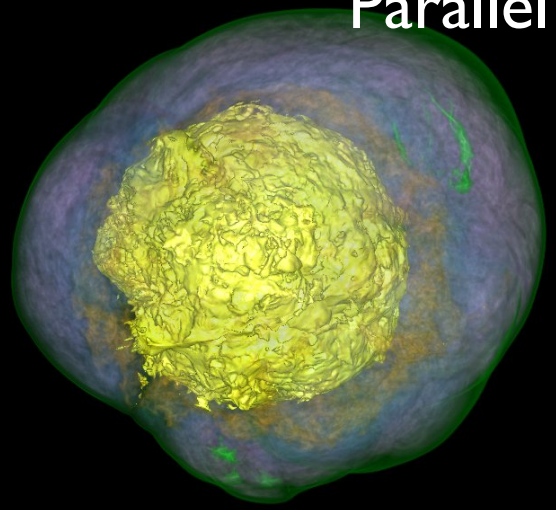


1. Group data into blocks and assign blocks to processors.

2. Each processor casts rays through its data blocks and produces an image of its data.

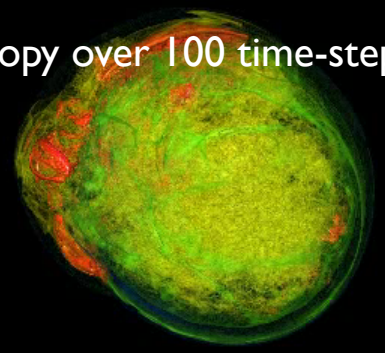
3. These images have yet to be composed into a single, final image.

Parallel Volume Rendering Performance

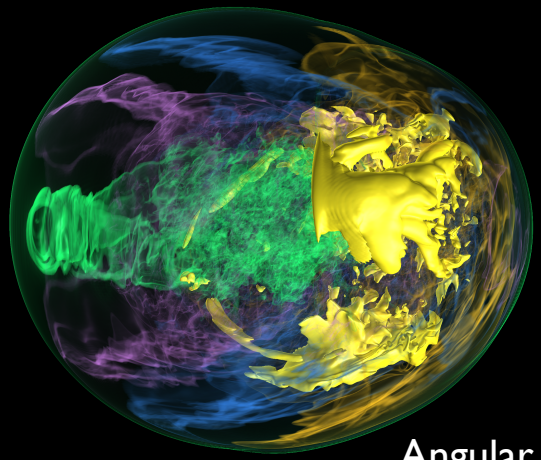


Volume rendering of shock wave formation in core-collapse supernova dataset, courtesy of John Blondin, NCSU. Structured grid of 1120^3 data elements, 5 variables per cell.

Entropy over 100 time-steps

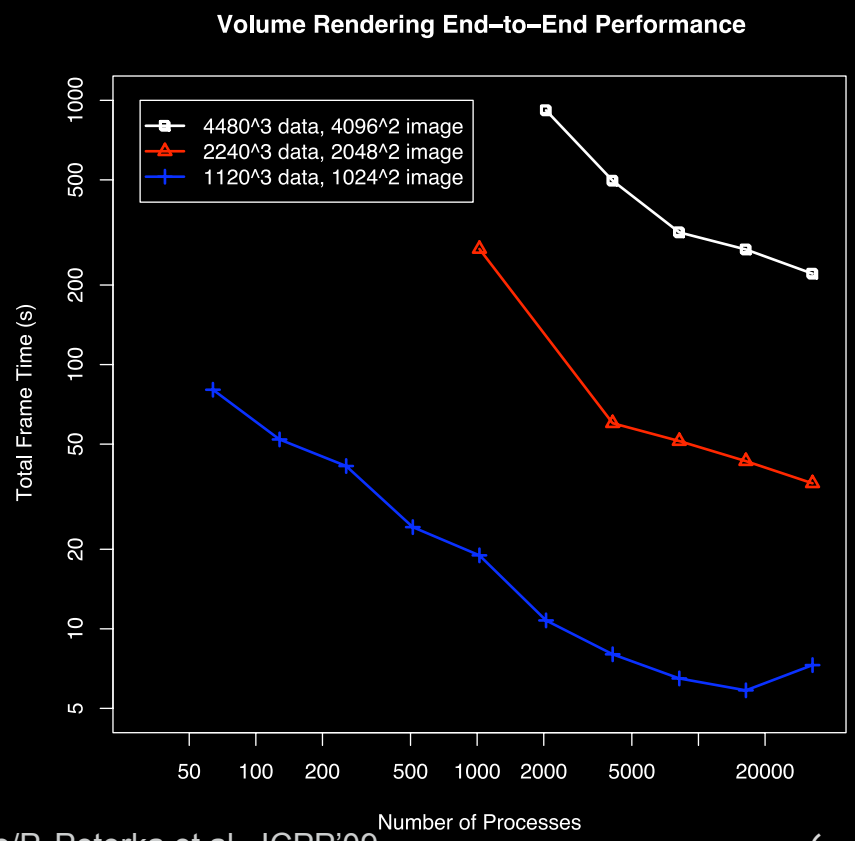


Pressure at time-step 1530



Angular momentum at time-step 1492

Scalability over a variety of data, image, and system sizes.



Case Study: Large Scale Parallel Image Compositing

The final stage in sort-last parallel visualization algorithms:

