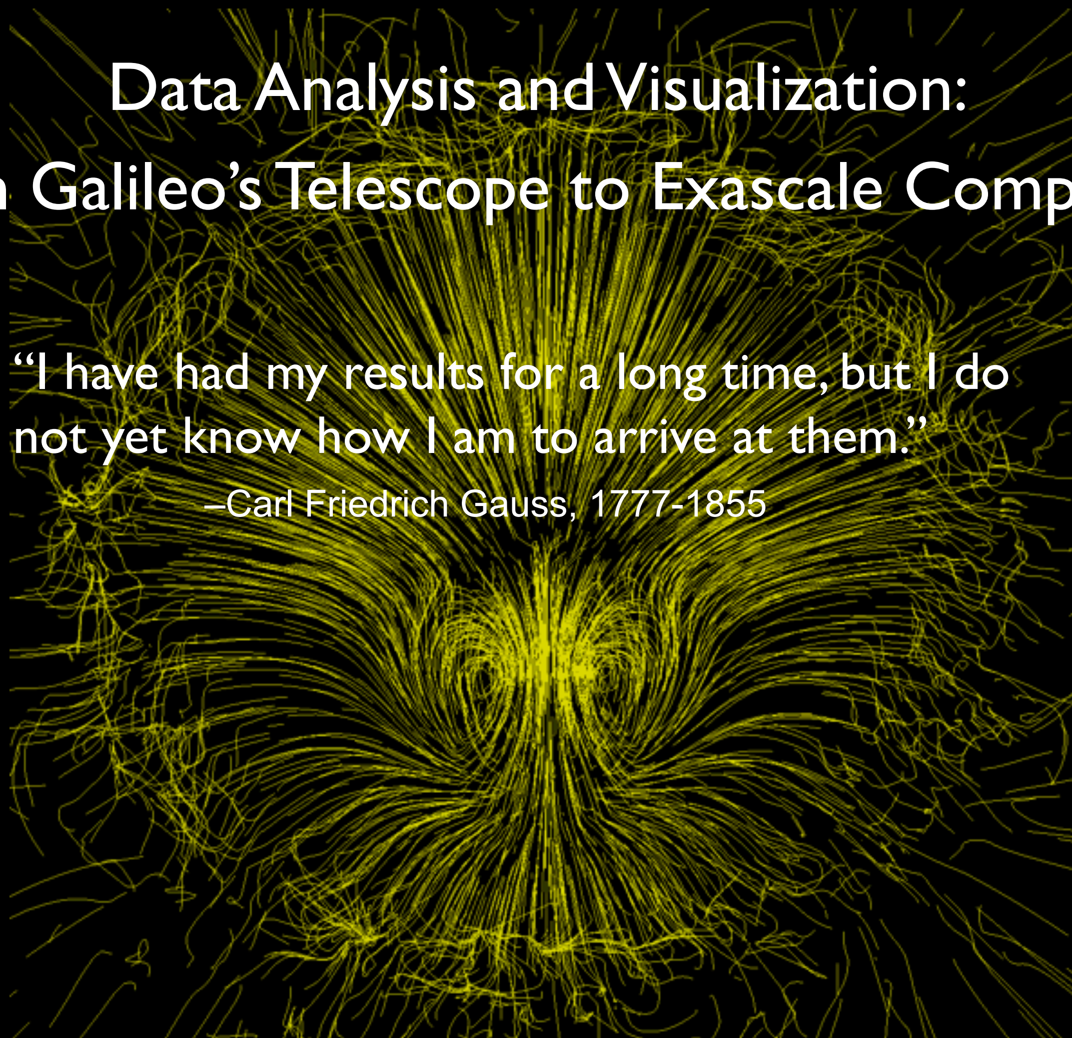
Data Analysis and Visualization:
From Galileo's Telescope to Exascale Computing

"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



Type 1A SN data courtesy Cal Jordan, UofC Flash Center

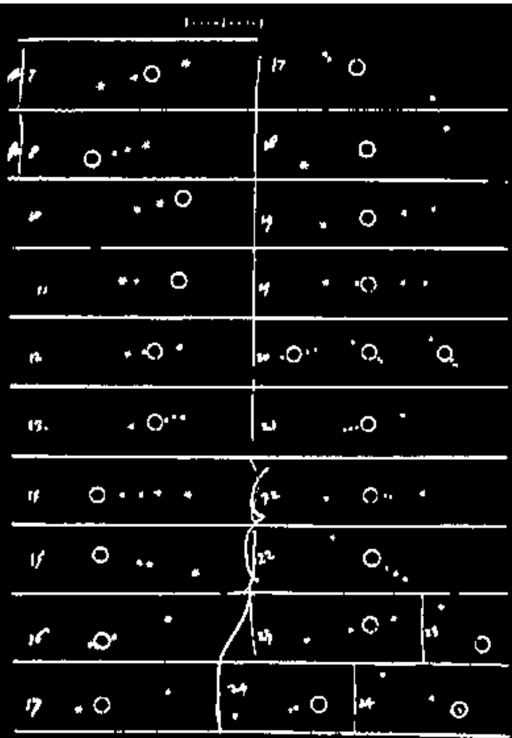
Tom Peterka

tpeterka@mcs.anl.gov

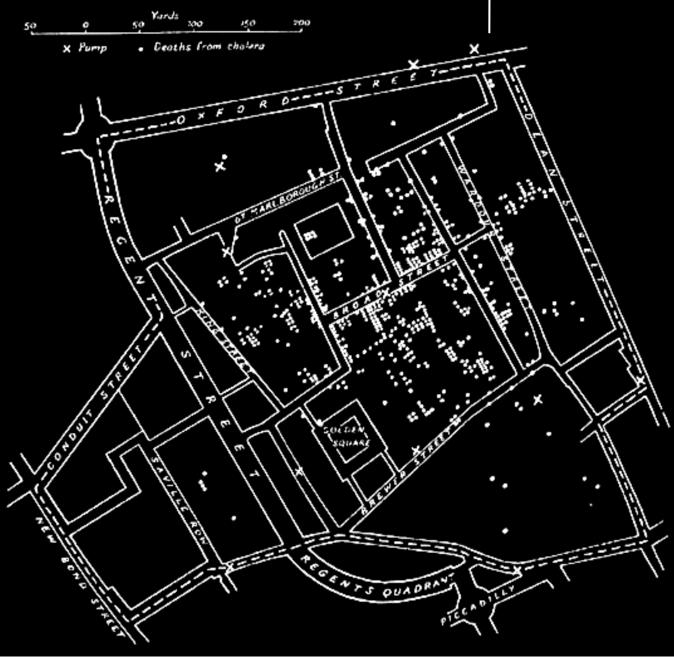
Mathematics and Computer Science Division

