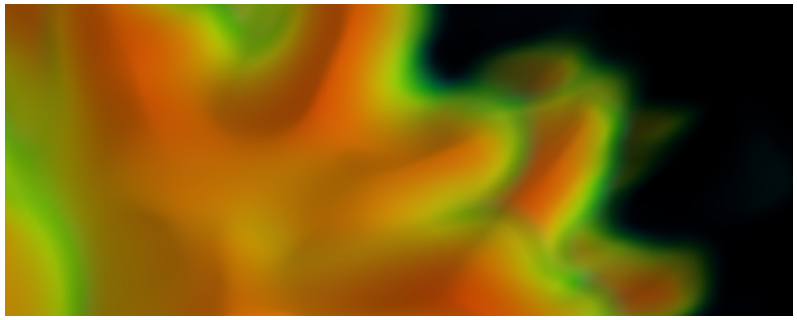


Visual Data Analysis Computational Requirements

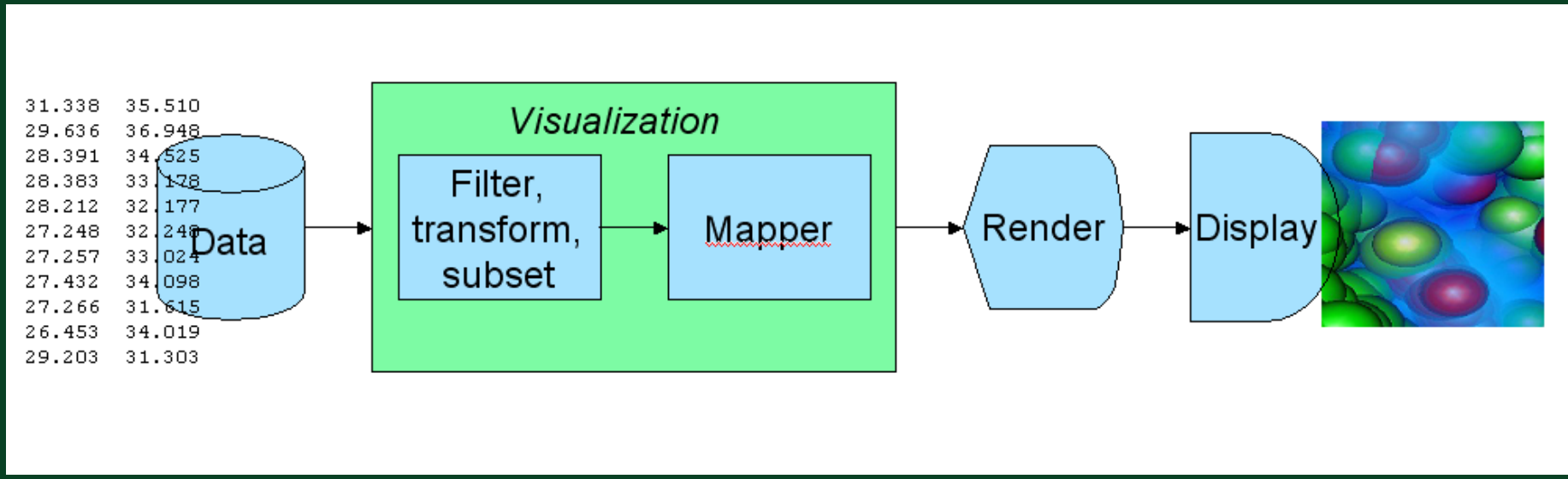
E. Wes Bethel

5 January 2011

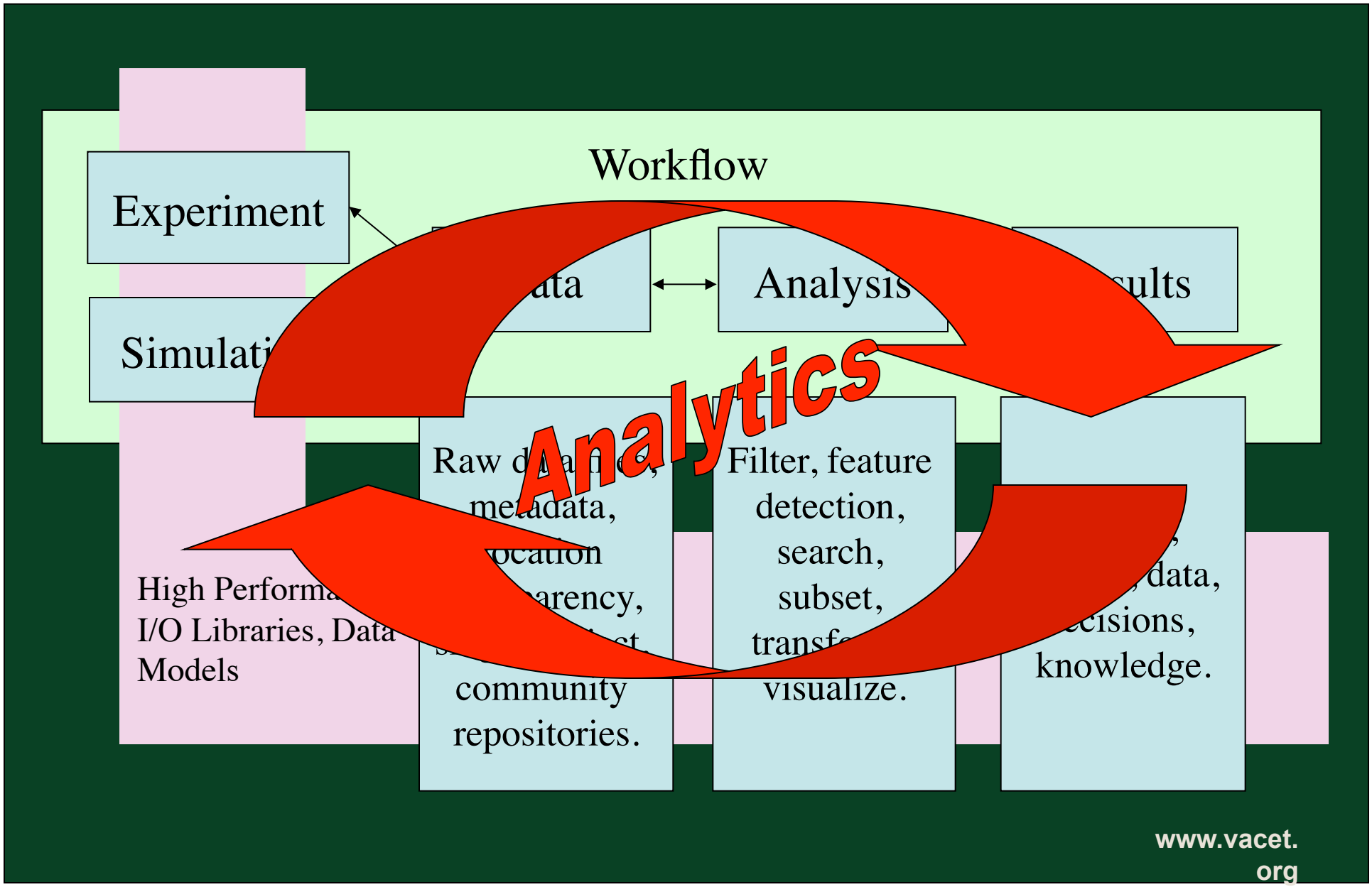
Oakland, CA

Visual Data Analysis?

- Visualization: transformation of data into images.
- Visual data analysis:
  - Reflects one of three different visualization use models:
  - Exploratory, analytical, presentation.