1. Partition data among processes
2. Visualize local data
3. Composite resulting images into one

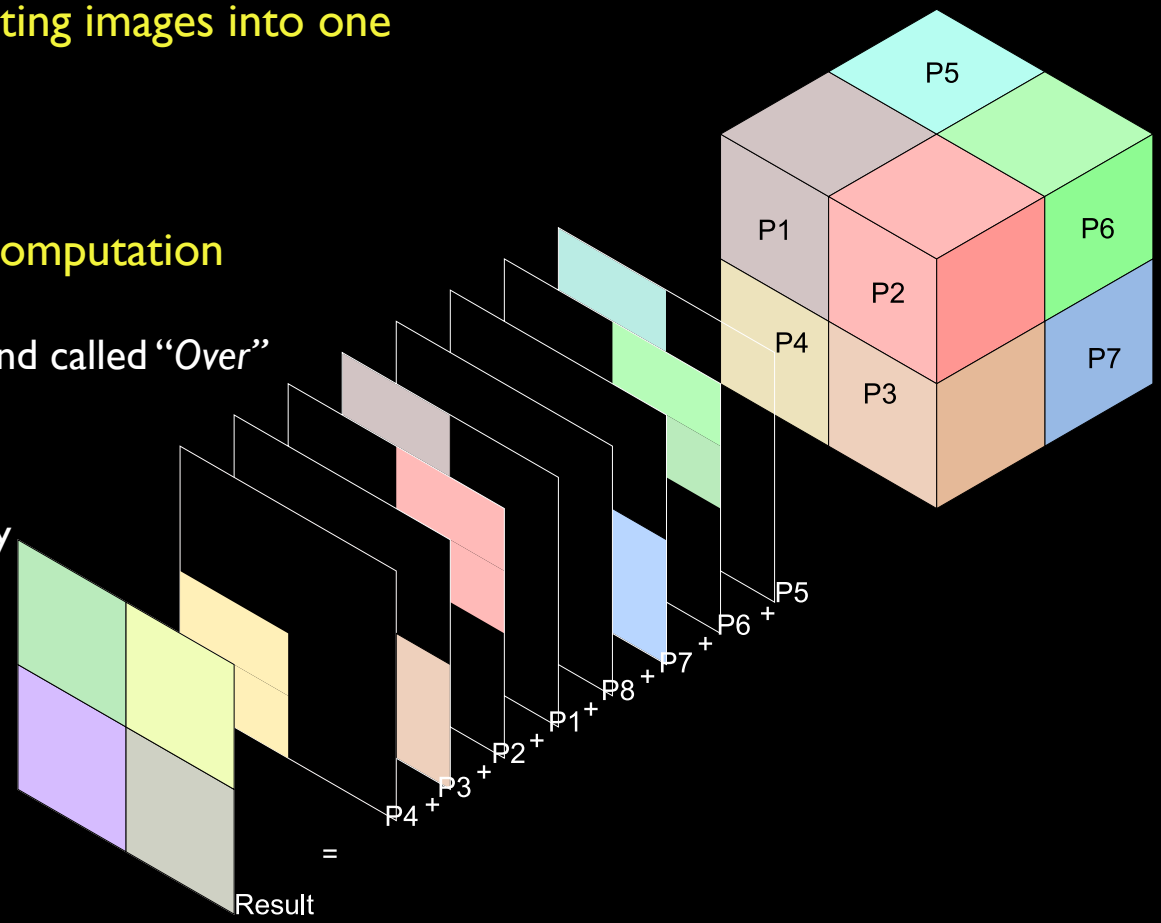
Composition = communication + computation

The computation is usually an alpha-blend called "*Over*"

$$i = (1.0 - \alpha_{old}) * i_{new} + i_{old}$$

$$\alpha = (1.0 - \alpha_{old}) * \alpha_{new} + \alpha_{old}$$

where i = intensity (R,G,B), α = opacity



How the Radix-k Algorithm Works

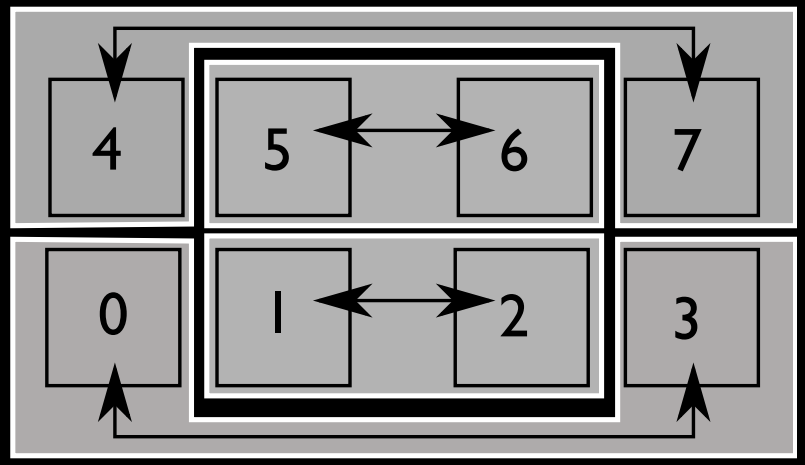
-Increase Concurrency: More participants per group than binary swap (k > 2)

-Manage contention: limiting k value (k < p)

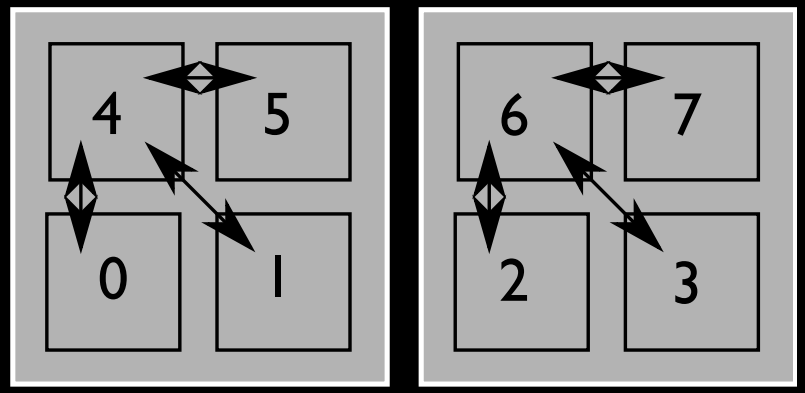
-Overlap communication with computation: nonblocking and careful ordering of operation