Adler Research Division Colloquium 9/13/10

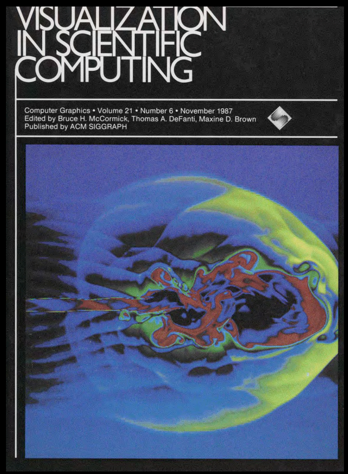
400 Years of Visualization



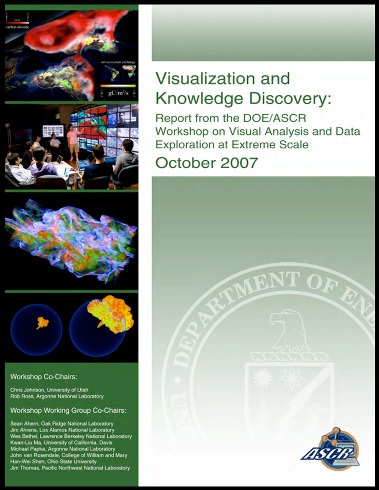
Galileo, 1610



John Snow, 1854

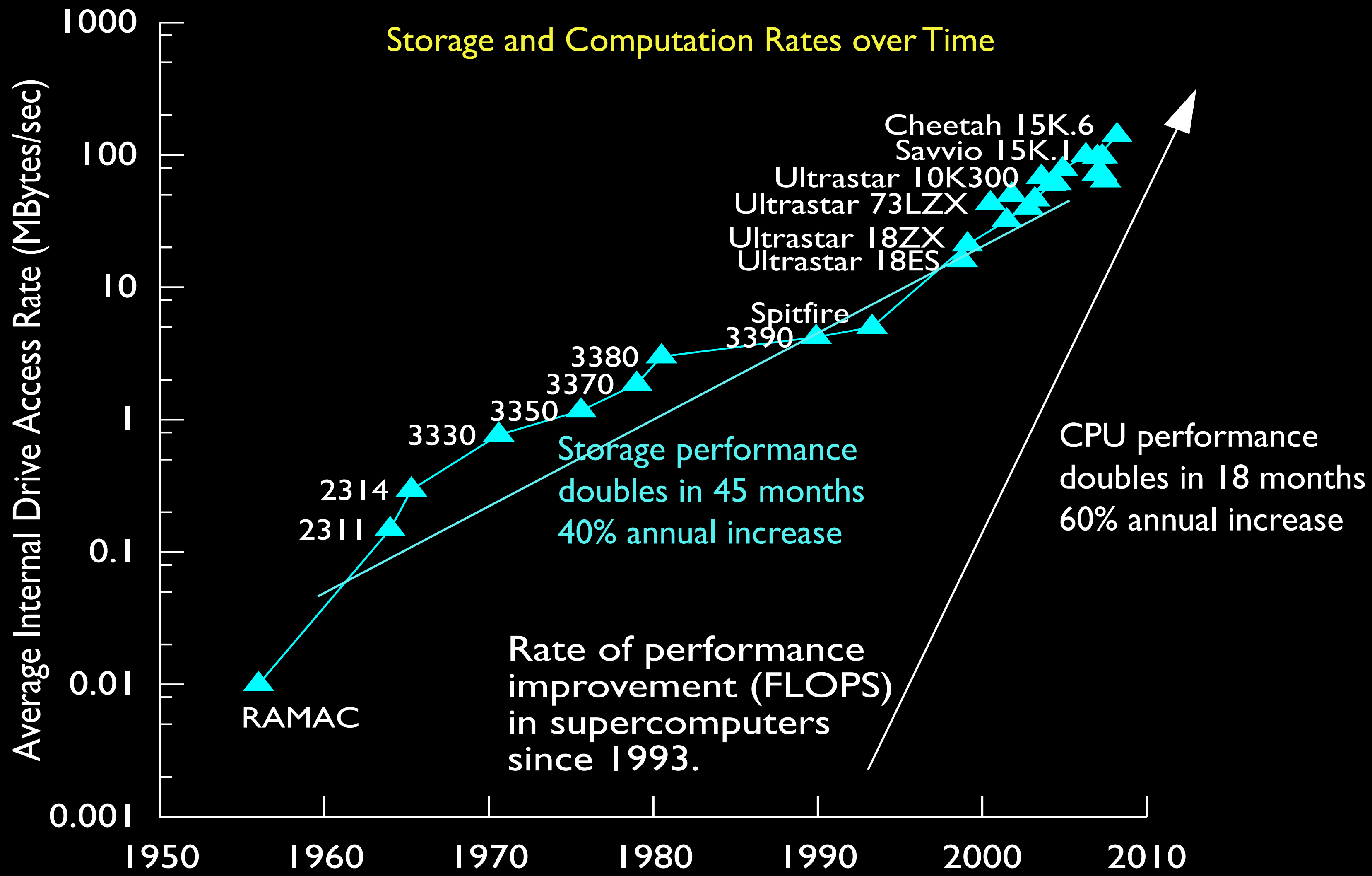


McCormick et al., 1987



Johnson et al., 2007

Supercomputing: A Delicate Balance



“Datasets being produced by experiments and simulations are rapidly outstripping our ability to explore and understand them” –Johnson et al., 2007.

The Data-Intensive Nature of Computing and Analysis

"Analysis and visualization will be limiting factors in gaining insight from exascale data."
–Dongarra et al., International Exascale Software Project Draft Road Map, 2009.

Normalized Storage / Compute Metrics

Machine	FLOPS (Pflop/s)	Storage B/W (GB/s)	Flops per byte stored	Bytes comp. per byte stored
LLNL BG/L	0.6	43	$O(10^4)$	$O(10^3)$
Jaguar XT4	0.3	42	$O(10^4)$	$O(10^3)$
Intrepid BG/P	0.6	50	$O(10^4)$	$O(10^3)$
Roadrunner	1.0	50	$O(10^5)$	$O(10^4)$
Jaguar XT5	1.4	42	$O(10^5)$	$O(10^4)$

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

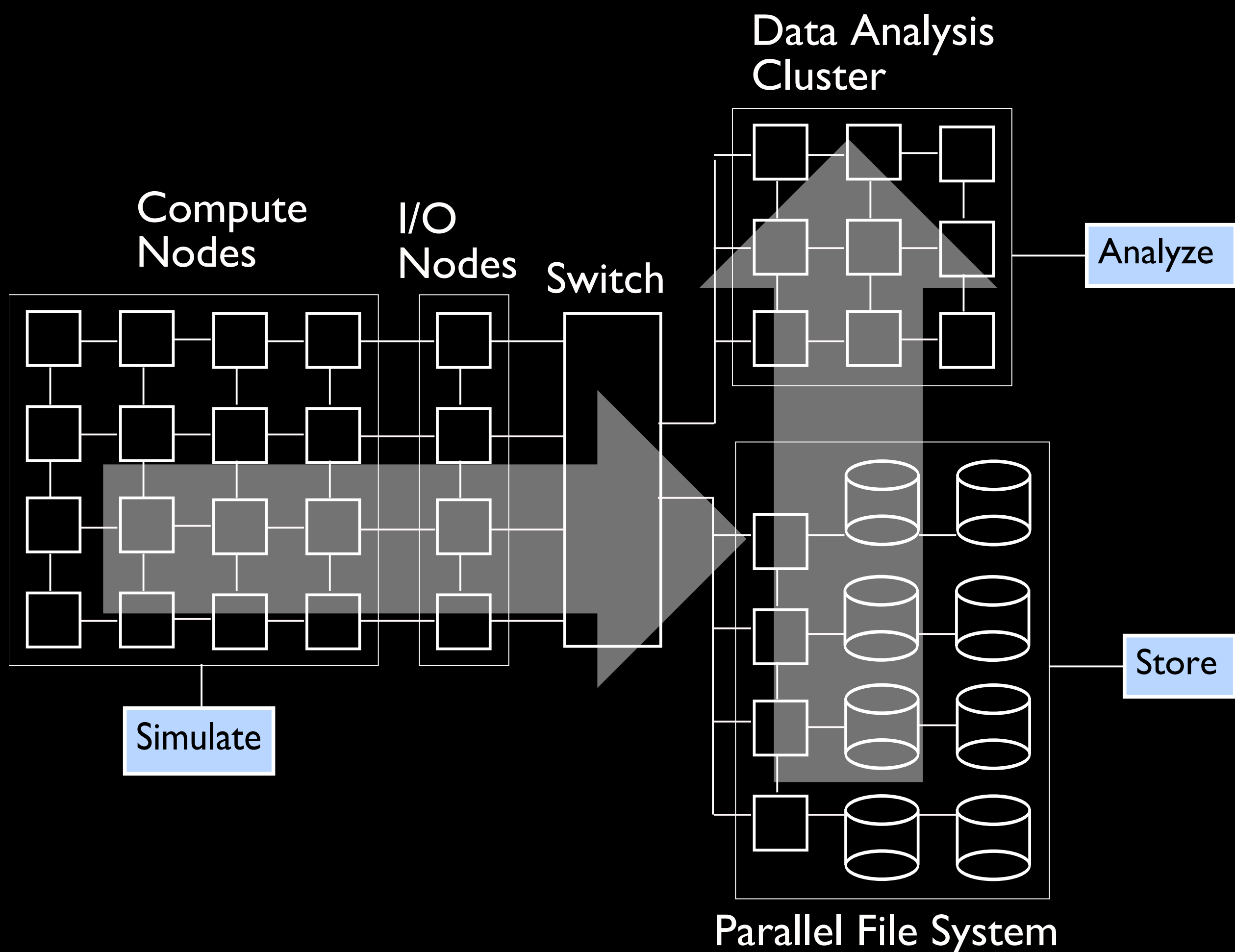
Today: Scientific Data Analysis in HPC Environments

Examples:

2D statistical graphics using R

3D scientific visualization using ParaView

Scientific visualization using VisIt



A linear, sequential pipeline where tasks mapped to architectures in fixed fashion is robust but not necessarily scalable.