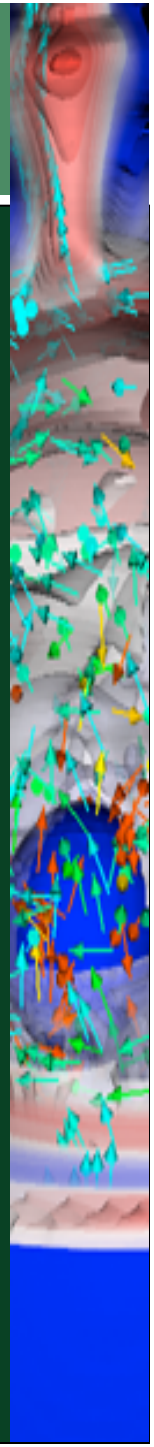


Map of the Problem Space



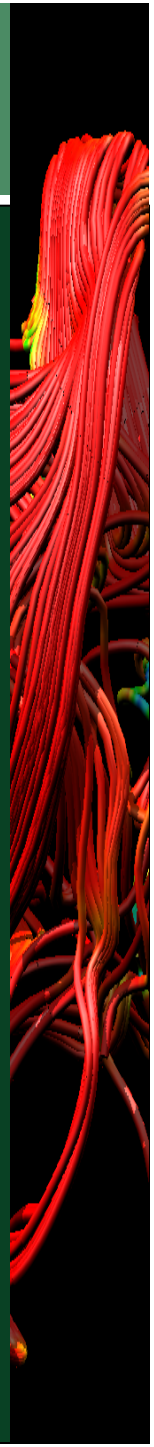
Berkeley Lab Visualization Group Mission

- Enable scientific knowledge discovery through the research, development, deployment, and application of visual data analysis technologies in the modern regime of HPC and data intensive science.
- We accomplish this mission by:
  - Focusing R, D, & D efforts at all stages of the visualization pipeline.
  - Close collaborations with science stakeholders to maximize likelihood of science impact.
  - Tightly integrated and well coordinated interaction between research, development, and production deployment activities.



Projects Represented Today

- (ASCR) SciDAC Visualization and Analytics Center for Enabling Technology (Research, development, deployment)
  - SciDAC-e: two additional projects – computational analysis infrastructure for carbon sequestration.
- (ASCR) LBNL Visualization Base Program (research)
- (ASCR) High performance parallel I/O (R&D)
- (BER) Visual data analysis of ultra-large climate data
- (EM) Advanced Simulation Capability for Environmental Management

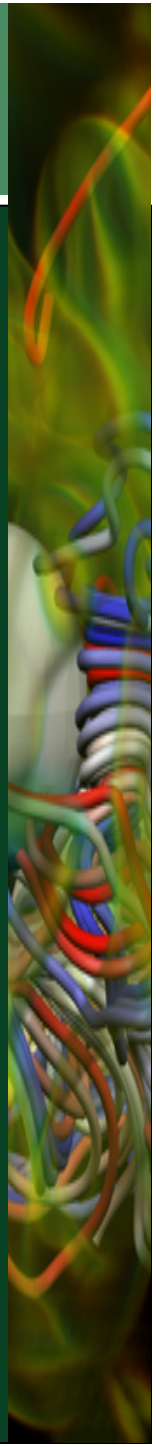


Visual Data Analysis Programs: What do we do?

- Basic research:
  - What are issues in doing visual analysis to the XX-scale?
  - How to solve impedance mismatch between Moore's Law growth in data size/complexity and (1) slowly growing I/O infrastructure and (2) "limited" human cognitive pathway?
- Applied research:
  - Enabling insight: finding needles in haystacks, new forms of analysis algorithms
- Development
  - Production-quality, petascale capable visual analysis software infrastructure.
- Deployment and application
  - Make s/w work reliably on large machines
  - Solve specific user problems in visual data analysis.

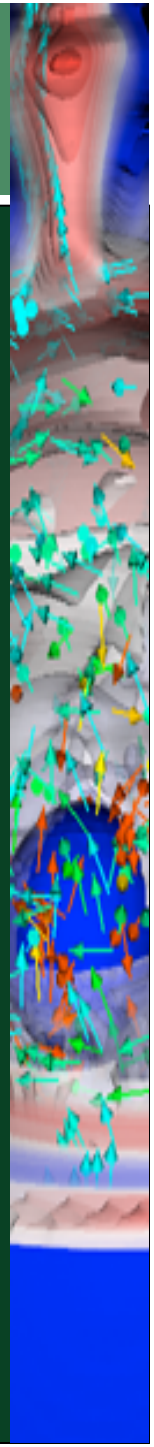
Role of SC Facilities for Visual Analysis Research

- Support basic and applied research
  - Provide access to emerging platforms for algorithmic R&D: big, parallel machines; distributed-memory GPU clusters
  - Testbeds and experimental facilities
- Provide infrastructure for conducting applied work
  - Apply software tools to specific user problems to produce scientific insight
- Vehicle for deploying research & development products to the scientific community.



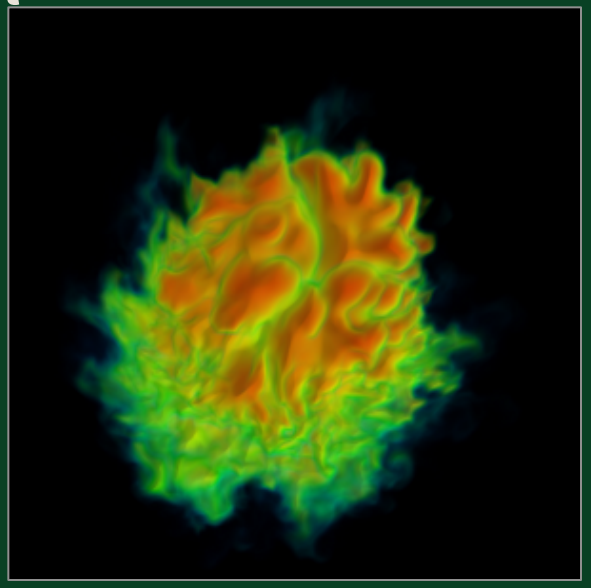
Two Case Studies

- Hybrid-parallelism, extreme concurrency visualization on large, distributed-memory systems.
  - Lots of work on computational/computer science studying hybrid parallelism, but mostly for solver-type code.
  - Our work: explore this space from a visual data exploration and analysis perspective.
- Bucket of other ideas
  - Diversity of projects begets diversity of requirements.



Hybrid-parallelism: proof at the petascale holds promise for the exascale.

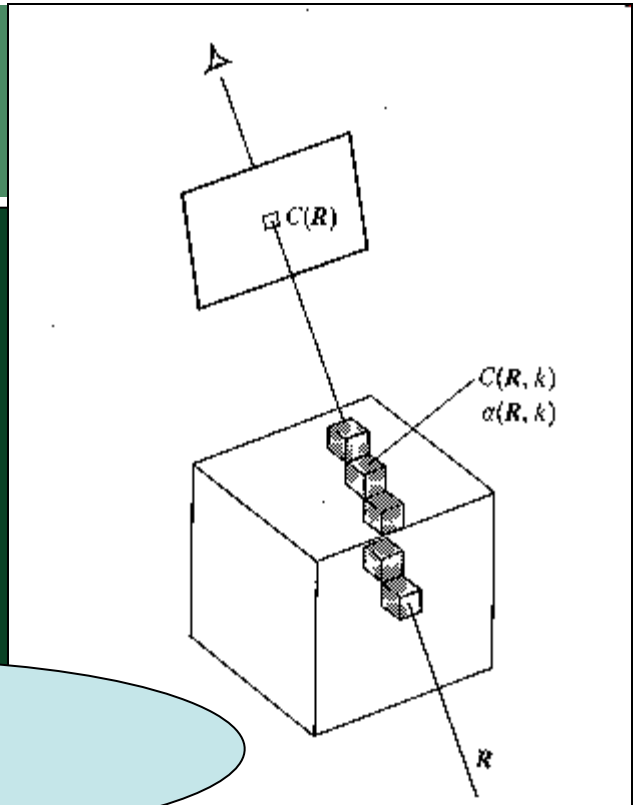
- Existing programming models may not work well at the exascale: multi- and many-core processors.
- Early studies show promise: hybrid-parallel approach outperforms MPI-based approaches on largest-ever visualization runs on DOE supercomputers.
- These results suggest hybrid-parallelism likely a good approach for exascale class machines.