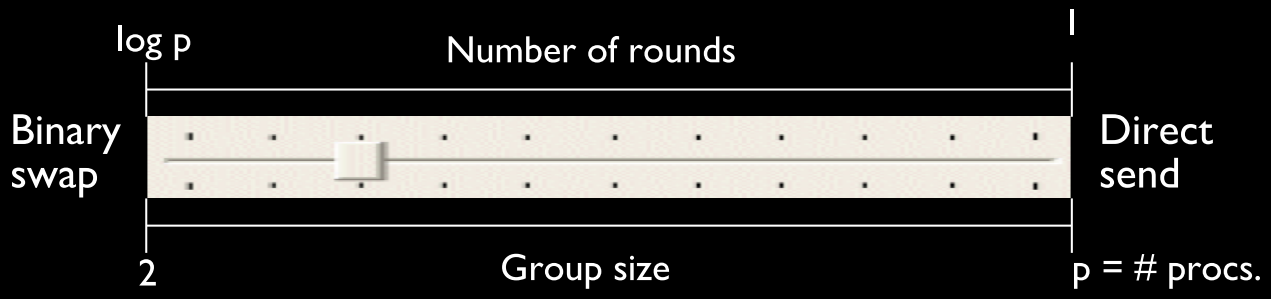
-No penalty for non-powers-of two numbers of processes: inherent in the algorithm design

Round 1
k = 2



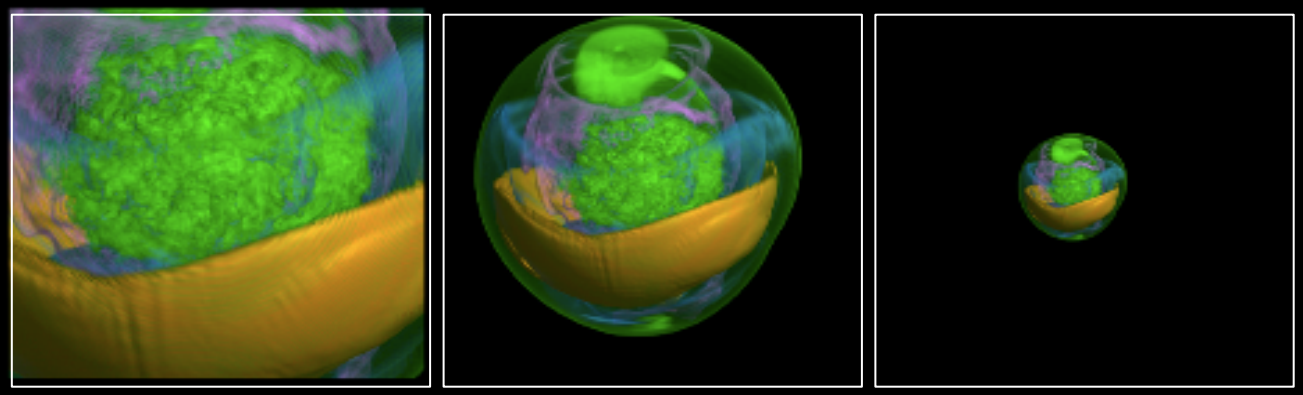
Round 0
k = 4



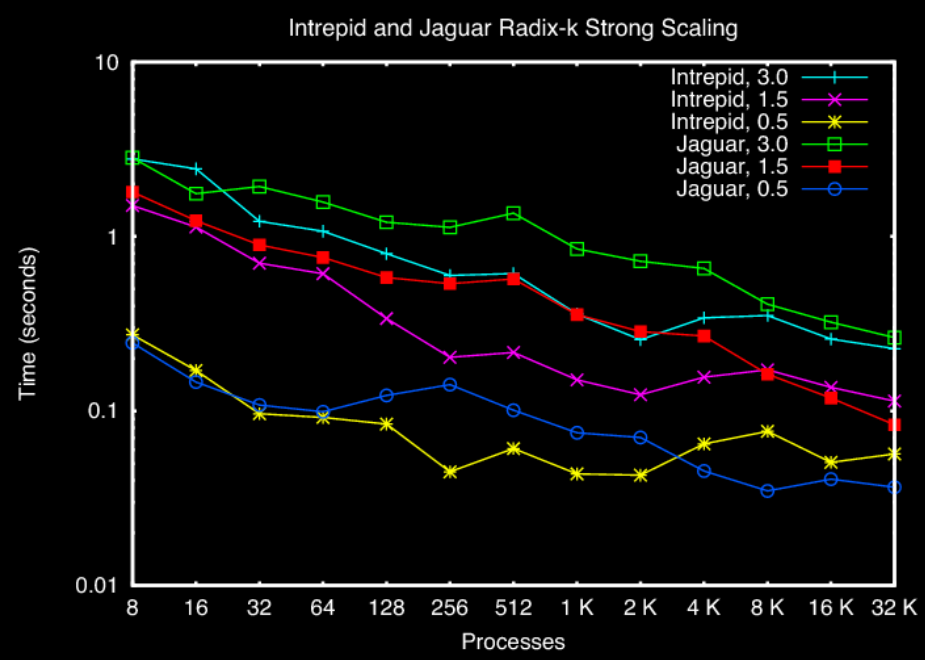


Radix-k Parallel Image Compositing at Scale

zoom = 3.0 zoom = 1.5 zoom = 0.5



Examples of volume rendering at the 3 zoom levels shown below

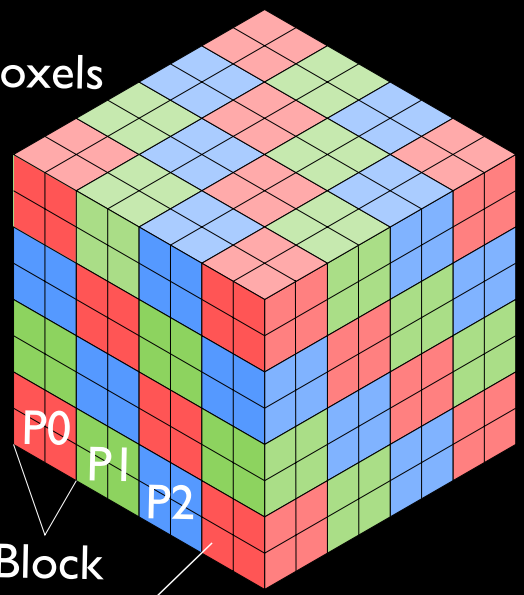


3X – 6X improvement over optimized binary swap (with bounding boxes and RLE) in many cases. 64Mpix at 32K processes can be composited at .08 s, or 12.5 fps.