Scalable Analysis & Visualization: The Data Parallel Approach

Treat analysis as any other parallel computation

-Decompose the domain

-Assign to processors

-Combine local and global operations

-Measure scaling and efficiency

-Balance load, minimize communication

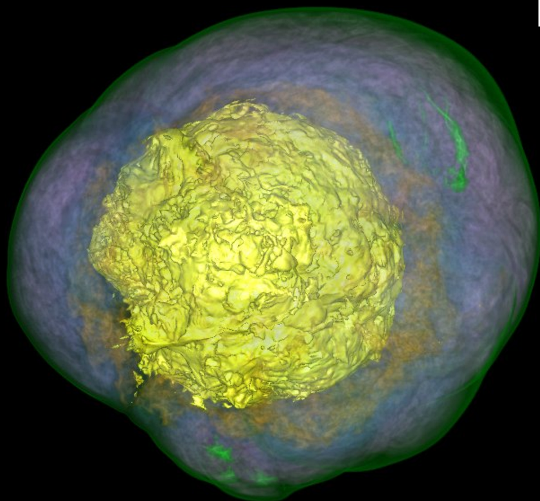
Integrate with simulation

"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."

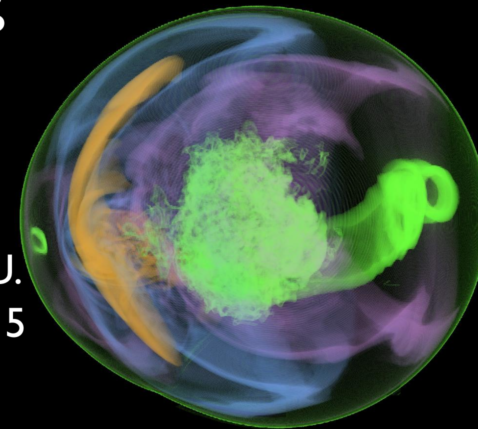
–Lucy Nowell, Scientific Data Management and Analysis at Extreme Scale, Office of Science Program Announcement LAB 10-256, 2010.

Parallel Volume Rendering

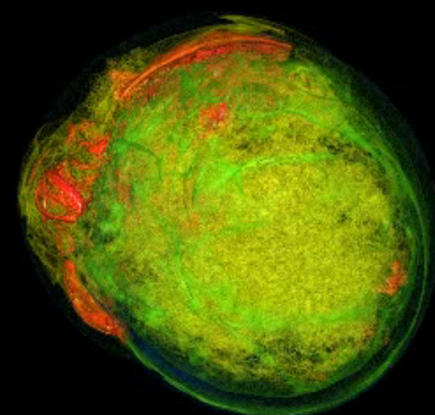
Volume rendering of shock wave formation in core-collapse supernova dataset, courtesy of John Blondin, NCSU. Structured grid of 1120^3 data elements, 5 variables per cell.



