Analysis and Visualization of HPC datasets

Sean Ahern

Oak Ridge National Laboratory

13th Workshop on Distributed Supercomputing

End-to-end schematic

HPC System

End-to-end schematic

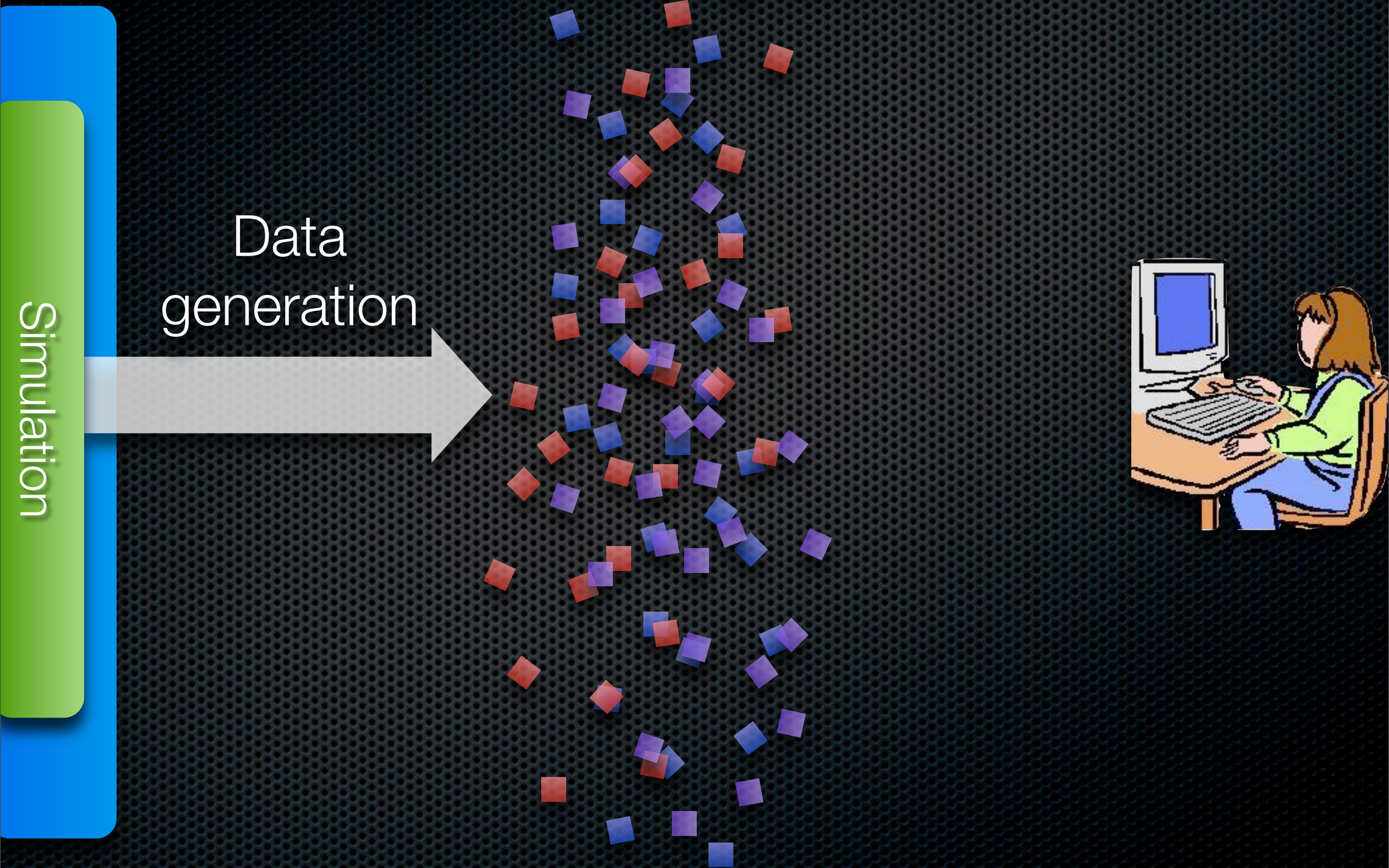