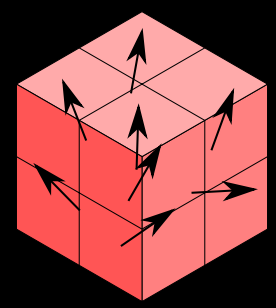
Accelerating and Benchmarking Radix-k Image Compositing at Large-Scale. Kendall et al., EGPGV'10

Case Study: Parallel Particle Tracing

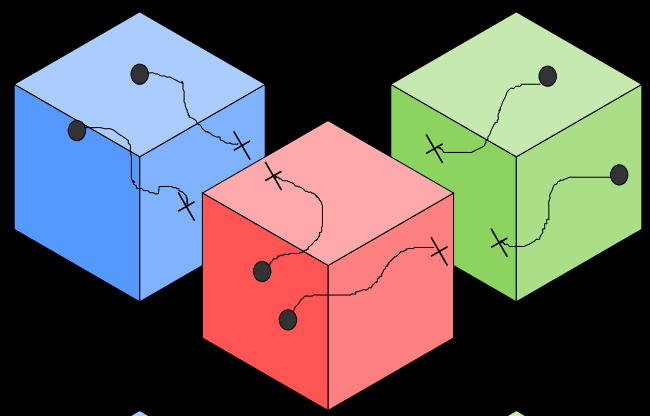
8^3 = 512 voxels
64 blocks
3 Processes



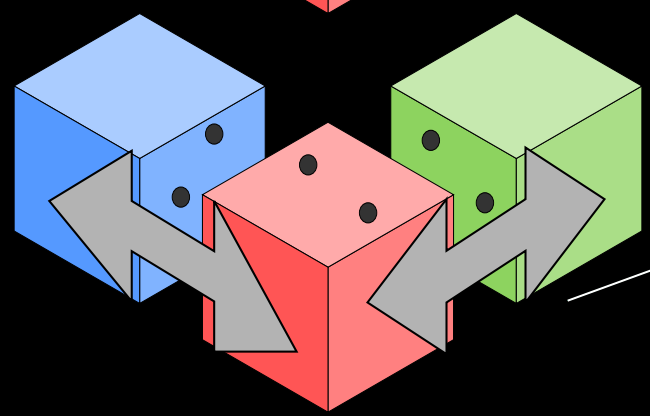
1. Group data into blocks and assign blocks to processors.



2. Each voxel contains a velocity vector



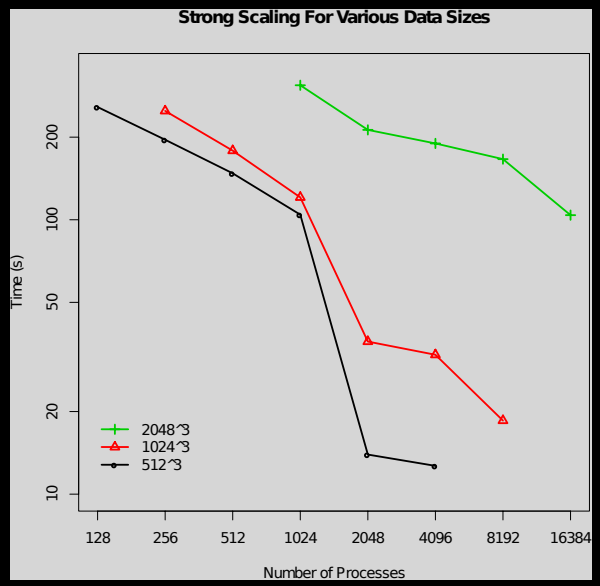
3. Advect particles along velocity vectors.

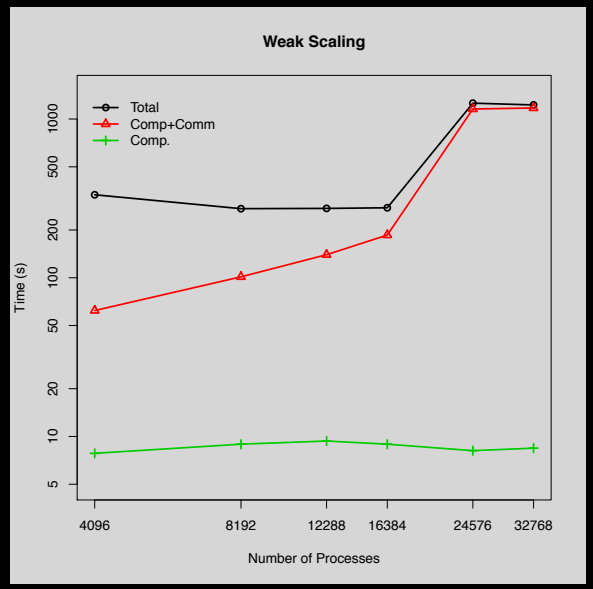


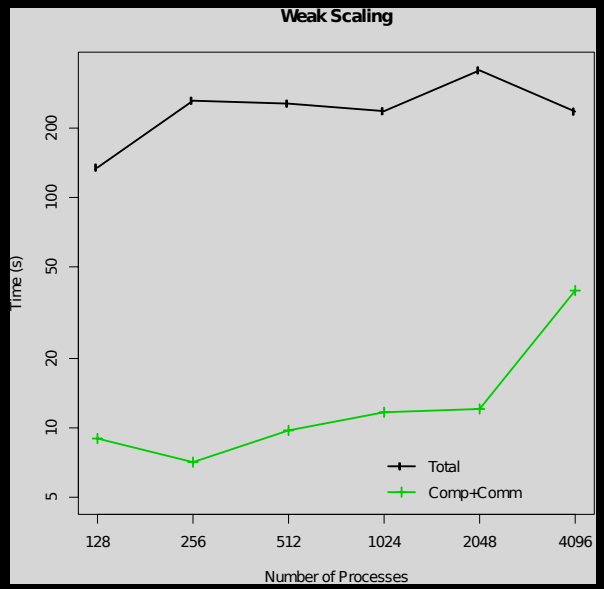
4. Exchange particles among processes when they reach the block boundary.