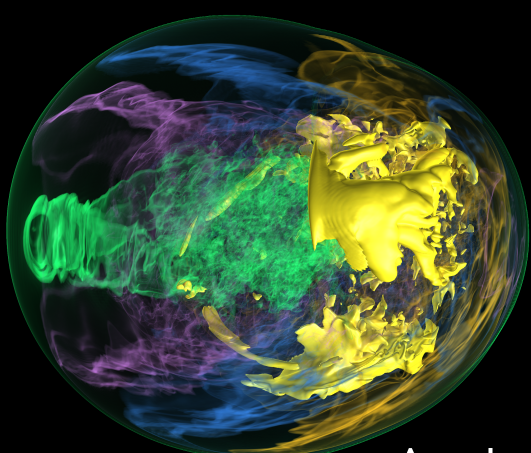
Pressure at time-step 1530



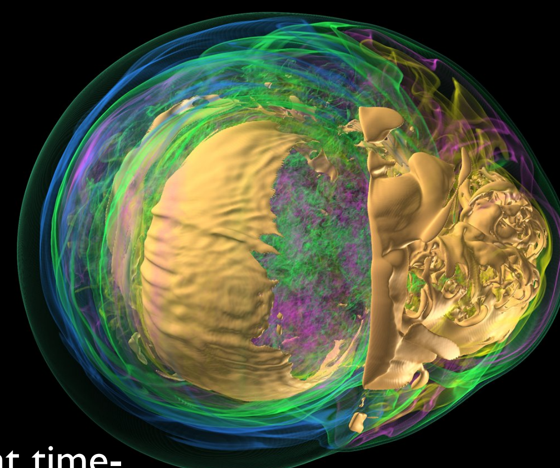
Angular momentum at time-step 1403



Entropy over 100 time-steps



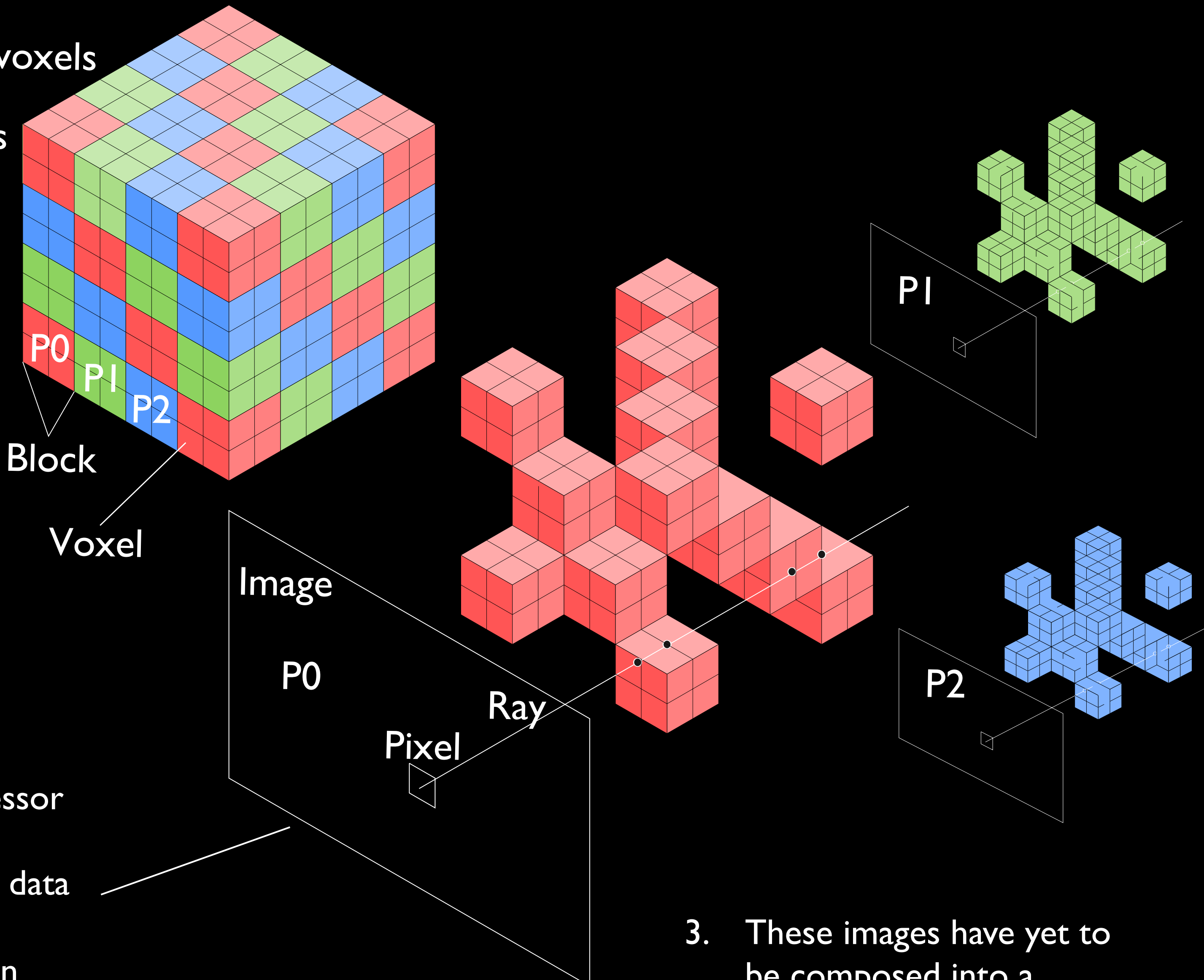
Angular momentum at time-step 1492



Entropy at time-step 1518

Parallel Volume Rendering Refresher

8^3 = 512 voxels
64 blocks
3 Processes

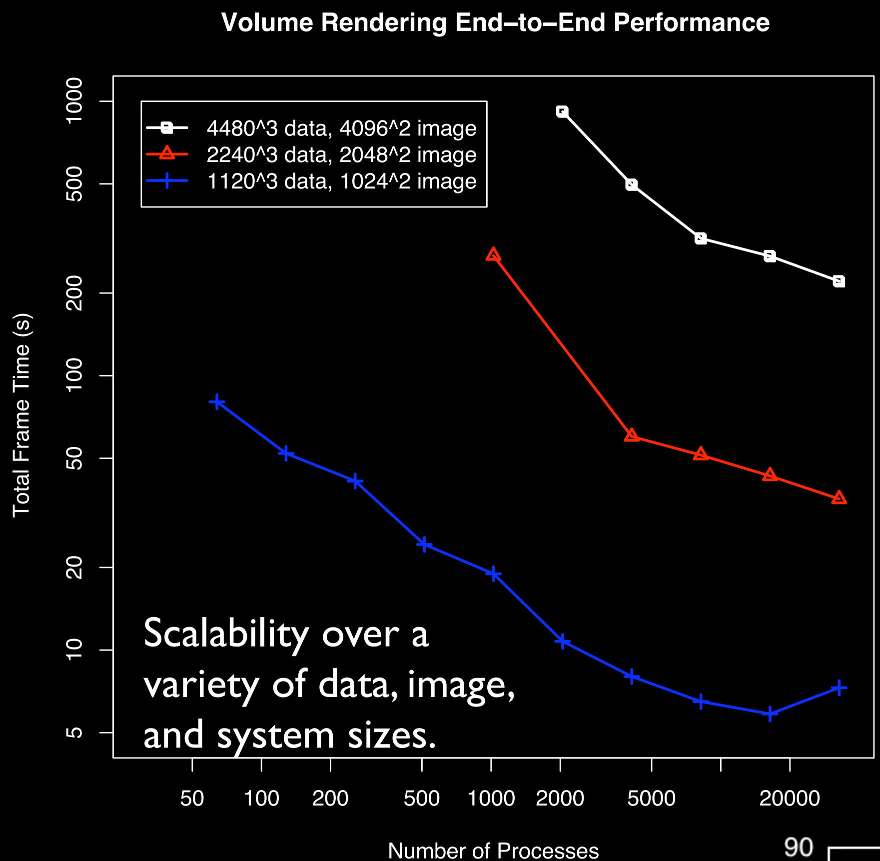


1. Group data into blocks and assign blocks to processors.

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.

Benchmarking Performance

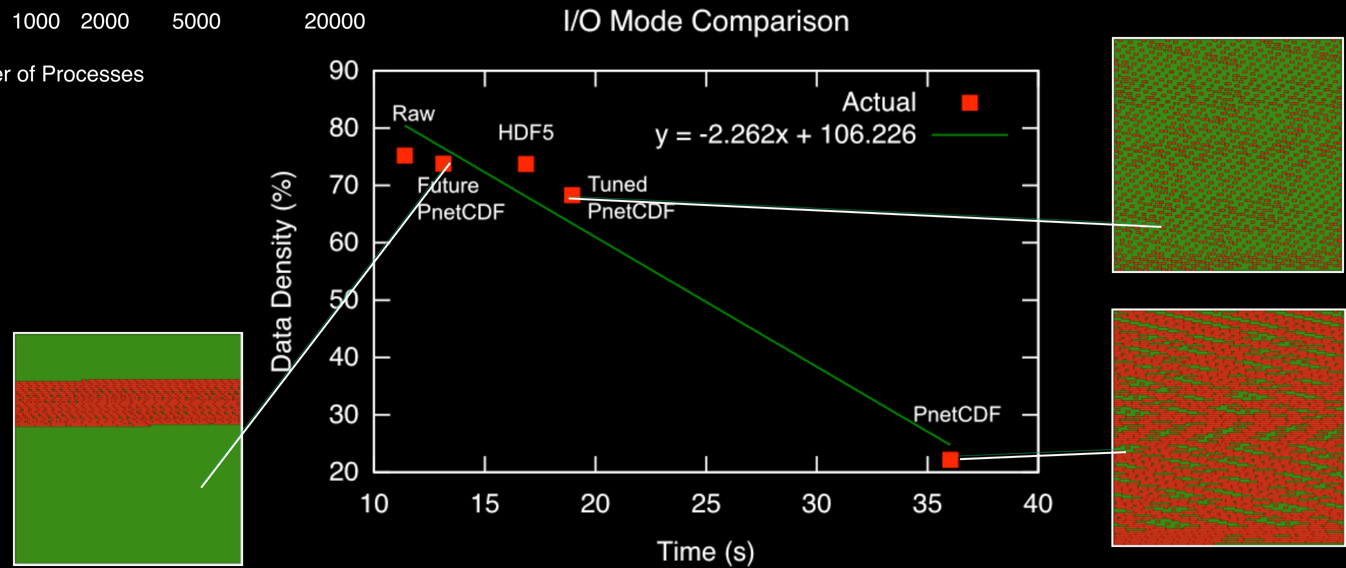


Scalability over a variety of data, image, and system sizes.

Grid Size	Time-step size (GB)	Image size (px)	# Procs	Tot. time (s)	% I/O	Read B/W (GB/s)
2240^3	42	2048^3	8K	51	96	0.9
			16K	43	97	1.0
			32K	35	96	1.3
4480^3	335	4096^3	8K	316	96	1.1
			16K	272	97	1.3
			32K	220	96	1.6

Volume rendering performance at large size is dominated by I/O.

Changing data file layout can improve I/O performance, shown by access pattern signatures and performance data.



End-to-End Study of Parallel Volume Rendering on the IBM Blue Gene/P. ICPP’09.

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

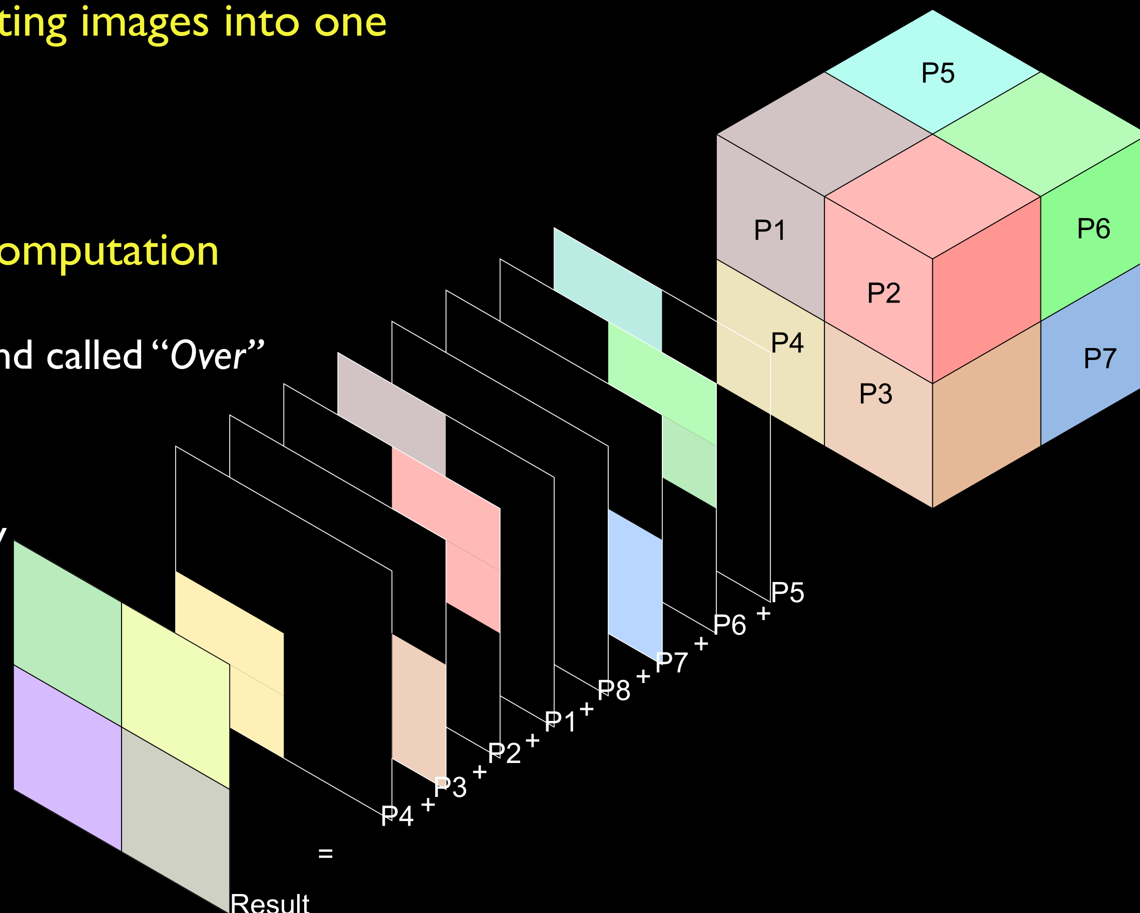
Composition = communication + computation