HPC System

Input

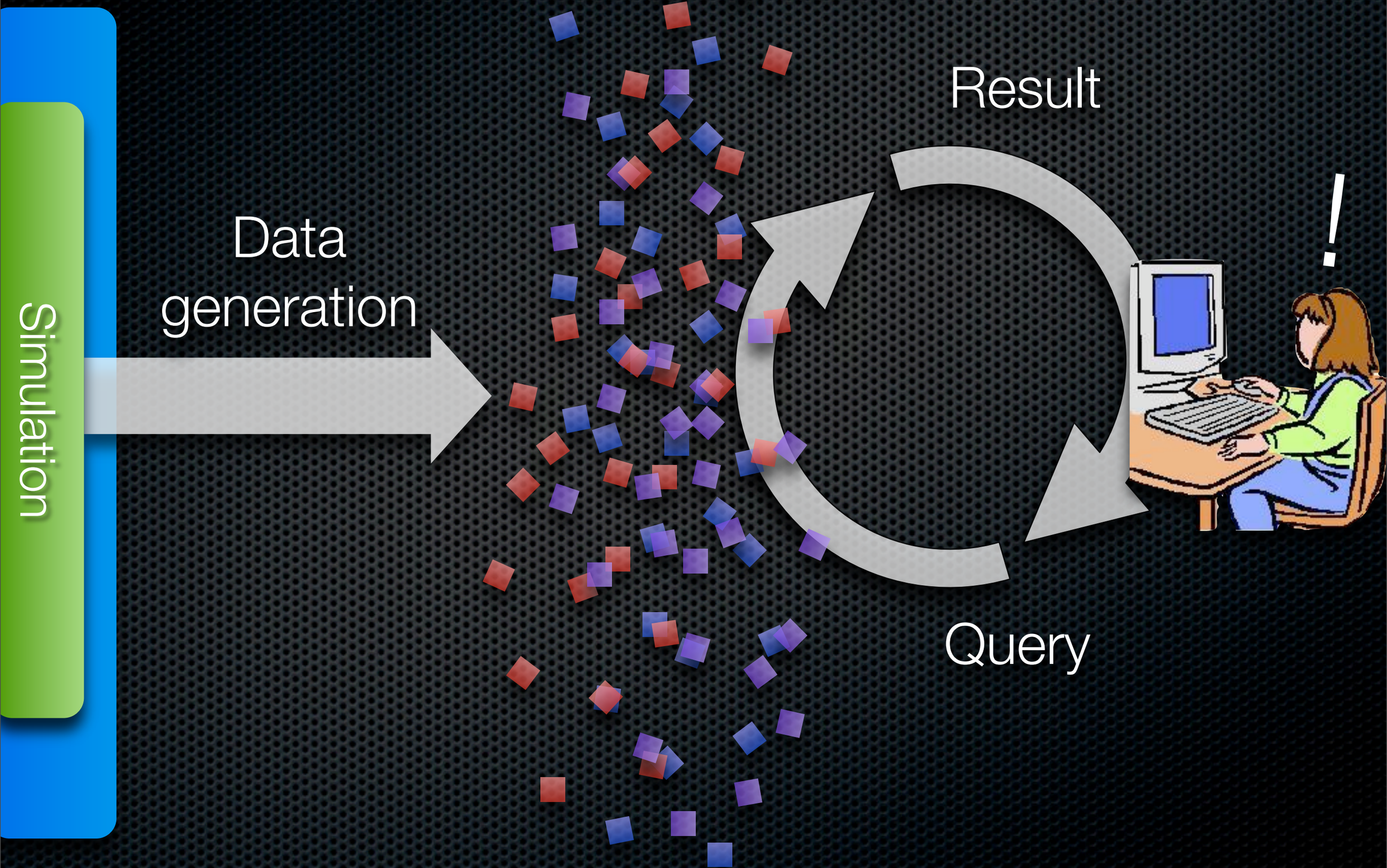
End-to-end schematic

Simulation

End-to-end schematic

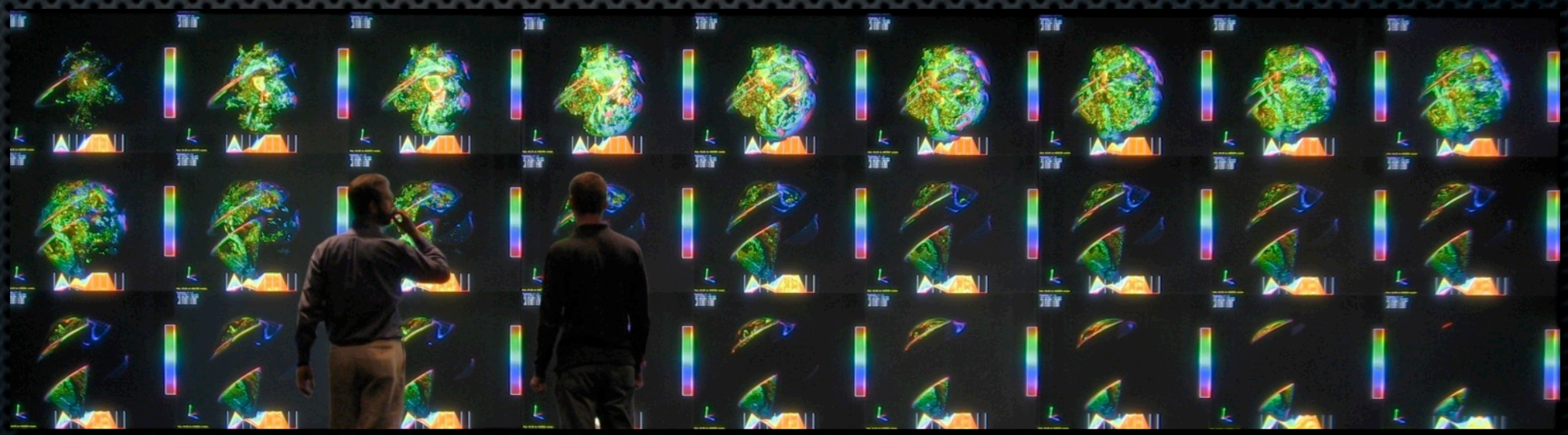


End-to-end schematic