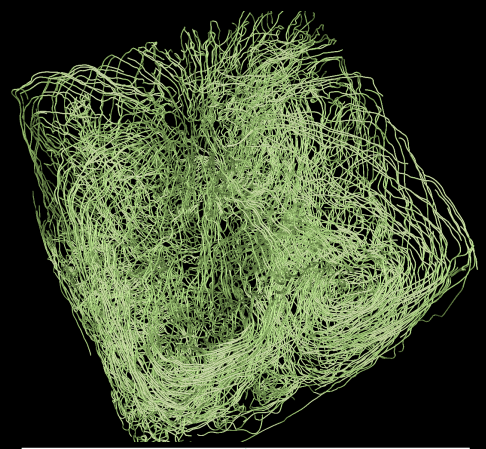
5. Repeat 3, 4

Parallel Particle Tracing Performance

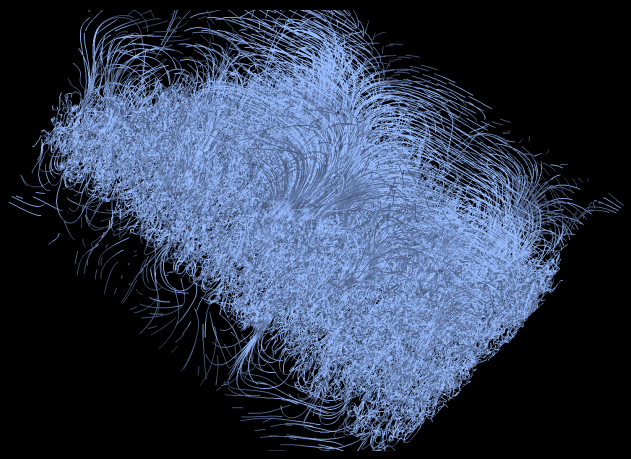




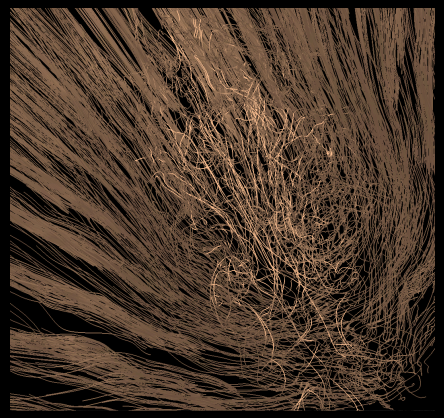




Thermal hydraulics data courtesy Aleks Obabko and Paul Fischer, ANL



Rayleigh-Taylor instability data courtesy Mark Petersen and Daniel Livescu, LANL



Flame stabilization data courtesy Ray Grout, NREL and Jackie Chen, SNL

Putting the Pieces Together: Building Blocks for Developing Scalable Parallel Analysis

Problem:

-Large data -> scalable, parallel analysis

-Analysis is *custom*

-Large data analysis has tough initial barriers

-Lack of resources

-Steep parallel learning curve

Achieve scalability through a library of core data movement components that:

-Balance load (computation, communication)

-Minimize / optimize data movement (storage and network)

-Hide data movement (overlap with work)

Benefits:

-Researchers can study new algorithms

-Computer / computational scientists can build custom applications

-Reuse core components

DIY Data movement components:

-Partitioning

- -Round robin
- -Graph
- -Repartitioning

-Parallel I/O

- -Input datasets
- -Output analysis

-Global reduction

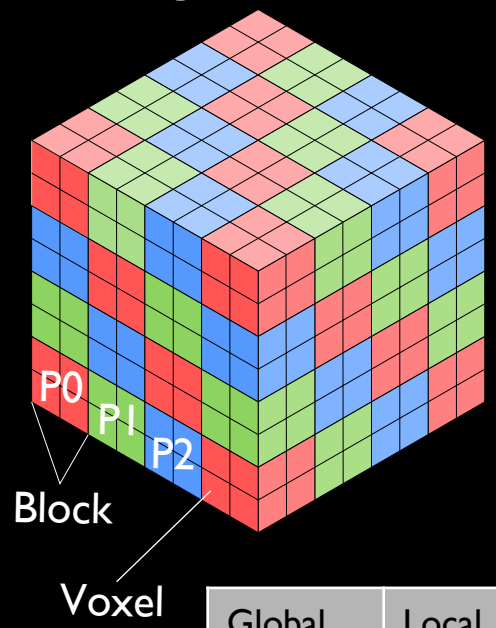
- -Merging
- -Compositing

-Local nearest-neighbor exchange

- -Particle tracing
- -Ghost cell exchange
- -Component labeling

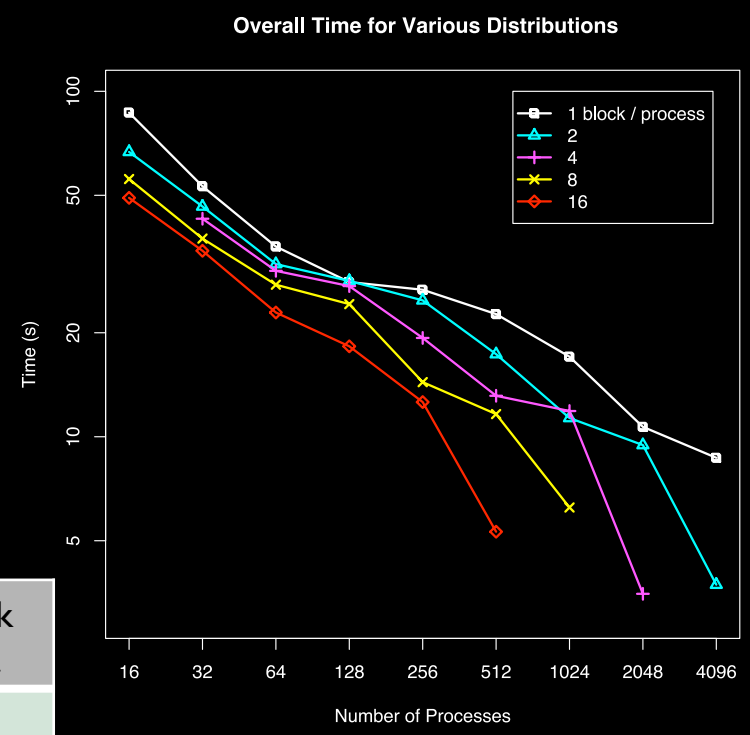
Partitioning and Repartitioning: Eg. Round Robin Assignment

1. Initial Partition
2. Compute, determine balance metric
3. Repartition to optimize metric



Partition data structure: Maintain local data only, not a global table of the partition. Do not want O(total data size) or O(total system size) memory use.