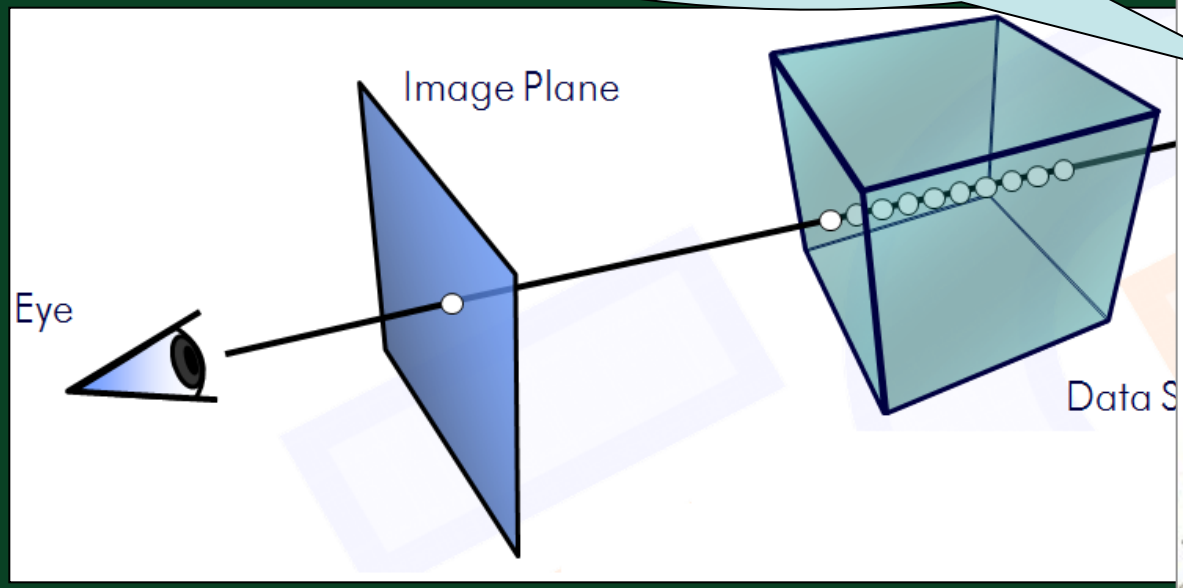
Hybrid-parallel volume rendering of 64-billion zones from combustion simulation on 216,000 cores of JaguarPF at ORNL.

Volume Rendering (serial)

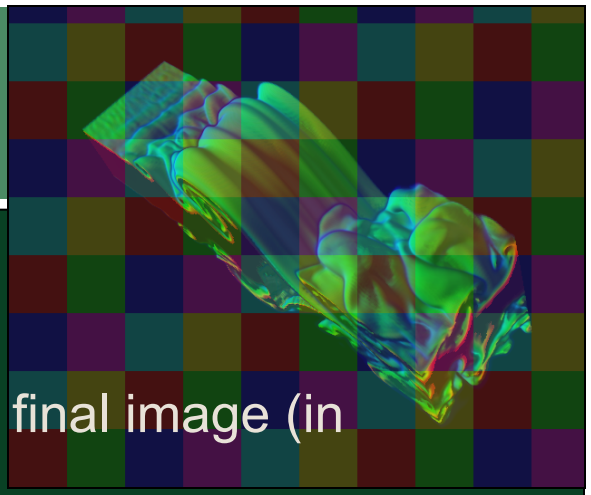
- Overview of Levoy's method
  - For each pixel in image plane:
    - Find intersection of ray and volume
    - Sample data (RGBa) along ray, integrate samples to compute final image pixel color



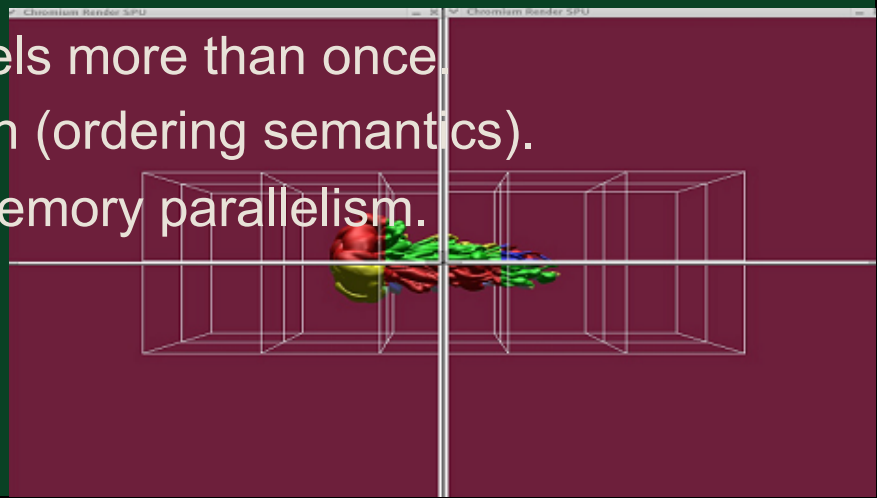
Unstructured!