How is “vis” different from a “regular” app

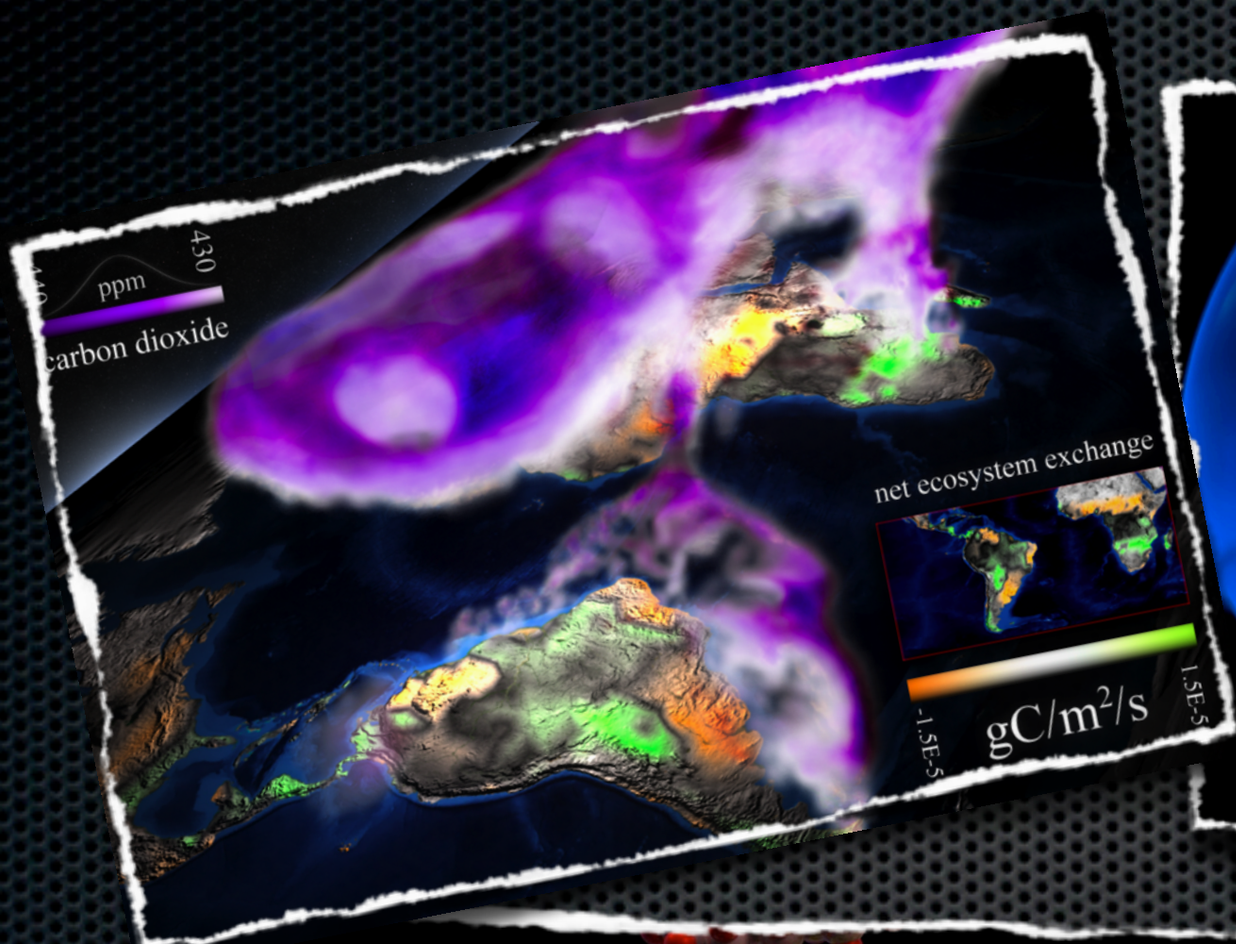
- I/O is different
  - Analysis/vis is a consumer of data vs. a producer
- Interactivity – broadly defined
- Connection to users’ eyeballs