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

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

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

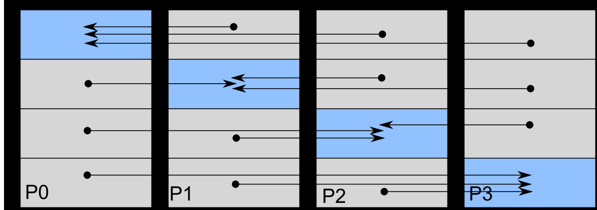
where i = intensity (R,G,B), α = opacity



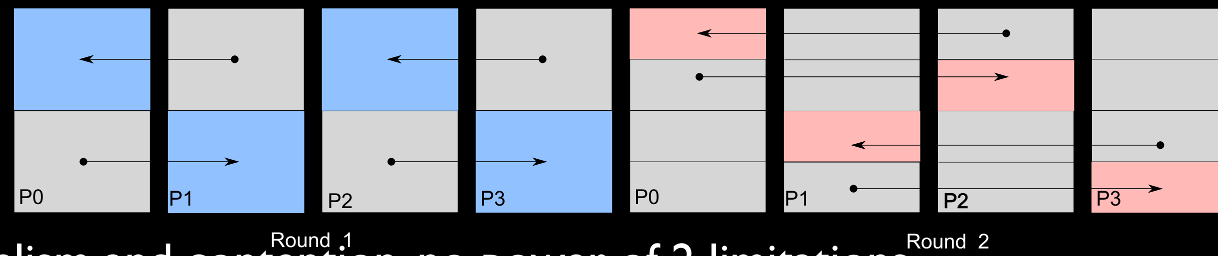
A Configurable Algorithm for Parallel Image-Compositing Applications. Peterka et al., SC09

Direct-Send, Binary Swap, and Radix-k

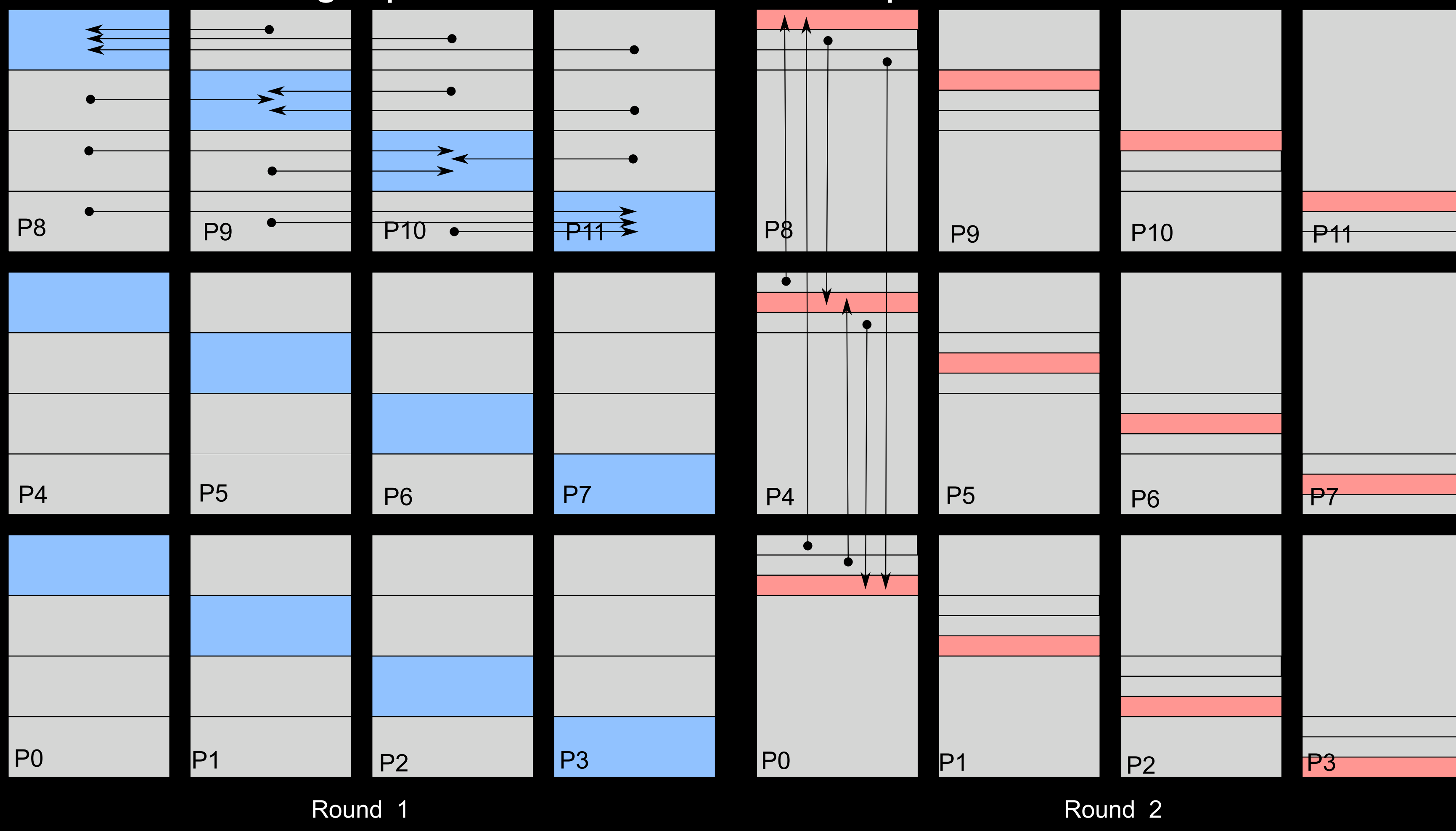
Direct-send: Parallel, contentious



Binary swap: Low parallelism, limited to powers of 2



Radix-k: Managed parallelism and contention, no power of 2 limitations



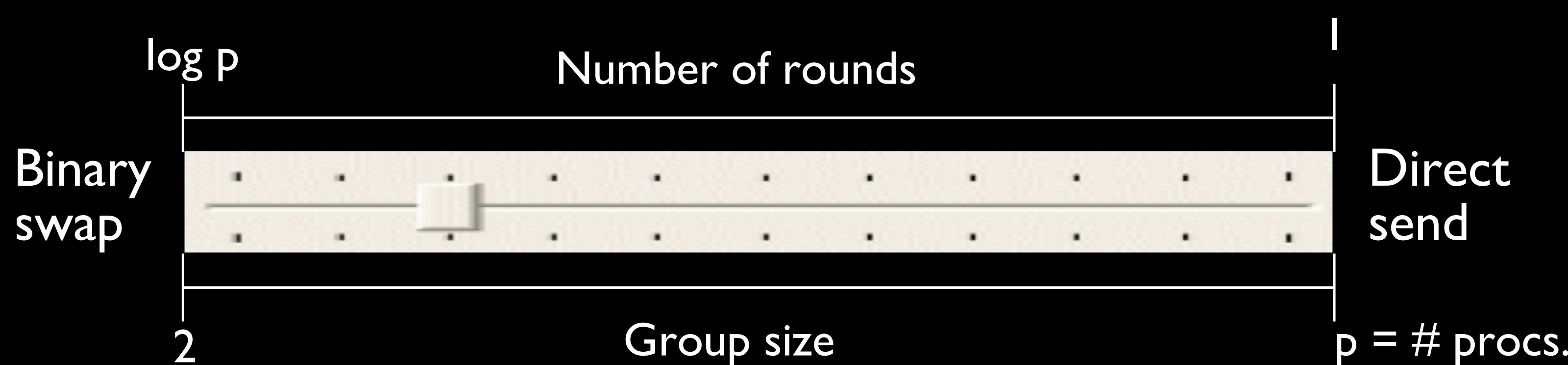
Radix-k: Configurable to Different Architectures