Parallelizing Volume Rendering

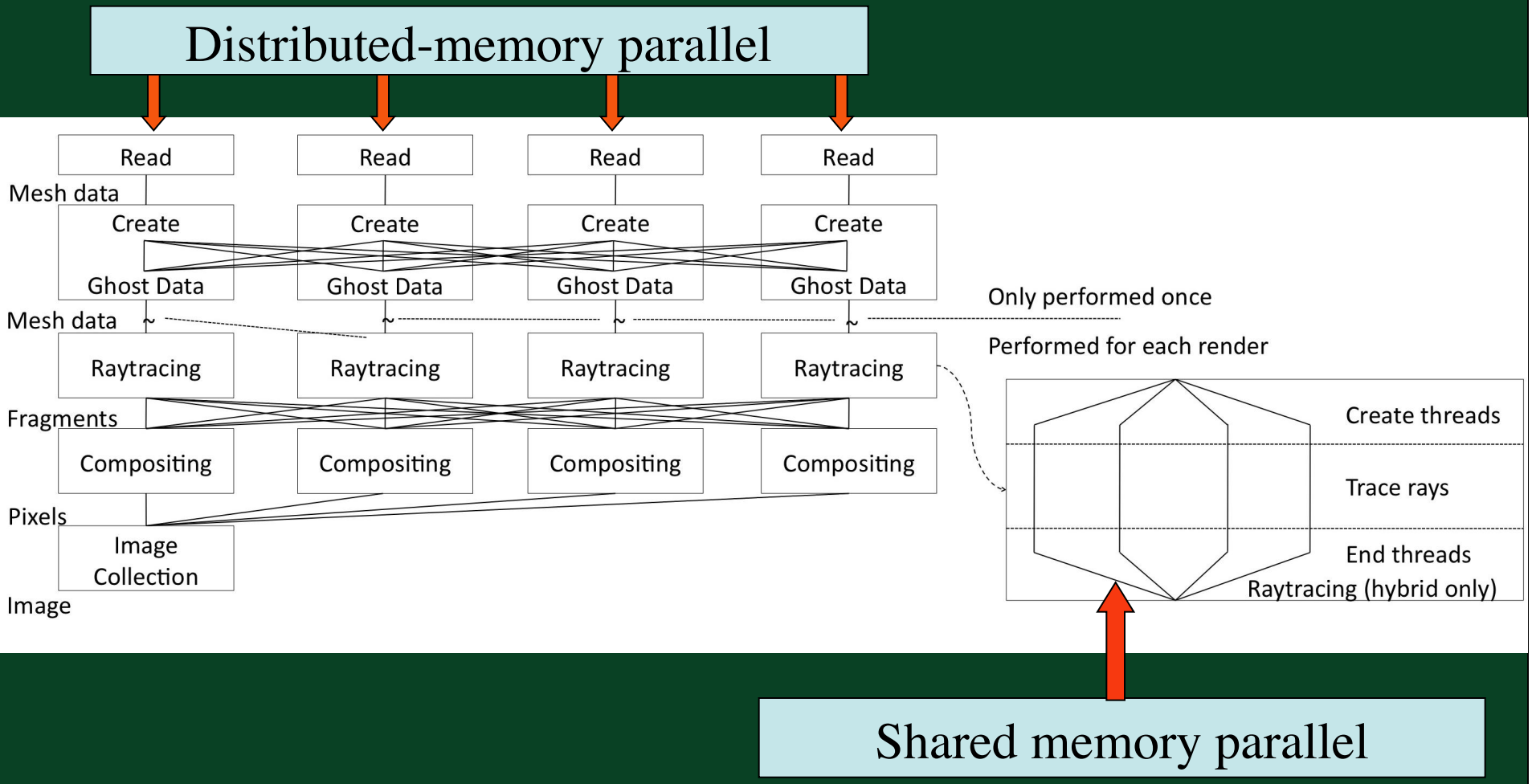


- Image-space decomposition.
  - Each process works on a disjoint subset of the final image (in parallel)
  - Processes may access source voxels more than once, will access a given output pixel only once.
  - Great for shared memory parallelism.
- Object-space decomposition.
  - Each process works on a disjoint subset of the input data (in parallel).
  - Processes may access output pixels more than once.
  - Output requires image composition (ordering semantics).
  - Typical approach for distributed memory parallelism.



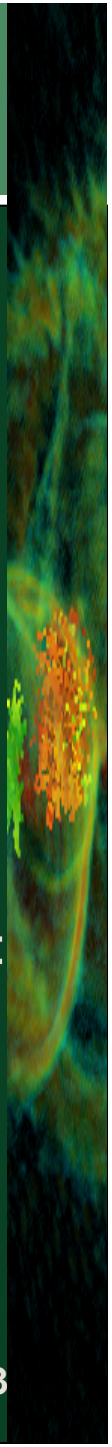
Hybrid Parallel Volume Rendering

- Our hybrid-parallel architecture:



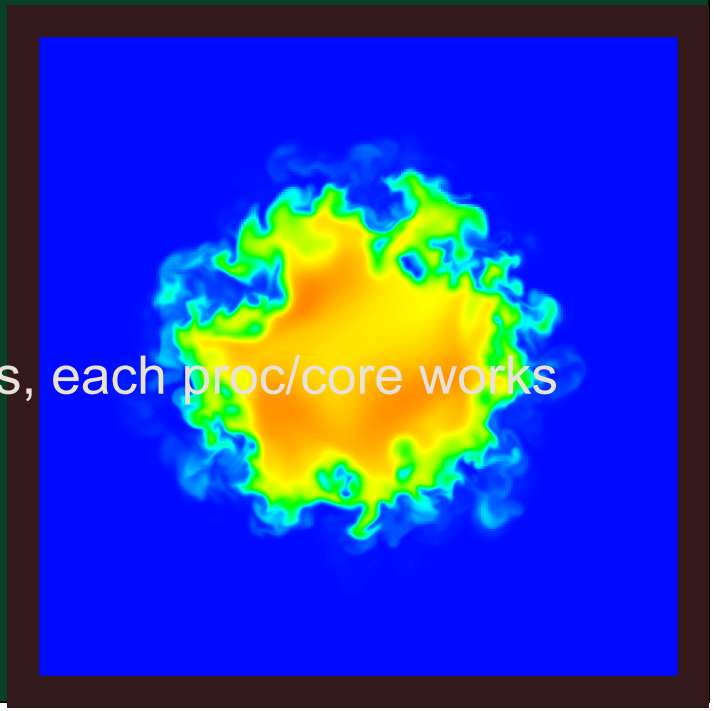
Our Experiment

- Thesis: hybrid-parallel will exhibit favorable performance, resource utilization characteristics compared to traditional approach.
- How/what to measure?
  - Memory footprint, communication traffic load, scalability characteristics, absolute runtime.
  - Across a wide range of concurrencies.
    - Remember: we're concerned about what happens at extreme concurrency.
  - Algorithm performance somewhat dependent upon viewpoint, data:
    - Vary viewpoints over a set that cut through data in different directions: will induce different memory access patterns.
- Strong scaling study: hold problem size constant, vary amount of resources.
- Weak scaling study: increasing problem size with increasing concurrency.

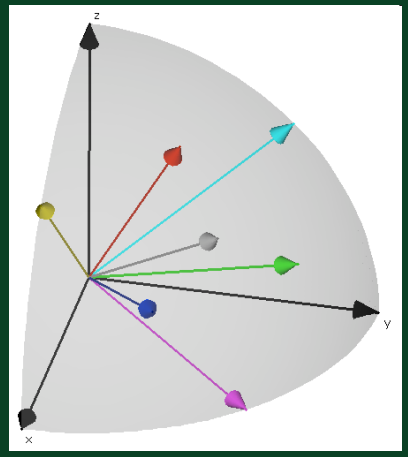


Experiment: Platform and Source Data

- Platform: JaguarPF, a Cray XT5 system at ORNL
  - 18,688 nodes, dual-socket, six-core AMD Opteron (224K cores)
- Source data:
  - Combustion simulation results, hydrogen flame (data courtesy J. Bell, CCSE, LBNL)
  - Effective AMR resolution: 1024^3, flattened to 512^3, runtime upscaled to 4608^3 (to avoid I/O costs).
  - 91B cells, ~3TB total memory footprint.
- Target image size: 4608^2 image.
  - Want approx 1:1 voxels to pixels.
- Strong scaling study:
  - As we increase the number of procs/cores, each proc/core works on a smaller-sized problem.
  - Time-to-solution should drop.



Experiment: The Unit Test



- Raycasting time: view/data dependent
  - Execute from 10 different prescribed views: forces with- and cross-grained memory access patterns.
  - Execute 10 times, result is average of all.
- Compositing
  - Five different ratios of compositing PEs to rendering PEs.
- Measure:
  - Memory footprint right after initialization.
  - Memory footprint for data blocks and halo exchange.
  - Absolute runtime and scalability of raycasting and compositing.
  - Communication load between RC and compositing.