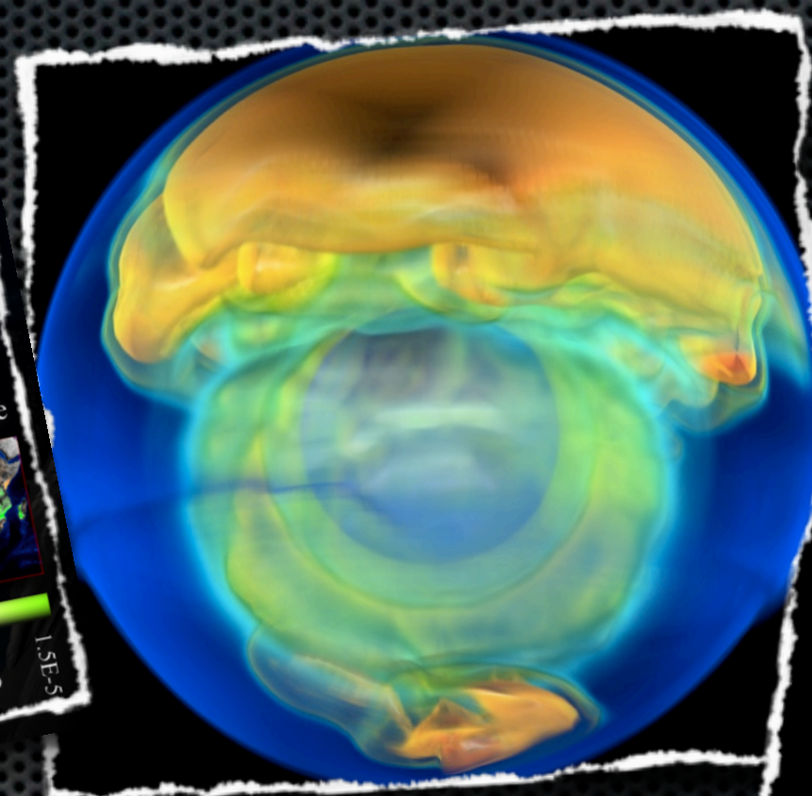


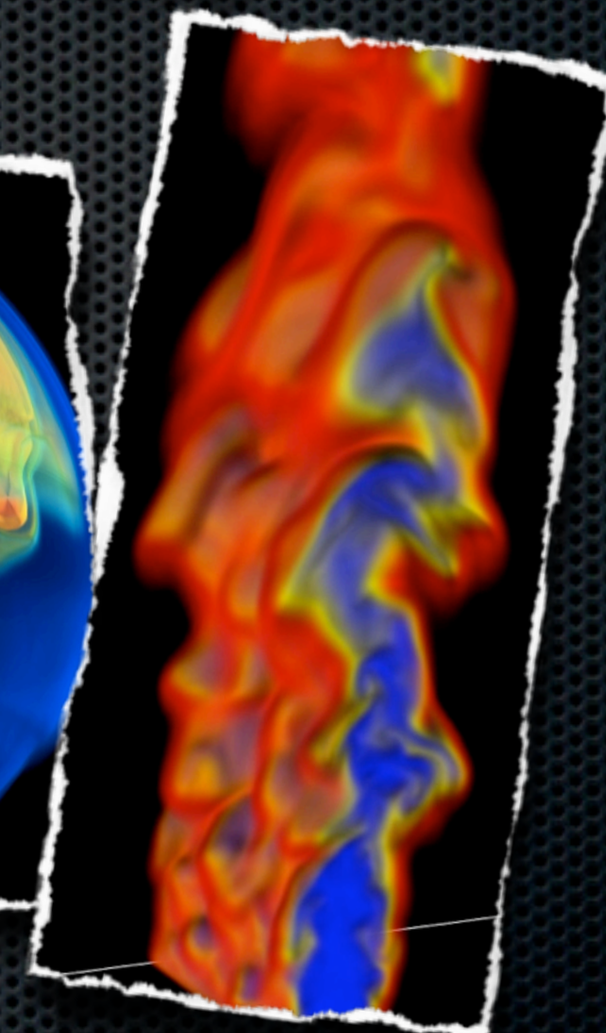
Session “take-home” messages

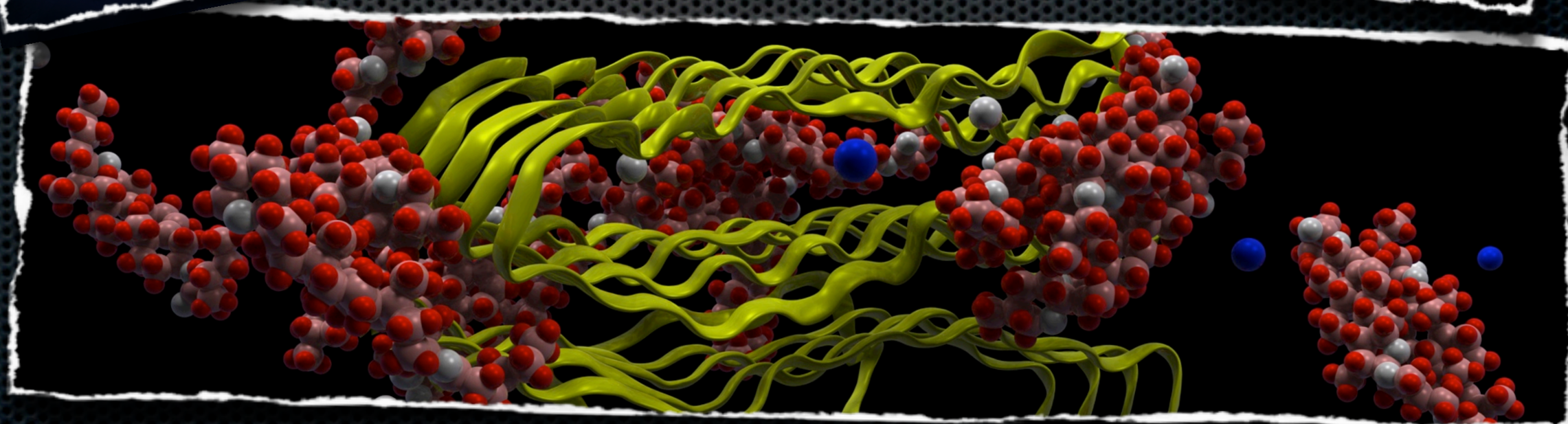
- I/O is our primary gating factor.
- “Vis clusters” are not sufficient for the petascale.
- It’s much more than just rendering.
- Analysis is very domain-specific.
- Pushing analysis/vis to the HPC system requires policy changes.