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

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

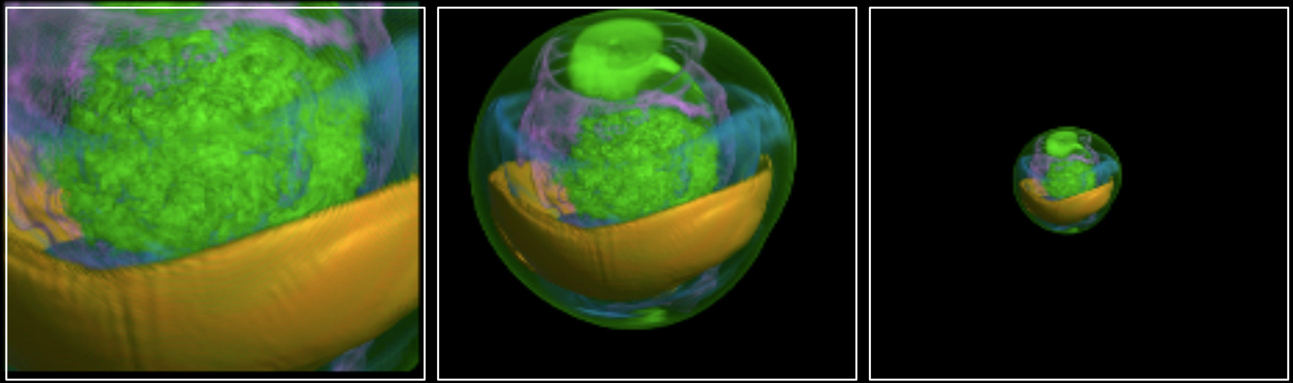
-Overlap communication with computation: nonblocking and careful ordering of operation

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

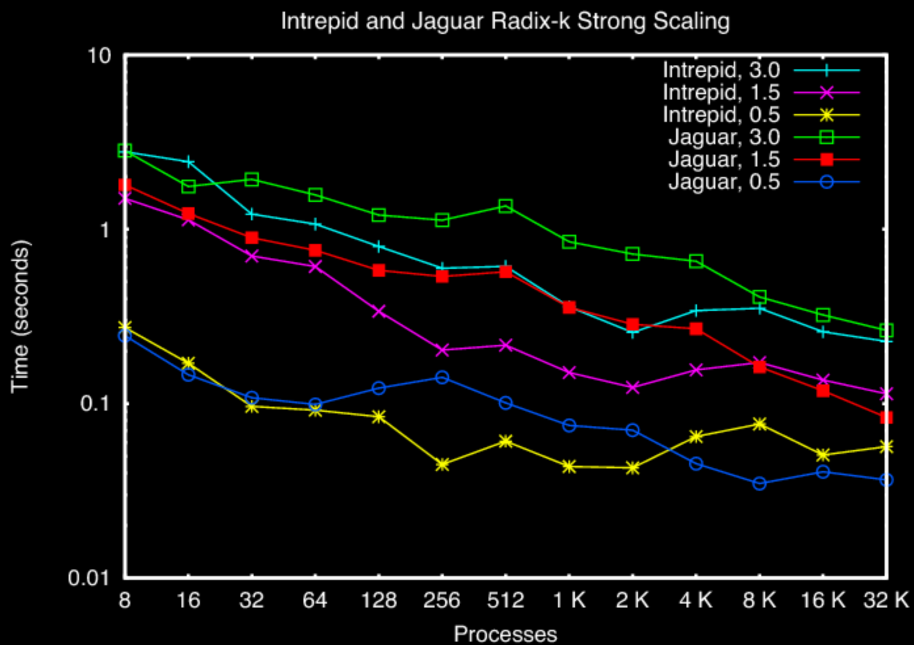


Radix-k at Scale

zoom = 3.0 zoom = 1.5 zoom = 0.5



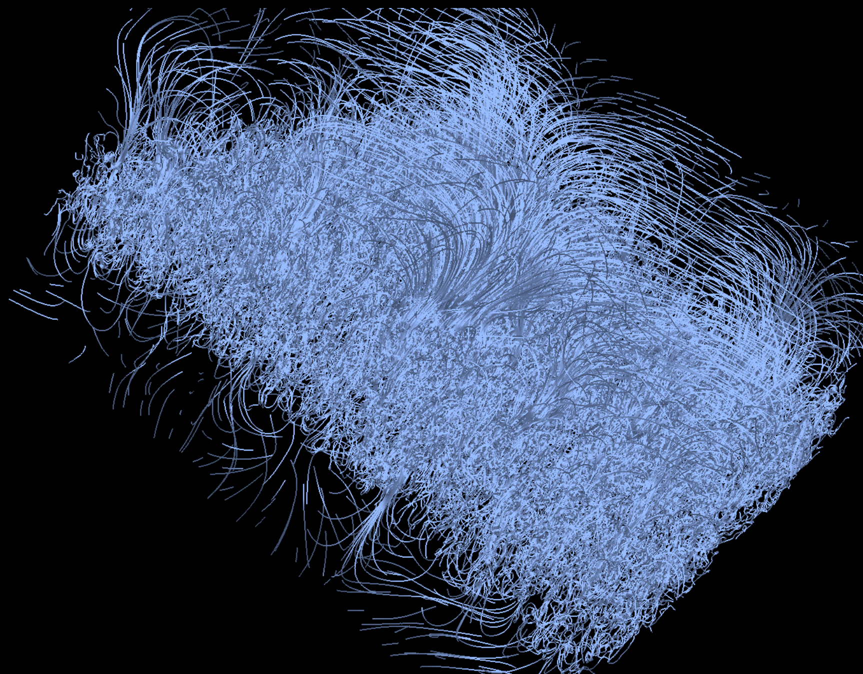
Examples of volume rendering at the 3 zoom levels shown below



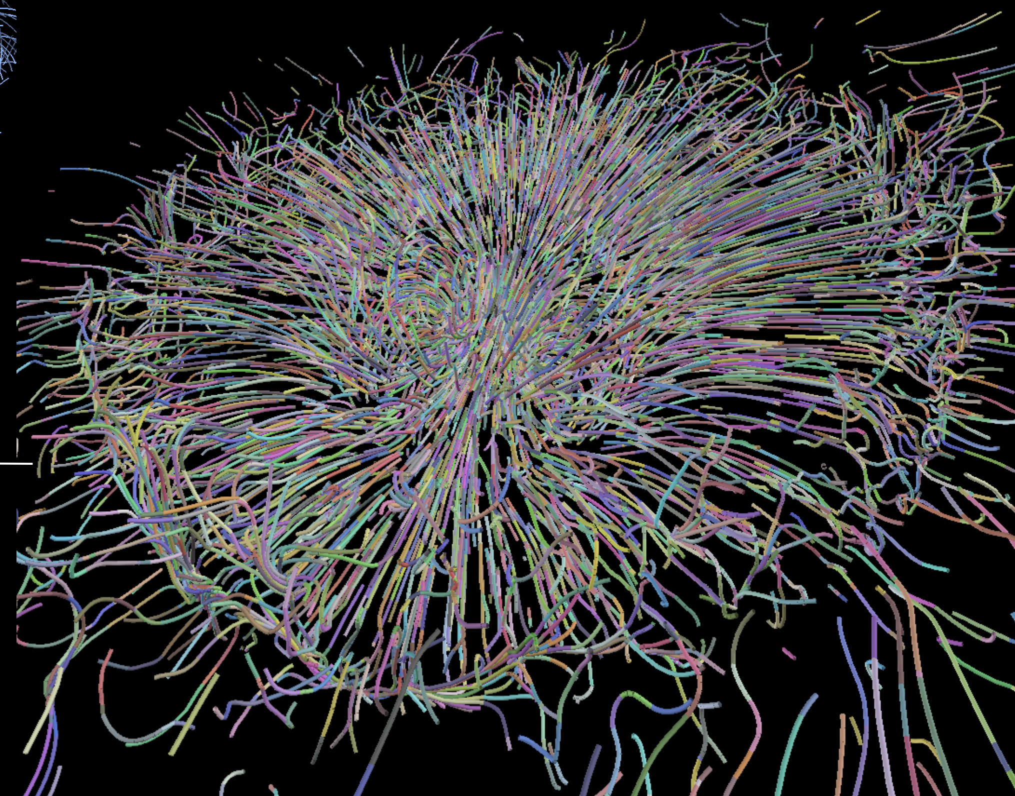
3X – 6X improvement over optimized binary swap (with bounding boxes and RLE) in many cases. 64Mpix at 32K processes can be composited at .08 s, or 12.5 fps.