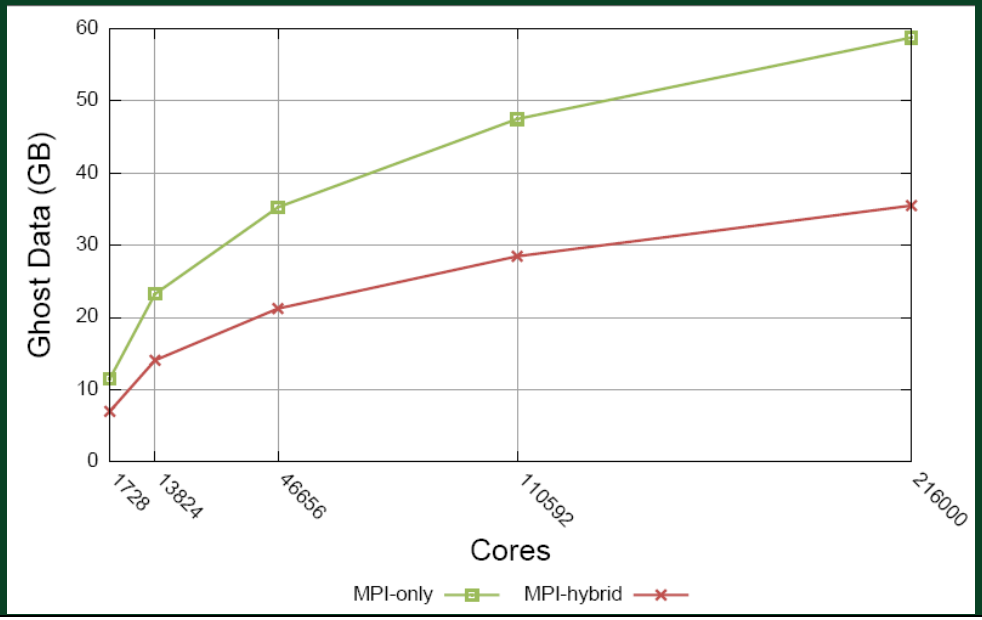
Memory Use – MPI_Init()

- Per PE memory:
  - About the same at 1728, over 2x at 216000.
- Aggregate memory use:
  - About 6x at 1728, about 12x at 216000.
  - At 216000, -only requires 2GB of memory for initialization per node!!!

Cores	Mode	MPI PEs	MPI Runtime Memory Usage		
			Per PE (MB)	Per Node (MB)	Aggregate (GB)
1728	MPI-hybrid	288	67	133	19
1728	MPI-only	1728	67	807	113
13824	MPI-hybrid	2304	67	134	151
13824	MPI-only	13824	71	857	965
46656	MPI-hybrid	7776	68	136	518
46656	MPI-only	46656	88	1055	4007
110592	MPI-hybrid	18432	73	146	1318
110592	MPI-only	110592	121	1453	13078
216000	MPI-hybrid	36000	82	165	2892
216000	MPI-only	216000	176	2106	37023

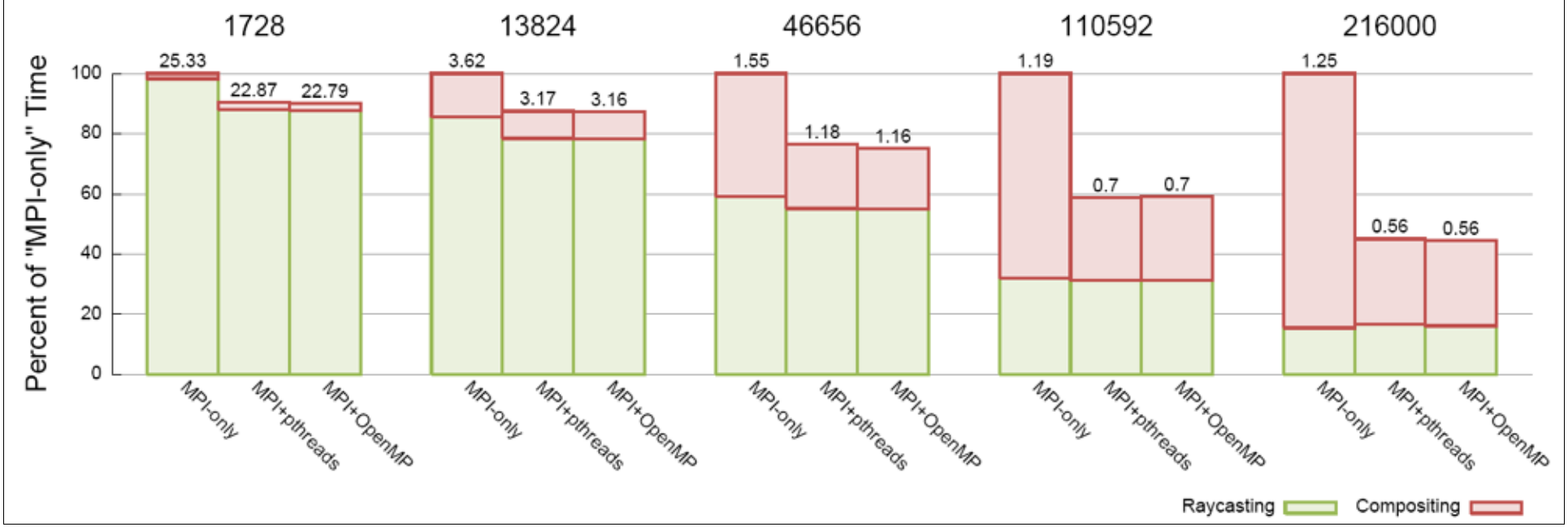
Memory Use – Ghost Zones

- Two layers of ghost cells required for this problem:
  - One for trilinear interpolation during ray integration loop.
  - Another for computing a gradient field (central differences) for shading.
- Hybrid approach uses fewer, but larger data blocks.
  - ~40% less memory required for ghost zones (smaller surface area)
  - Reduced communication costs



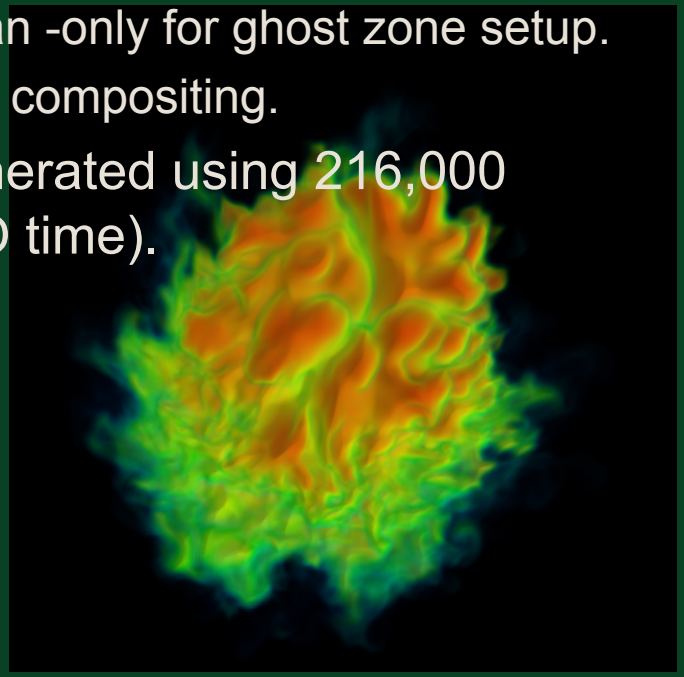
Absolute Runtime

- -hybrid outperforms –only at every concurrency level.
  - At 216K-way parallel, -hybrid is more than twice as fast as –only.
  - Compositing times begin to dominate: communication costs.



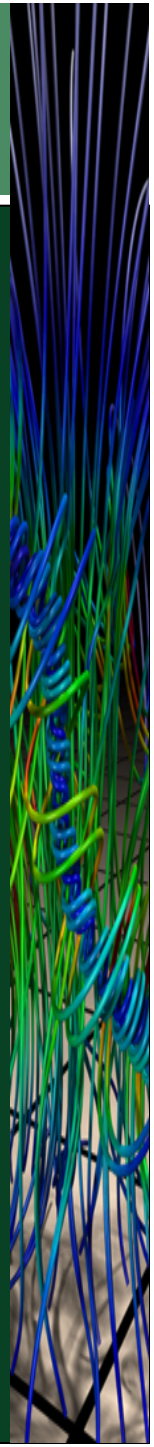
Summary of Results

- Absolute runtime: -hybrid twice as fast as –only at 216K-way parallel.
- Memory footprint: -only requires 12x more memory for MPI initialization then –hybrid
  - Factor of 6x due to 6x more MPI PEs.
  - Additional factor of 2x at high concurrency, likely a vendor MPI implementation (an N^2 effect).
- Communication traffic:
  - -hybrid performs 40% less communication than -only for ghost zone setup.
  - -only requires 6x the number of messages for compositing.
- Image: 4608^2 image of a ~4500^3 dataset generated using 216,000 cores on JaguarPF in ~0.5s (not counting I/O time).