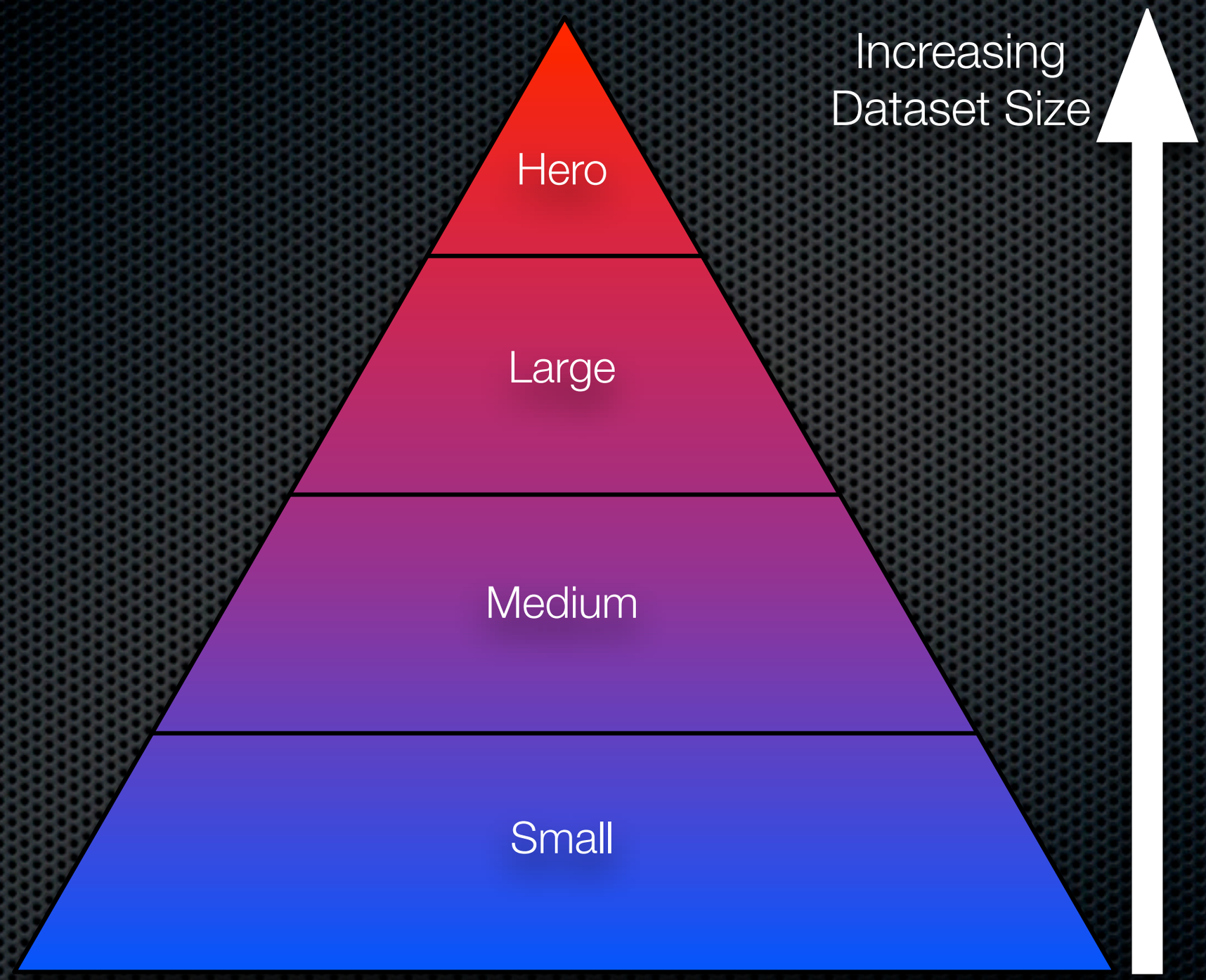
Success with “traditional” vis







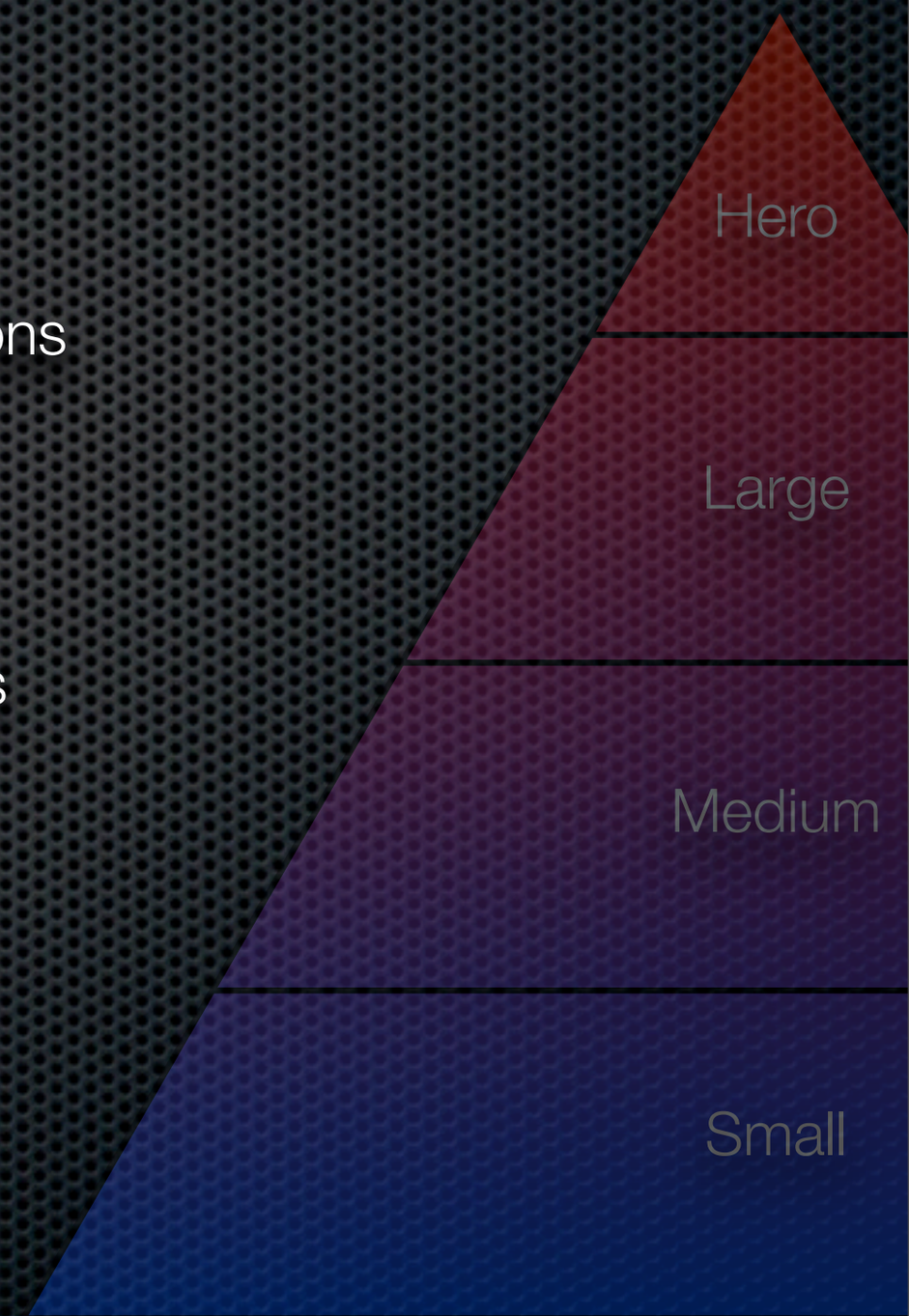




Dataset size growth

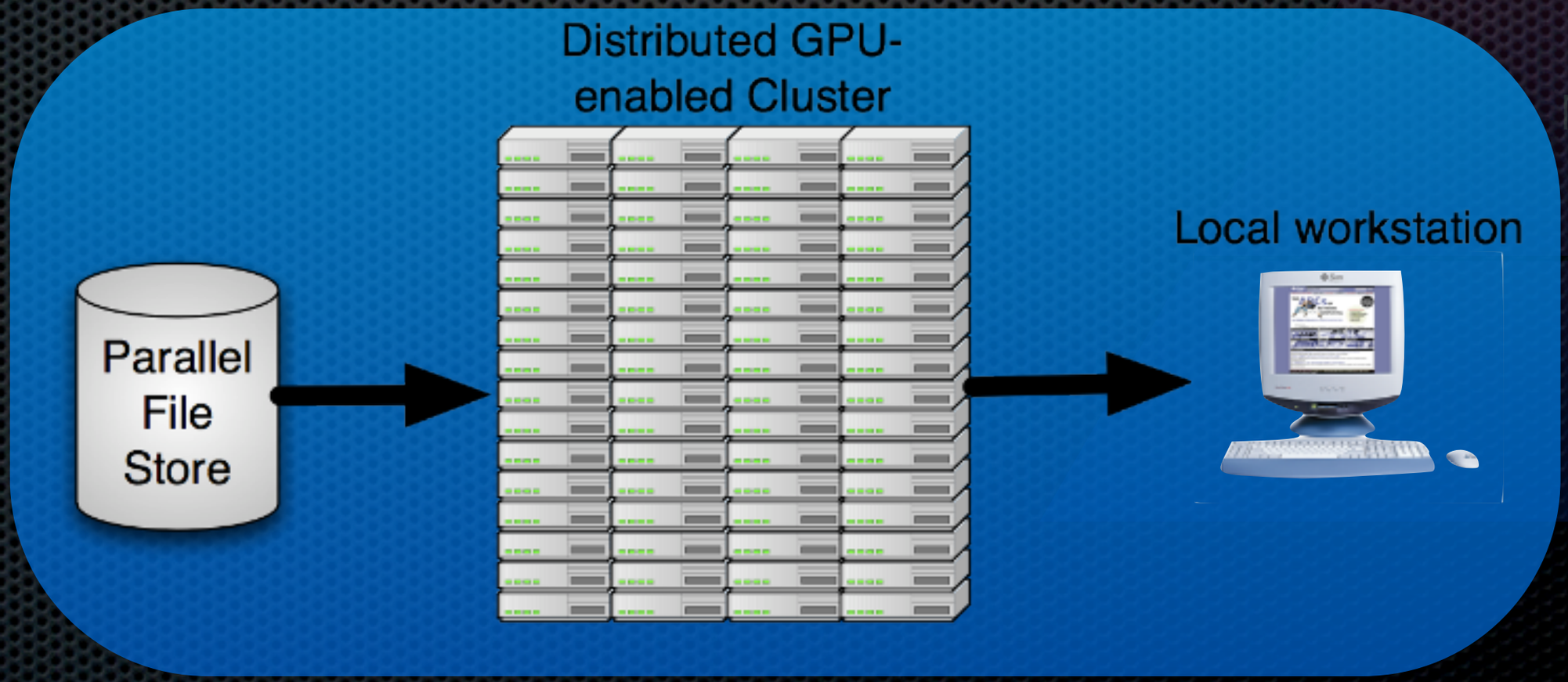
Data size categories

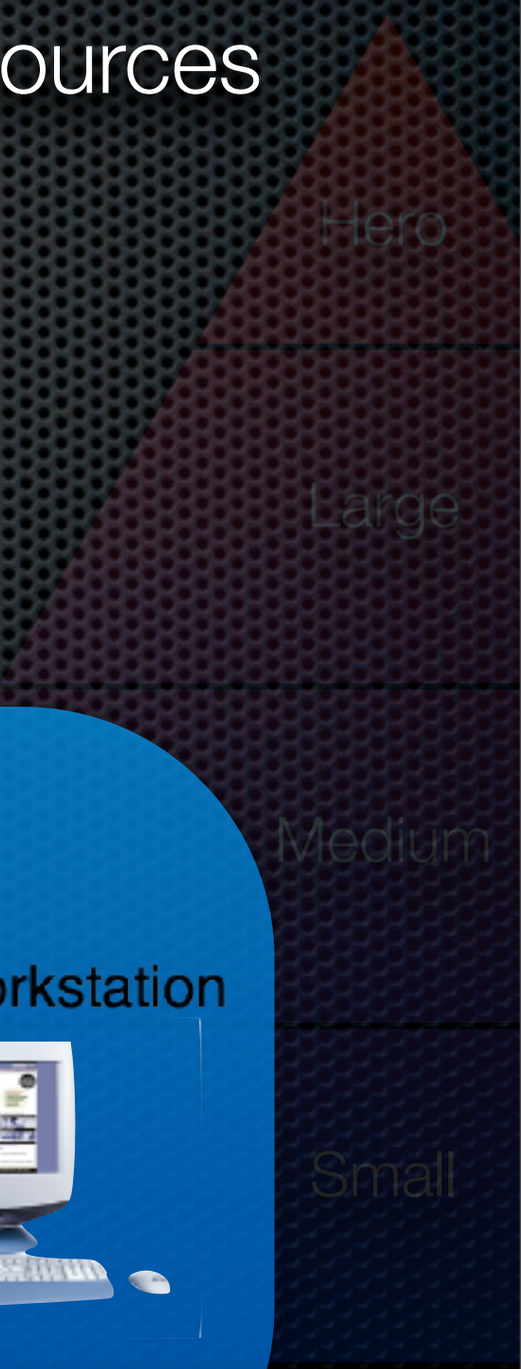
- Small:
  - Data is small enough to easily move anywhere