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

Large-Scale Parallel Particle Tracing



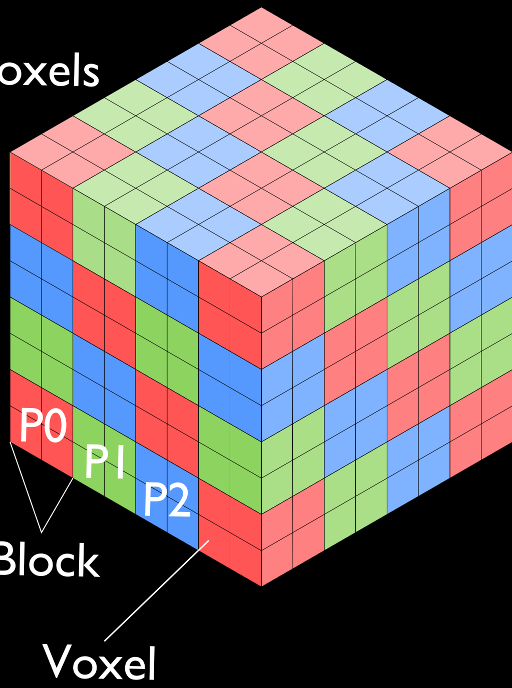
Rayleigh-Taylor instability data courtesy
Mark Petersen, Daniel Livescu, LANL



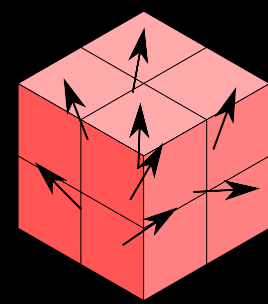
Type IA supernova data courtesy George
Jordan, UofC FLASH Center

Parallel Particle Tracing Crash Course

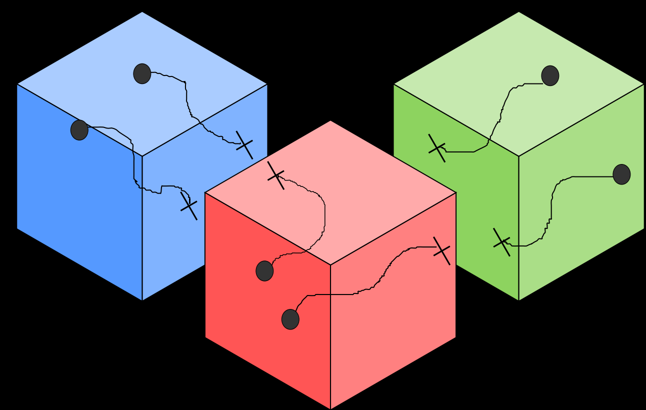
8^3 = 512 voxels
64 blocks
3 Processes



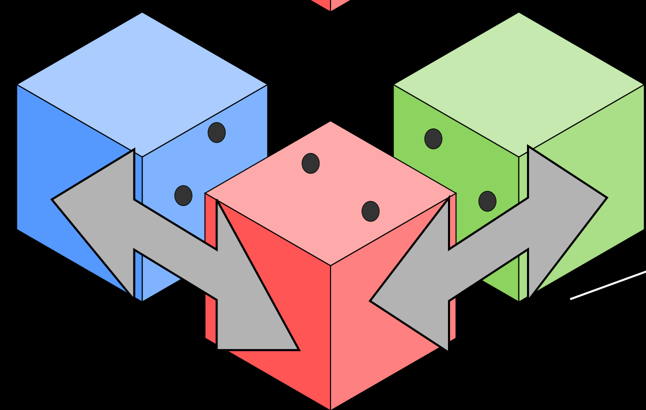
1. Group data into blocks and assign blocks to processors.



2. Each voxel contains a velocity vector



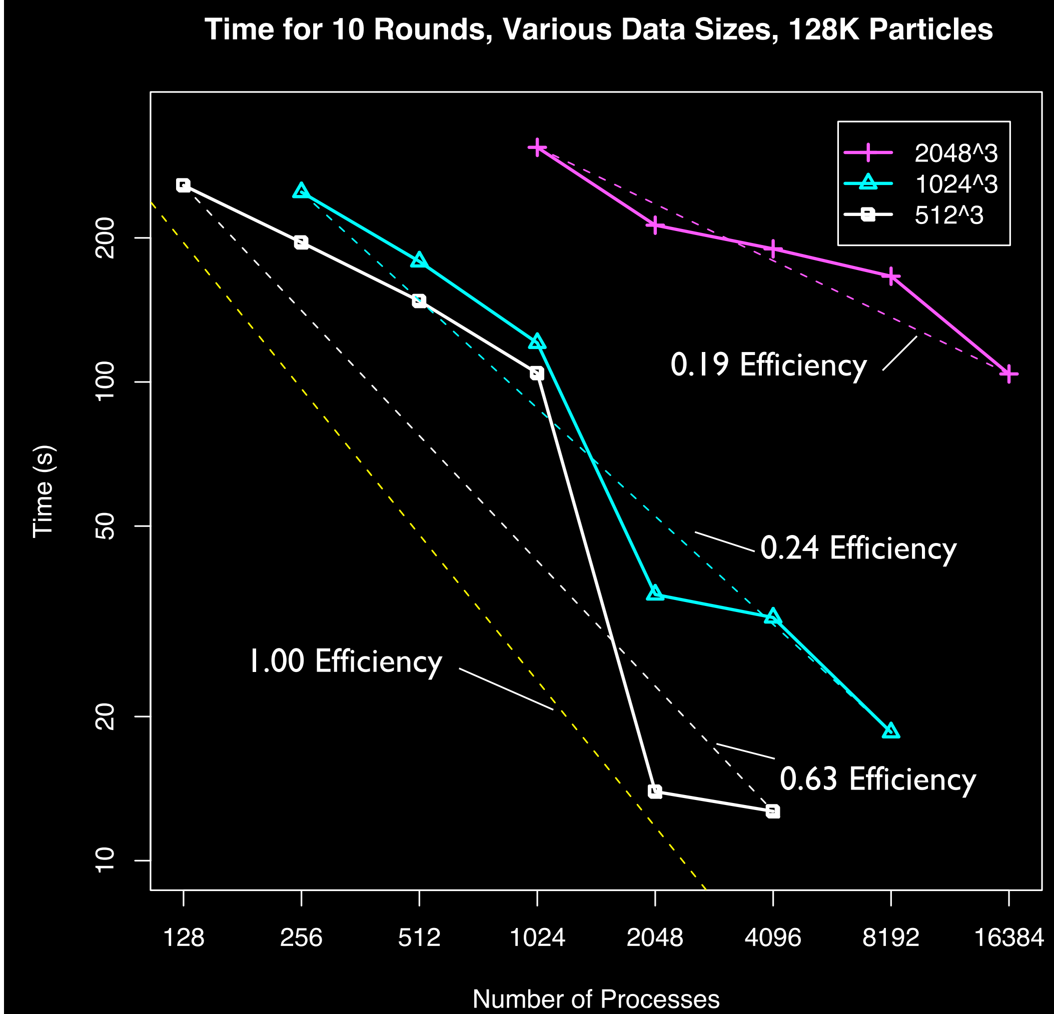
3. Advect particles along velocity vectors.