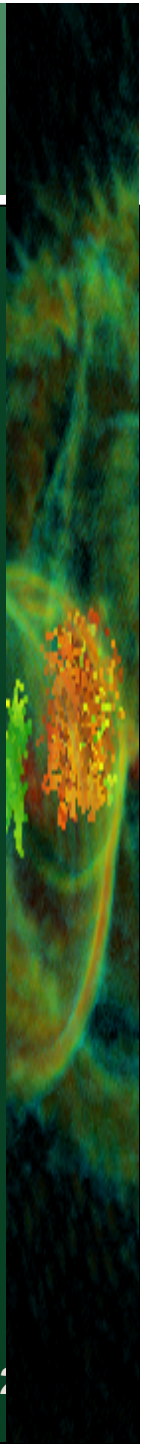
More recent results

- Weak scaling study
  - Up to $23{,}000^3$ grids, $23{,}000^2$ image size.
  - ~300TB memory footprint
- Include many-core platform: GPU
  - CUDA implementation of “kernel”
  - 256 GPU system ~= 40K cores Cray XT5
  - Small memory footprint
  - Hardware performance counters?
- Results:
  - Similar to strong scaling study results

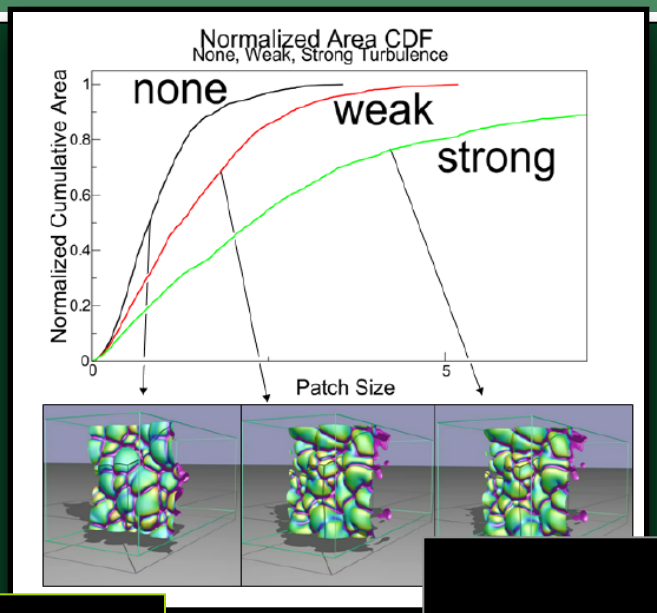


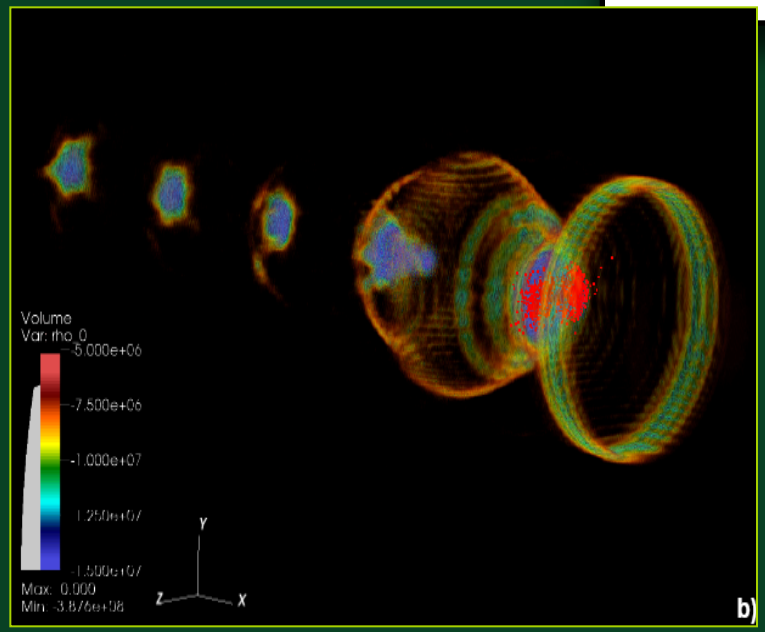
Resource requirements for this work

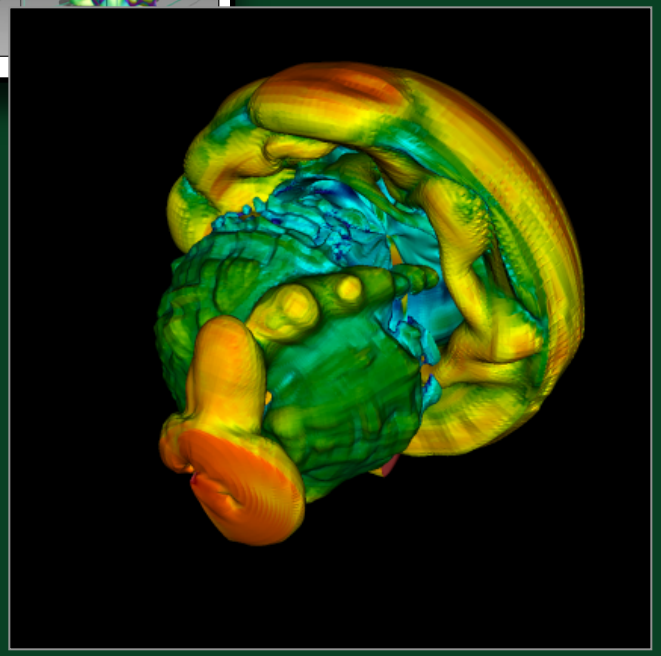
- NERSC
  - Multiple 32K-way runs on franklin (~500K hours)
  - Consulting help with making 32K-way parallel MPI jobs work, understanding behavior characteristics of interconnect fabric.
  - Compilers, MPI, libraries (pthreads, OpenMP)
  - Hardware performance counters.
- OLCF
  - Multiple 216K-way parallel runs (~7M hours)
  - Consulting help with high-concurrency jobs
  - Notes:
    - 300TB memory footprint, avoided doing 300TB of I/O per run by using upsampling.
    - For “real use” (in postprocessing mode), a BIG, unavoidable I/O cost is coming.