Global Block ID	Local Block ID	Block Data
0		
1	0	X,Y,Z
2		
3		
4	1	X,Y,Z
5		
6		
7	2	X,Y,Z

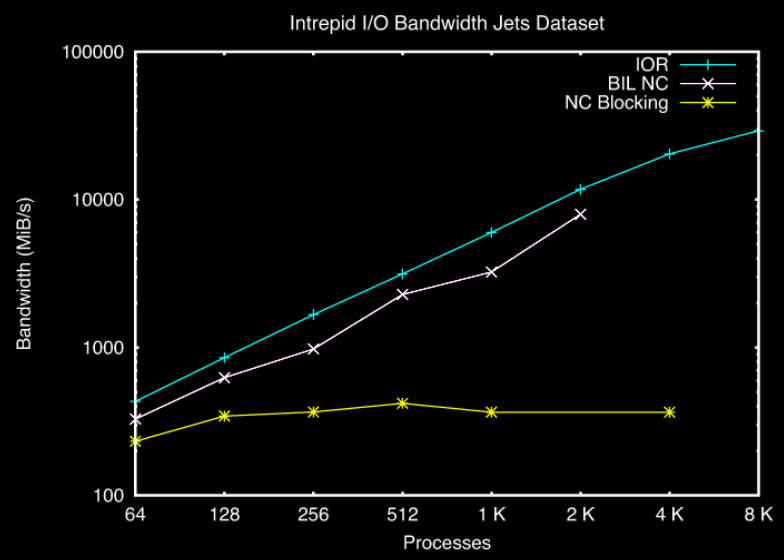


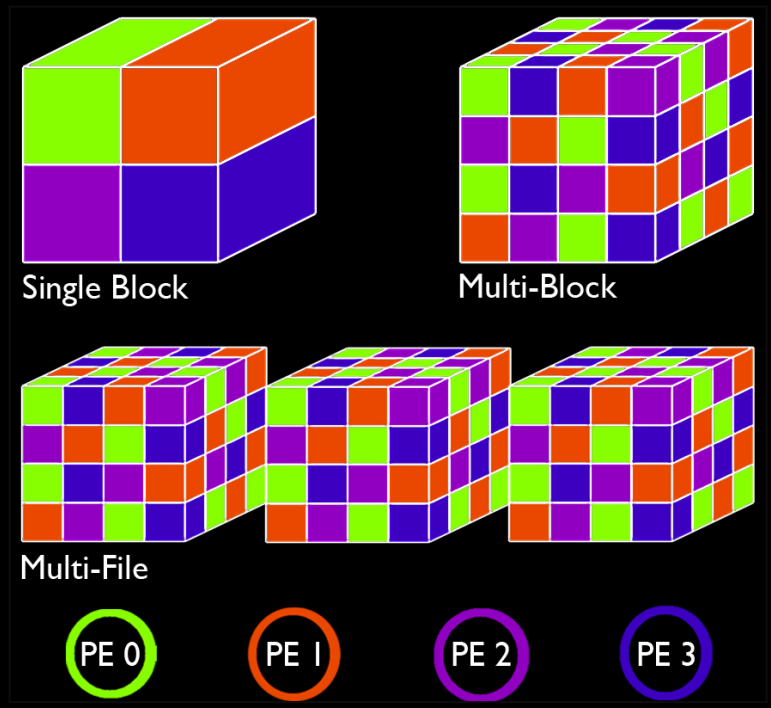
Particle tracing with 1, 2, 4, 8, and 16 blocks per process. A larger number of smaller blocks is better, to a limit.

Parallel I/O: BIL – The Block I/O Layer

Courtesy of Wesley Kendall, University of Tennessee, Knoxville

I/O patterns in analysis/visualization often revolve around block-oriented patterns. BIL abstracts these patterns across files and variables in raw, netCDF, and HDF formats.





Multifile test: BIL runs at 75% of the IOR Benchmark. Scales with number of processes, up to 10X improvement over original MPI-IO implementation.

API:

-BIL_Add_{r,w}block{raw,nc,hdf}(
block_bounds, file, variable, buffer);

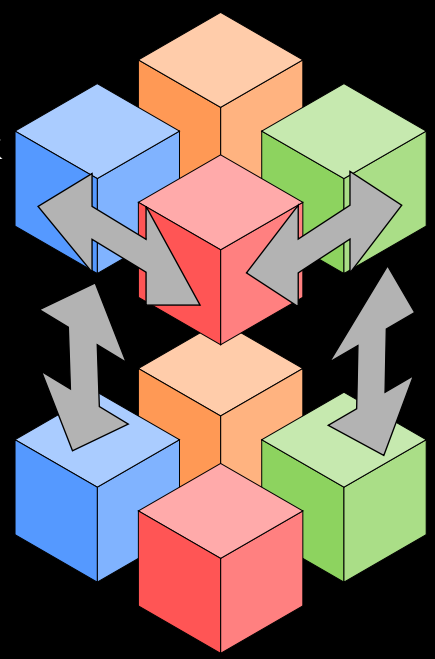
-BIL_{Read,Write}();

Visualization Viewpoint: Towards a General I/O Layer for Parallel Visualization Applications.
Kendall et al., To appear IEEE Computer Graphics and Applications, 2011