  - Analysis/vis is generally done on local workstations
- Medium:
  - Data won't fit on local workstations
  - Generally have to process on "fat" SMP systems
- Large:
  - Rather painful to move
  - Requires distributed parallelism
- Hero:
  - Functionally impossible to move
  - Only approachable on largest computational platforms



“Large” data – remote visualization

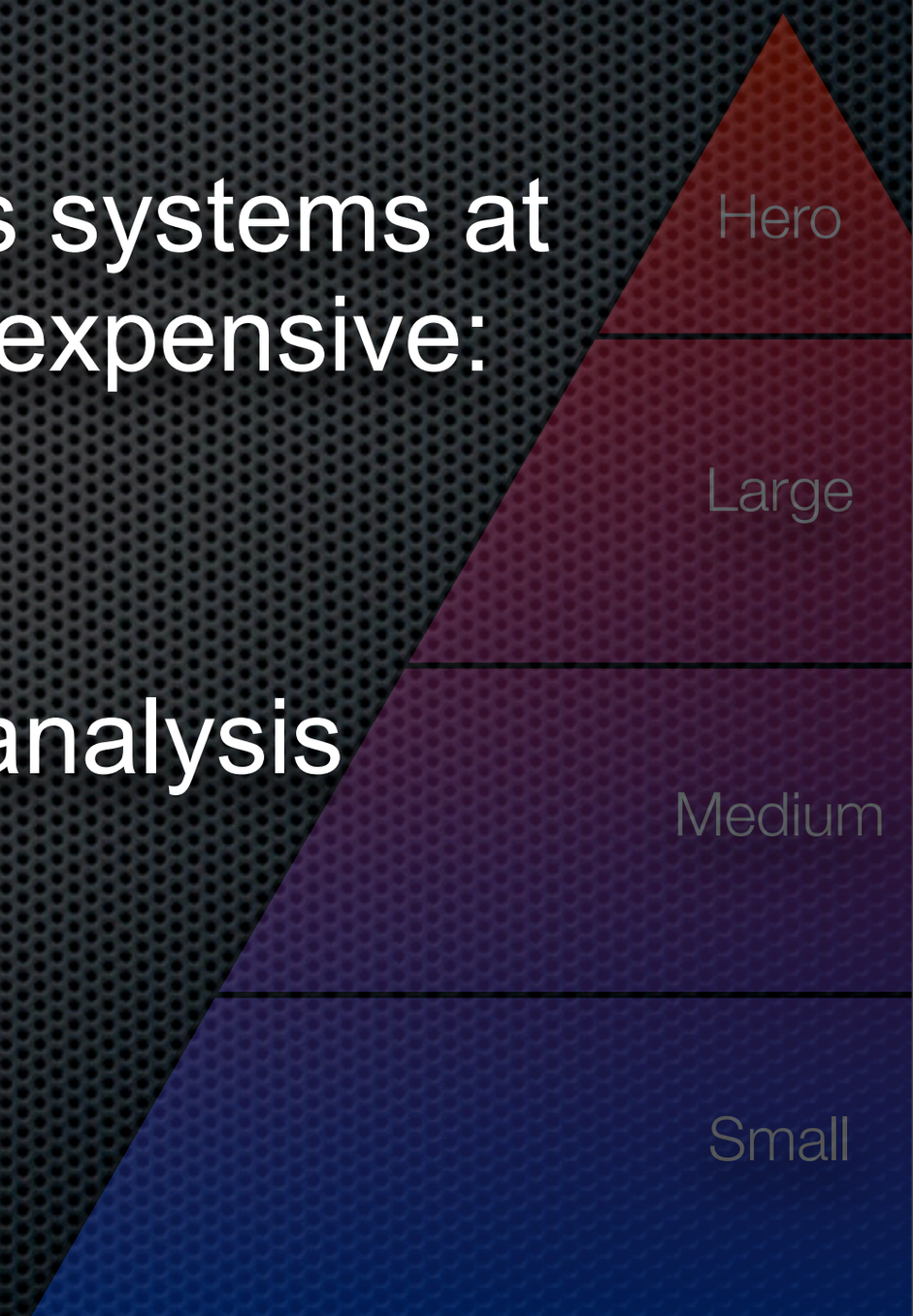
- Largest datasets require use of institutional resources
- Reduces data movement issues
- Allows exploitation of multiple GPUs
- Provides visualization to remote users
- Exploited by VisIt, ParaView, EnSight