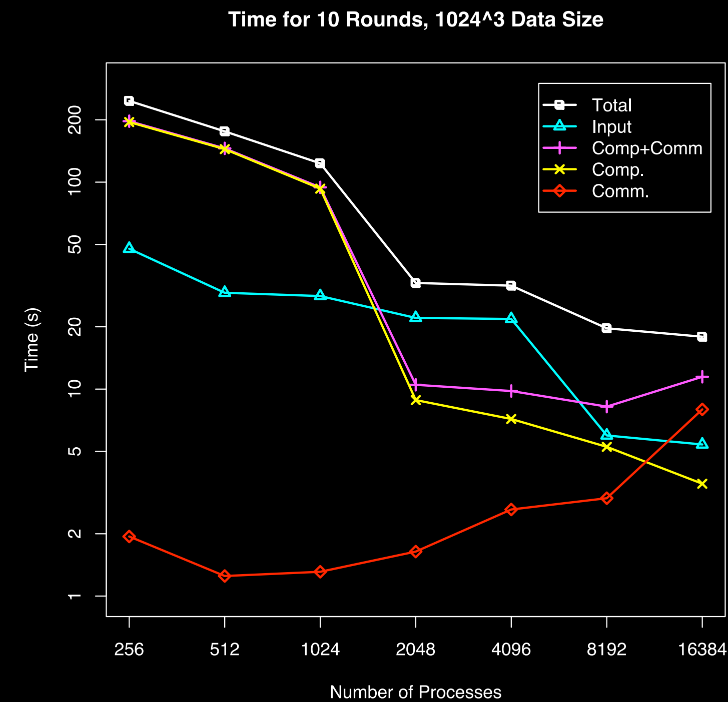


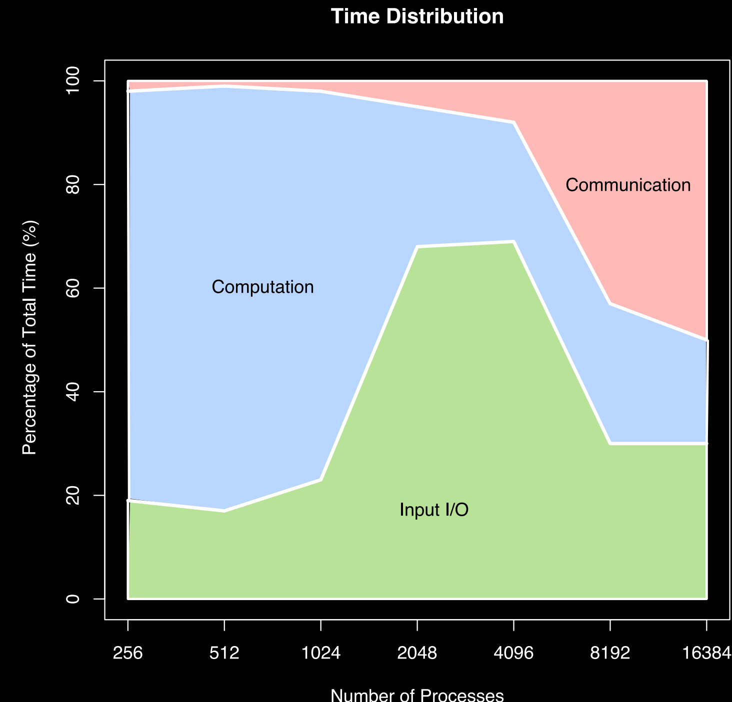
4. Exchange particles among processes when they reach the block boundary.

5. Repeat 3, 4

Strong Scaling Baseline Performance







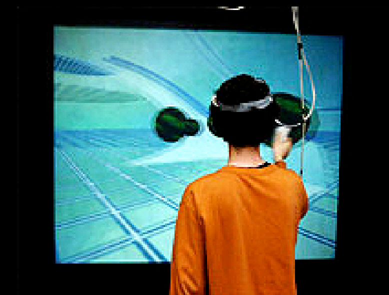
Thermal hydraulics flow. 134M cells, 8K particles.
1,2,4,8,16 round robin blocks per process.

Virtual Environments for Science: Be the Data

Stereo

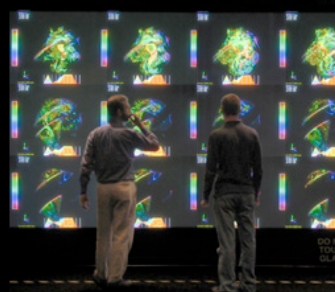
HMD

CAVE



GeoWall

Mono



Power Wall

Tiled Display

Autostereo

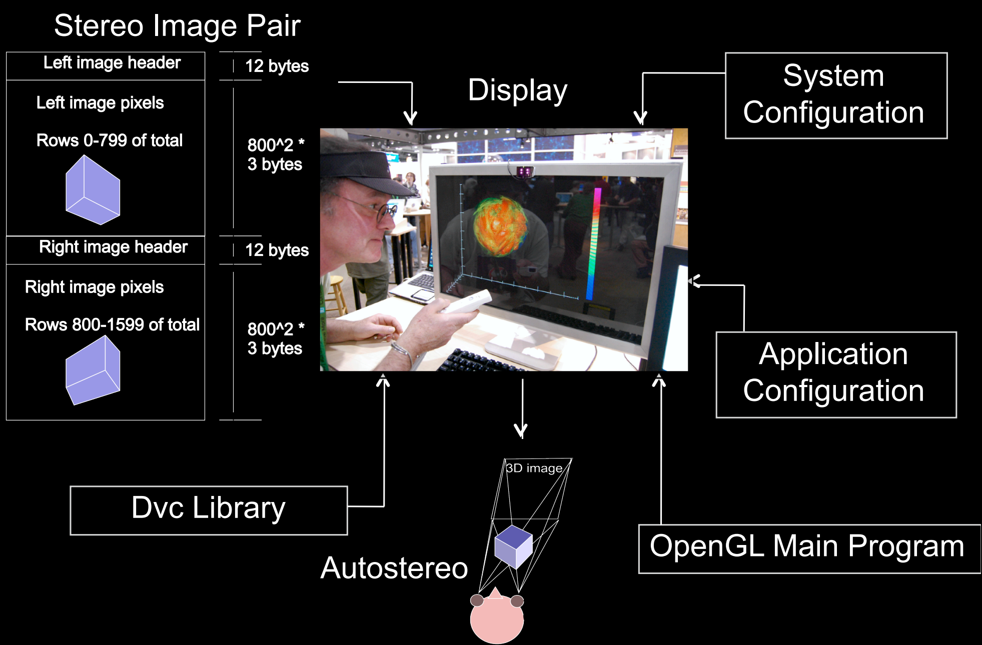
Varrier

Personal Varrier

Dynallax

Stereo Parallel Volume Rendering

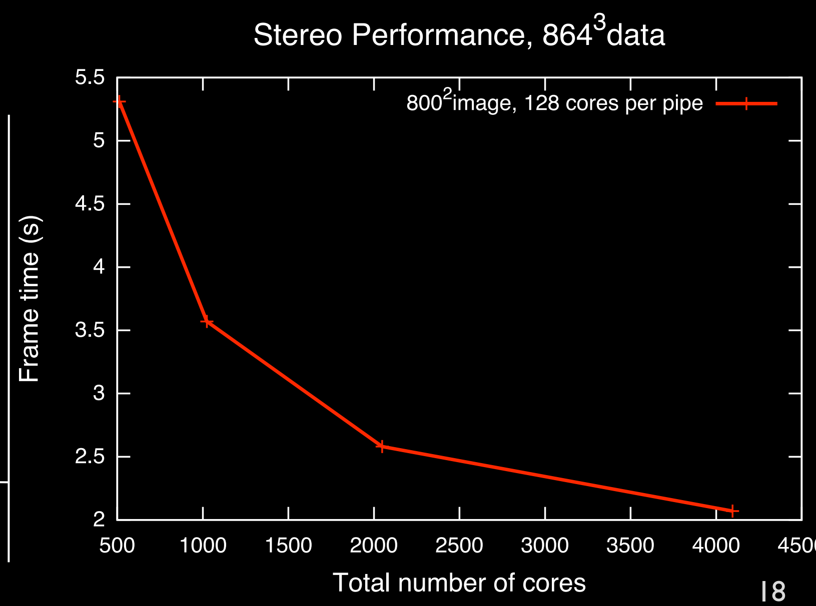
Display of Large-Scale Scientific Visualization. SPIE'09



Stereo parallel volume rendering: The server (BG/P) computes stereo pairs of volume-rendered images and streams them to the client, which runs the dvc library to display them remotely in autostereo.

End-to-end frame times of 2 s. per frame were achieved over a 3-hour demo from Argonne to Austin, TX.

Display devices and interaction techniques bring virtual environments to scientific visualization.



Rethinking Data-Intensive Analysis

Conclusions

-Exascale requires new thinking about analysis
-HPC resources can be harnessed for scalable analysis
-Scalable analysis is data-intensive: Moving data, transforming data, interacting with data
-Detailed study of data movement, both network and storage, is needed
-Virtual environments can help manage data complexity

Ongoing, Future

-Continue to collaborate with others in developing infrastructure for scalable analysis in other HPC subsystems

-Strengthen collaborations with scientists to integrate analysis with applications

-Continue to develop immersive interfaces and environments for science

“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

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