Communication Kernels

Global Reduction

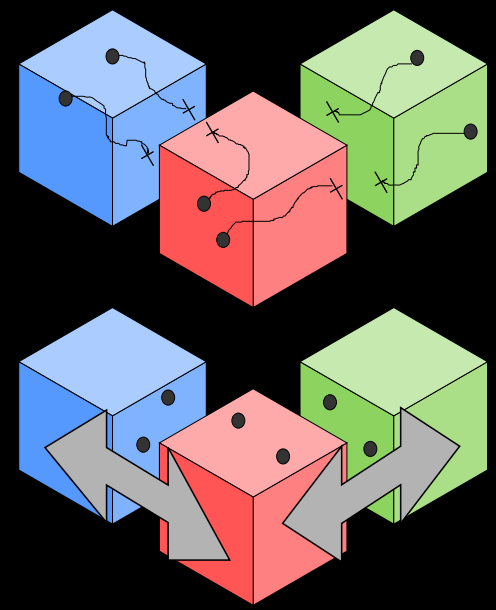
1. Round 1 exchange with k = 4, eg.
2. Round 2 exchange with k = 2, eg.
3. Repeat for as many rounds as desired (may be partial merge)



Parameters:
Number of rounds, k-values per round, swap or transfer, gather to root or parallel output, complete or partial merge

Local Nearest Neighbor Exchange

1. Perform local computations on blocks
2. Exchange objects among processes when they reach the block boundary.
3. Repeat



Parameters:
Number of rounds, terminating criteria, degree of synchony in communication

Ongoing Work: Information-Theoretic Analysis

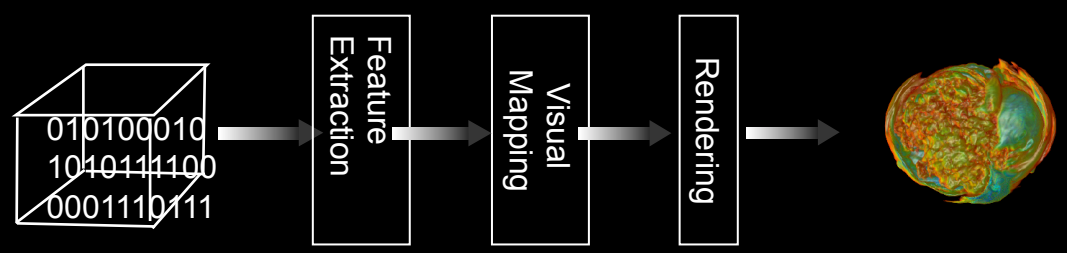
Collaboration with the Ohio State University and New York University Polytechnic Institute

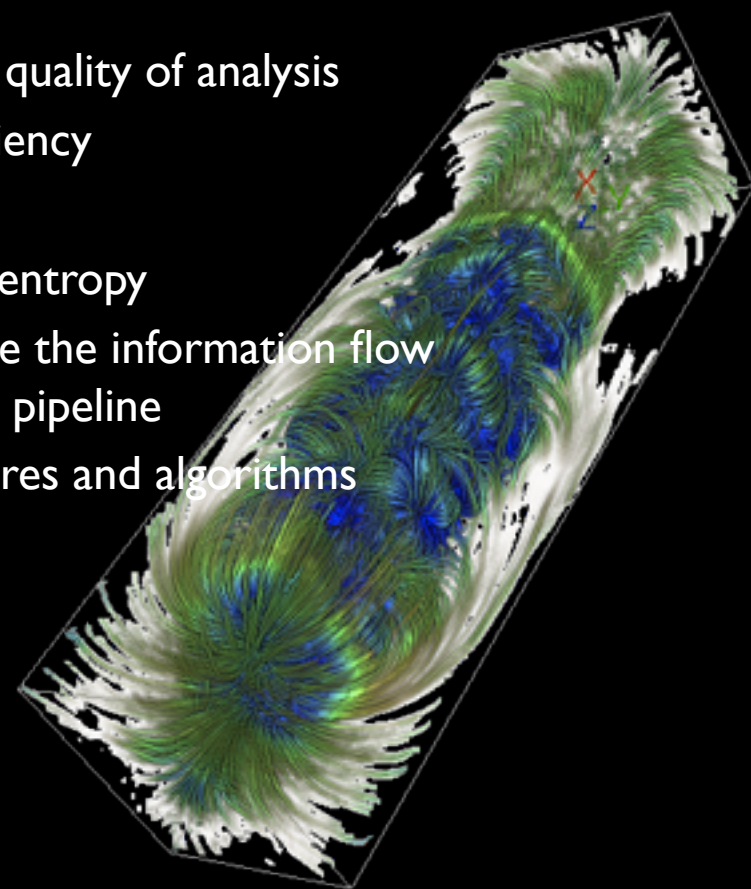
Objective

- Decide what data are the most essential for analysis
- Transform data into effective representations/visualizations that rapidly convey the most insight
- Minimize the information losses and maximize the quality of analysis
- Steer the analysis of data based on information saliency

An Information-theoretic approach

- Quantify Information content based on Shannon's entropy
- Create a quantitative data analysis model to analyze the information flow across the entire data analysis and visualization pipeline
- Use this model to design new analysis data structures and algorithms