Case Study #2 – Collection of Projects

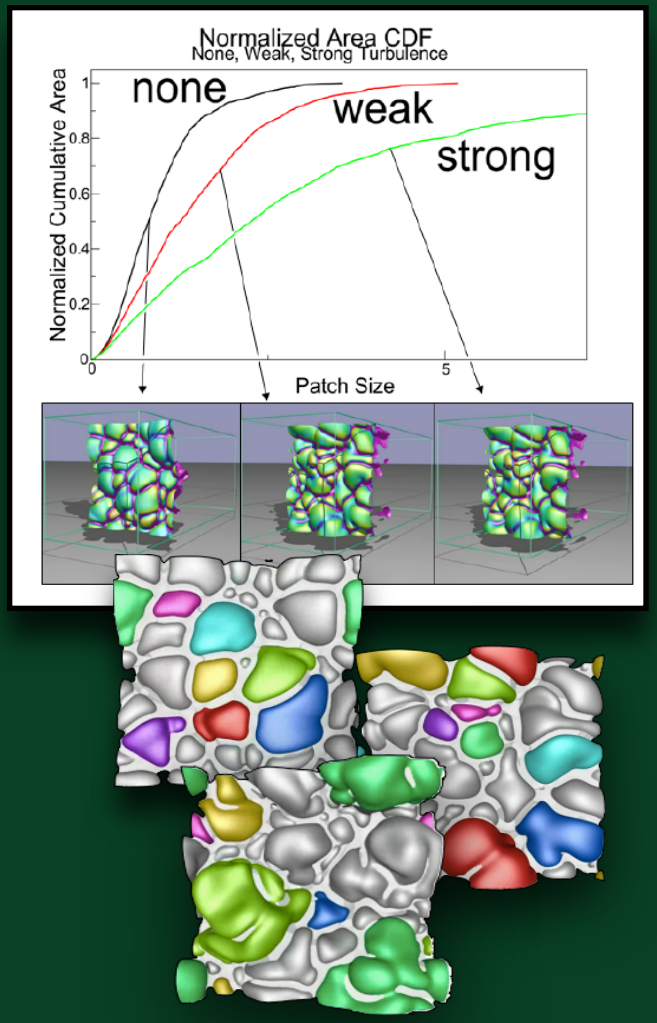






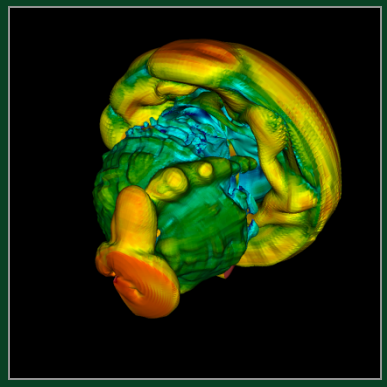
Analysis of Combustion Simulation Data

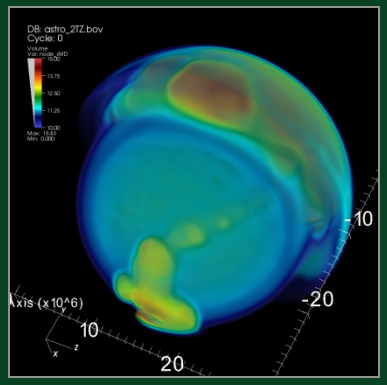
- **Problem:** Data of increasing size and complexity increasingly difficult to analyze.
- Accomplishments:
  - New approaches based upon topological methods offer the means to discover relationships, features, and characteristics in today's largest datasets.
- Science Impact:
  - First-ever quantitative analysis of large, time-varying combustion simulation data to study influence of turbulence on size/shape of combustion regions in lean, premixed hydrogen flames.
- PI: John Bell (LBNL), SciDAC Community Astrophysics Consortium Partnership, Incite Awardee.



Production Visualization at the Petascale

- Petascale machines are unique, need visual data analysis tools capable of leveraging the entire resource to ingest and process today's largest scientific datasets.
- SciDAC Visualization and Analytics Center for Enabling Technologies produces such software, proves its effectiveness on all major DOE computational platforms, and distributes it at no charge to the science community.
- Investments in software infrastructure pay off by producing visualization software that can effectively harness the power of today's largest supercomputers for scientific data analysis.

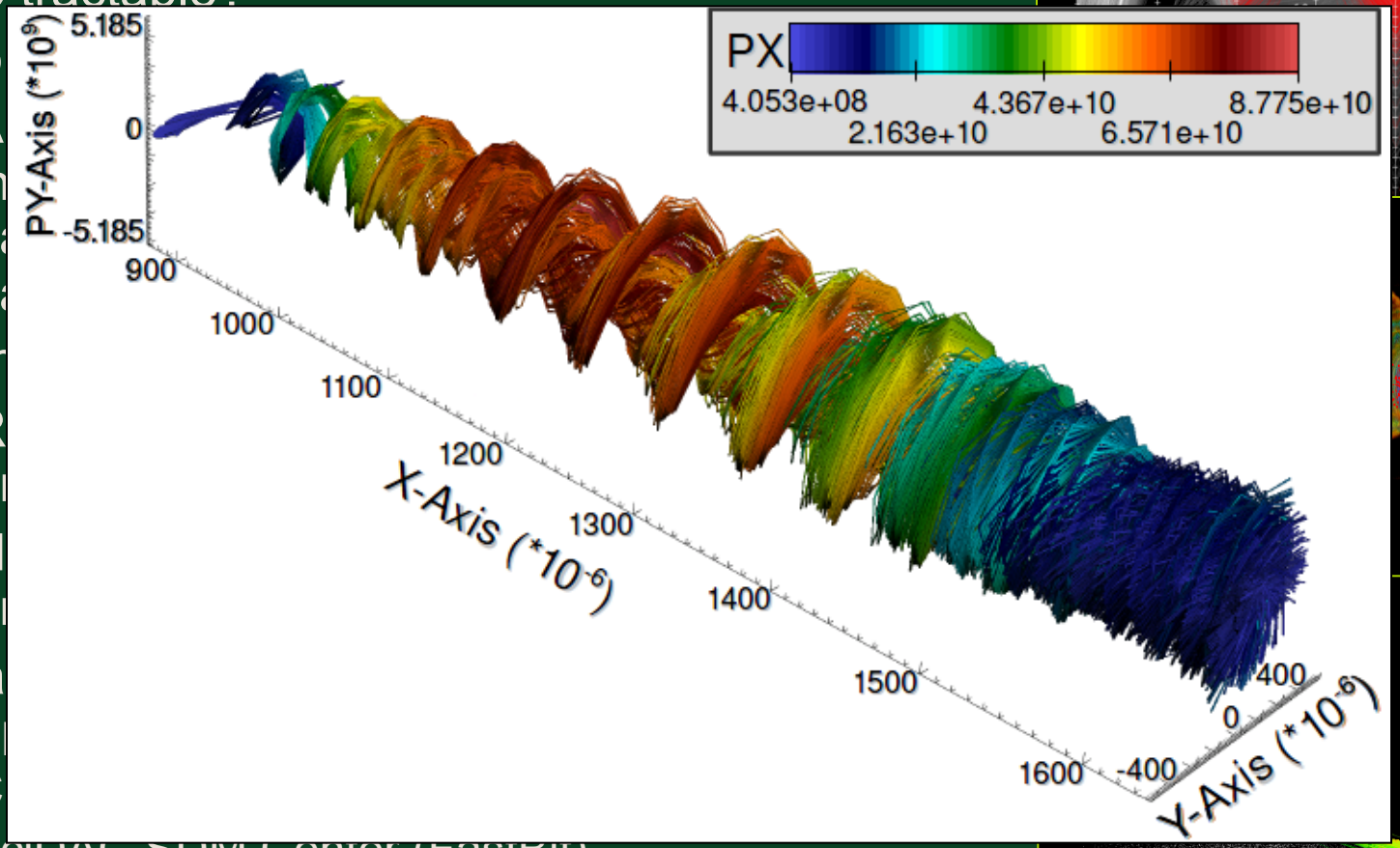


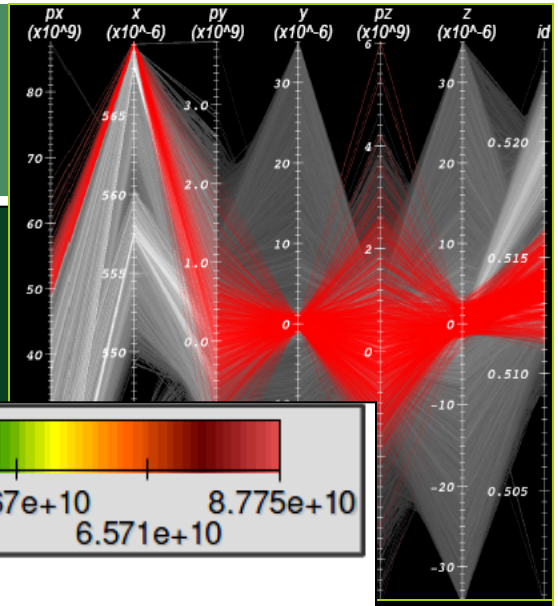


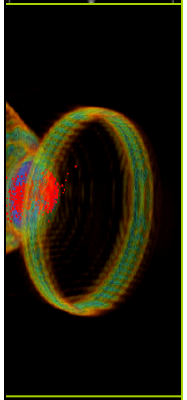
Visualization of supernova simulation results, conducted at 32,000-way parallel on JaguarPF (ORNL) and Franklin (NERSC).

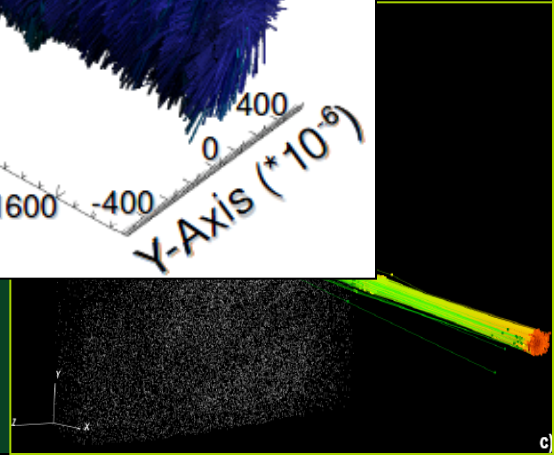
QDV and Accelerator Modeling

- Problem: sheer size and complexity of data is a barrier to analysis. How to make the problem more tractable?
- Acco
  - A
    in
    da
    pa
- Scier
  - R
    or
  - N
    ar
- Colla
  - Pl
    C
  - SciDAC SDM Center (FastBit)
  - Tech-X (Accelerator scientists)







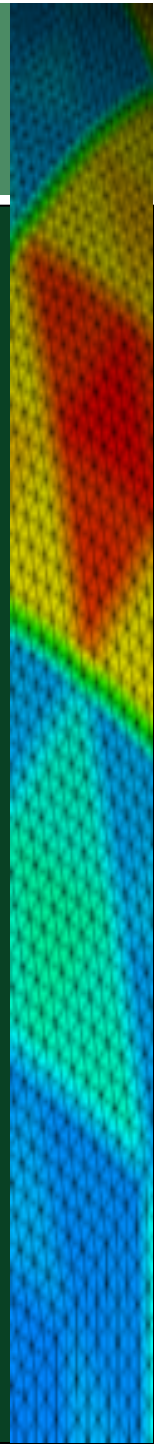


Glimpse of Current Work (partial)

- Climate data analysis
  - Increasingly refined simulations produce data too large for legacy visual data analysis and exploration software.
- High Performance I/O
  - Optimizing production-quality I/O for use on SCs, optimizing infrastructure for analysis-friendly metadata ops.
- Topological Data Analysis
  - New analysis methodology applied to multiple science domains.
- Carbon Sequestration
  - Machine learning, computer vision, multivariate analysis, geometric analysis, and visualization help provide traction on understanding how CO2 interacts with porous storage media.

Comments on Resources for the Future

- The “Labrador” effect
- Little Big Iron and the Tale of the Three Skinny Guys
- The Value of Services
- I/O, I/O, I/O
- Unusual requirements

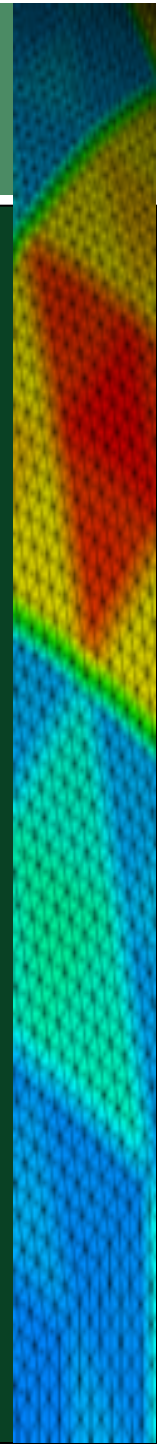


The “Labrador” Effect

- Visual data analysis research, development, deployment will “eat everything in sight.”
  - Different from “the telescope lemma”
  - Related to “chicken-egg” problem.
- Postprocessing use model:
  - Machine used for analysis should be commensurate in memory, I/O to the machine used to create the data.
- Concurrent, or in-situ use model:
  - Do visual data analysis concurrent with the simulation.
  - Good for some use models (single-timestep analysis), bad for others (temporal analysis, exploratory vis).

Visual Data Analysis Hardware Infrastructure

- Little Big Iron and the Tale of the Three Skinny Guys
  - “Little Iron” == platform for doing visual data analysis
- How big? What architecture?
  - How big? Memory and I/O capacities commensurate with machine used to create data.
  - What architecture?
    - Same as SC: similar obsolescence characteristics, not a parasitic expense, access to best interconnect fabric, reduced redundancy of costs for I/O hardware.
    - Different from SC: may have better commercial s/w support, may provide better support for specialized apps/processing modes (e.g., large memory serial jobs).
  - Testbeds: GPUs, FPGAs, others that are not available “on the desktop”
  - Needs support for our diverse program requirements.

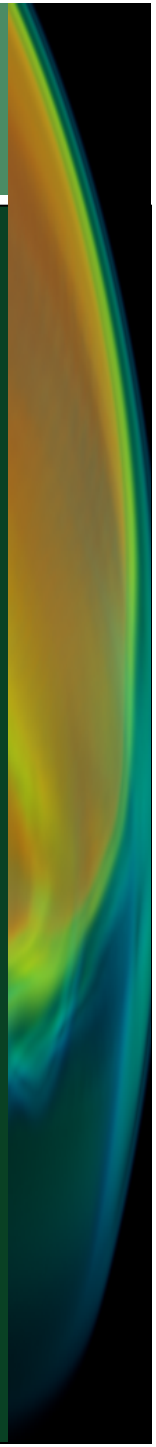


Other Services that Would be Helpful

- Web hosting for project websites (e.g., www.vacet.org)
- User/PI-administrable
  - Project email lists
  - CVS/SVN revision control/repositories
  - Project wiki's

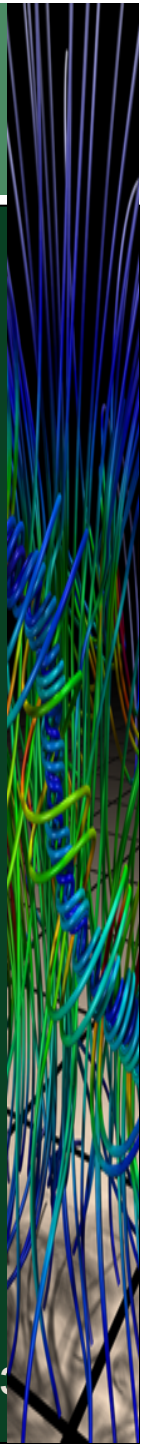
I/O, I/O, I/O

- I/O is the most costly part of analysis.
  - Lots of research focusing on concurrent compute +analysis.
  - Good for, say, 50% of use cases.
- I/O capacity needs to grow proportionally with flop rate
- Instrumentation, performance analysis infrastructure



Unusual Requirements: Deployment

- (Workflow proxy)
- Climate data analysis (10-05 project)
  - Problem: unusual requirements for software deployment, execution
  - Front-end:
    - UV-CDAT is visual interface to data management, analysis software.
  - Middle-tier:
    - ESG nodes stage data, prepare it for processing
  - Back-end:
    - Heavy-lifting analysis runs on SCs
- Need ongoing, close help of SC staff to make this stuff work for science stakeholders!



The End