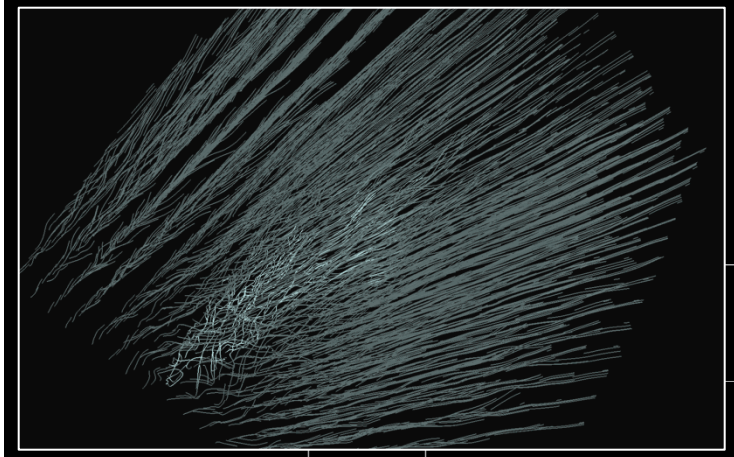


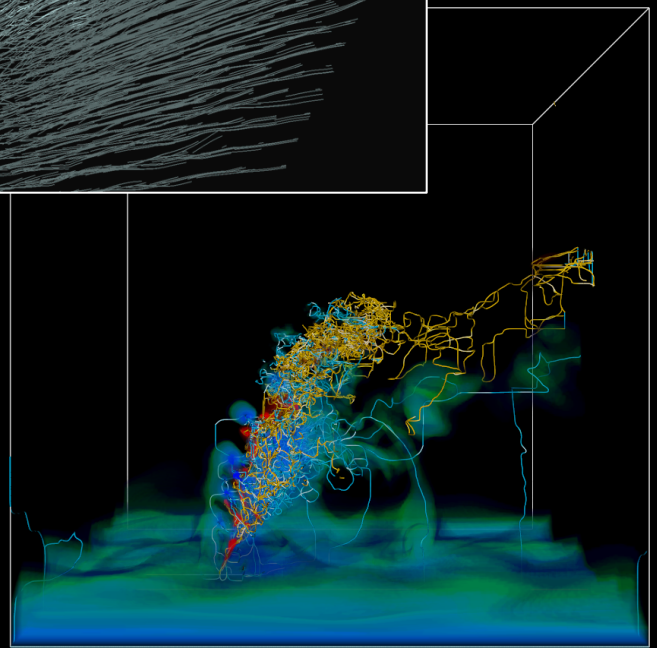
Images courtesy Han-Wei Shen, the Ohio State University

Ongoing Work: Parallel Topological Analysis

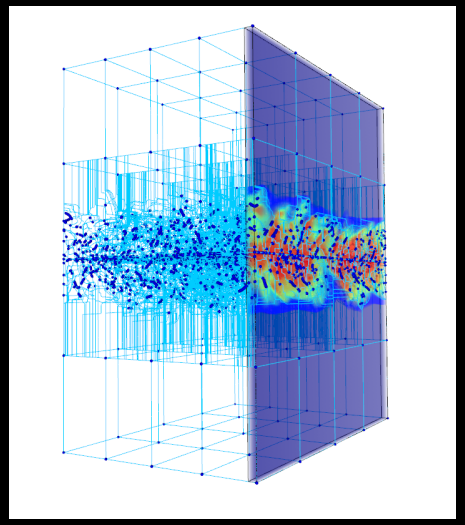
Collaboration with SCI Institute, University of Utah

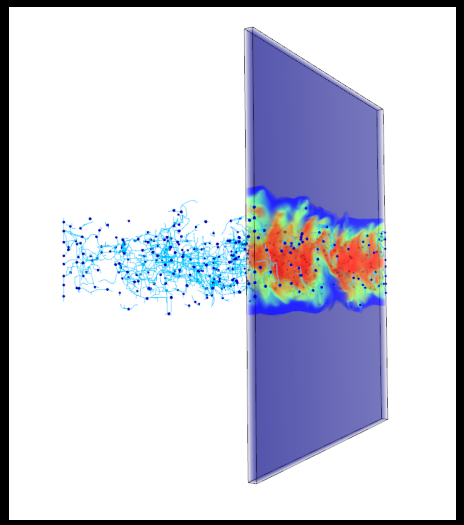
-Transform discrete scalar field into Morse-Smale complex
-Nodes are minima, maxima, saddle points of scalar values
-Arcs represent constant-sign gradient flow
-Used to quickly see topological structure
-Never parallelized before; we scaled to 32K nodes





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





Streamlines and Morse-Smale complex in turbulent region in flame stabilization

Combustion data courtesy Jackie Chen (SNL) and Ray Grout (NREL). Generated by the S3D combustion code.

Image courtesy Attila Gyulassy, University of Utah

Ongoing Work: Geometric Analysis

SciDAC-e collaboration with EVL, University of Illinois at Chicago & CNM, MSD

Material interfaces are key to energy breakthroughs.

Current analysis, visualization, and display methods are inadequate for complex multiscale materials science.

Nanobowls are nanoscale bowl-shaped aluminum oxide structures designed to trap catalysts. Scientists model these structures under different conditions and use visual analysis to determine whether they are stable.

Material interfaces require quantitative and visual analysis.

A unified volumetric representation for electrostatic material boundaries

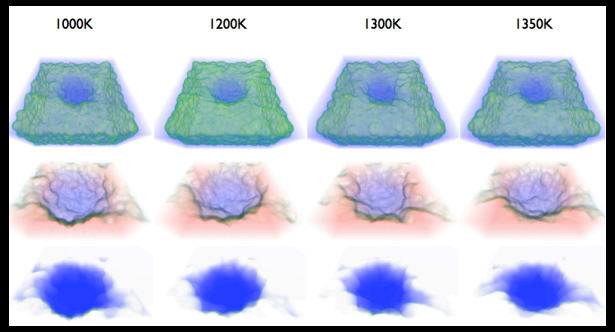
Advanced 3D display technology for improved depth perception

Analysis, 3D visualization, and validation in an end-to-end work environment

Exploration environment is assisting in scientific discovery.

Our solution revealed that the volume of the simulated nanobowl varied over time and temperature.

This result helped guide future simulations and facilitated communication with other scientists.



Evolution of 15 angstrom nanobowls at different temperatures, courtes Aaron Knoll

Custom-built autostereoscopic display for resolving complex structures

Data-Intensive Analysis at the Forefront of Science

Conclusions

-Exascale requires new thinking about analysis
-More analysis must (will) be integrated with simulation
-Scalable analysis is data-intensive: Moving data, transforming data, reducing data, analyzing data, storing data
-Scientists need to take ownership of their own analysis
-Less visual, more analytical analysis at early stages of the science pipeline
-Intelligent data reduction

Ongoing, Future

-Continue developing software infrastructure for scalable analysis in HPC systems

-Continue collaborating with scientists to integrate analysis with applications

"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)

Funding
US DOE SciDAC UltraVis Institute
US DOE SDMAV Exascale Initiative

People
Rob Ross, Han-Wei Shen, Jian Huang, Wes Kendall, Rajeev Thakur, Dave Goodell, Kwan-Liu Ma, Hongfeng Yu

Tom Peterka

tpeterka@mcs.anl.gov

Mathematics and Computer Science Division