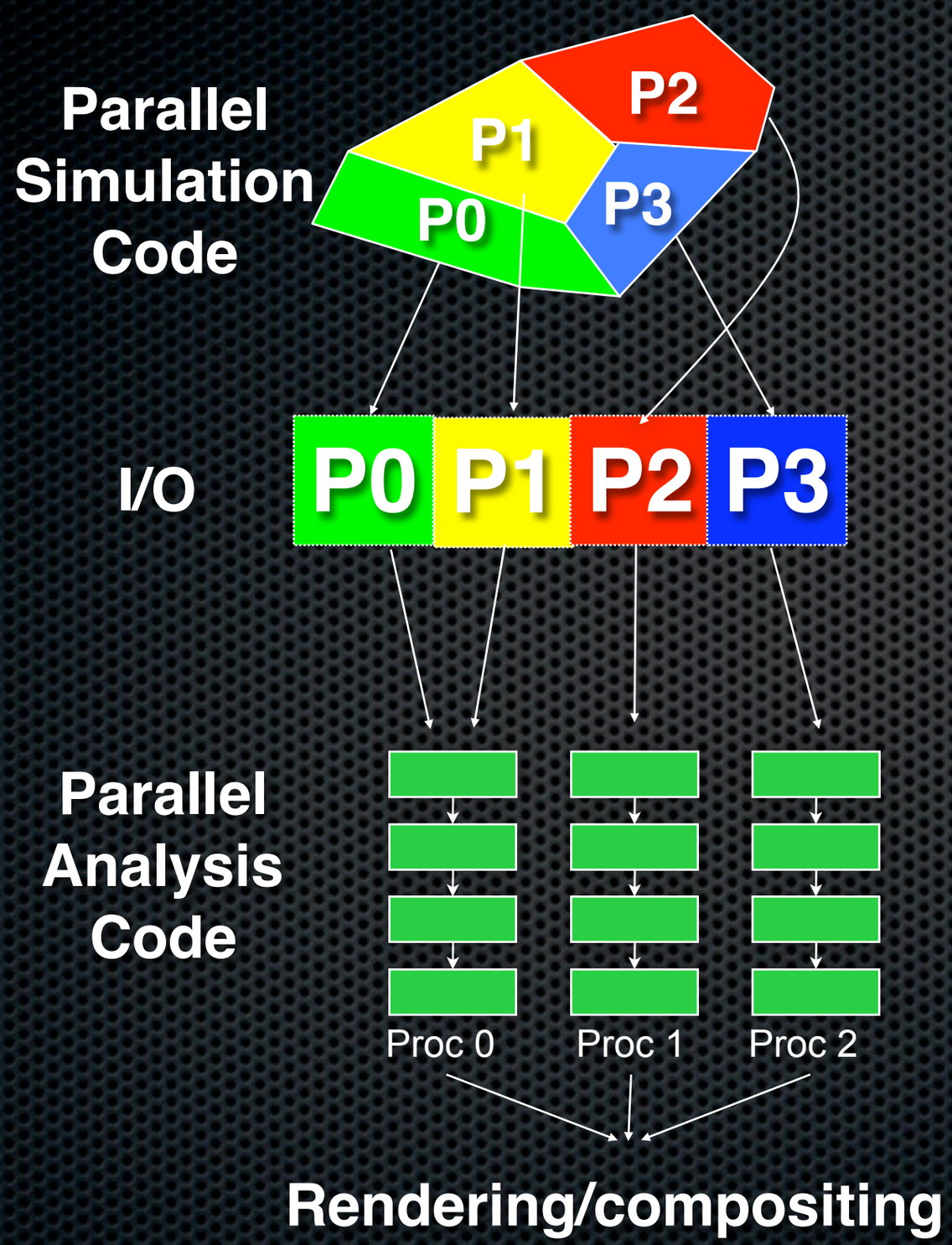


“Hero” data

- Purchasing separate analysis systems at the petascale is prohibitively expensive: *$10-30 million*
- Working to move largest vis/analysis tools to HPC architecture
  - *VisIt* on Cray XT4/5
    - See Cray User’s Group ‘08
  - *ParaView* on Cray XT4/5
    - See Cray User’s Group ‘09 (hopefully)



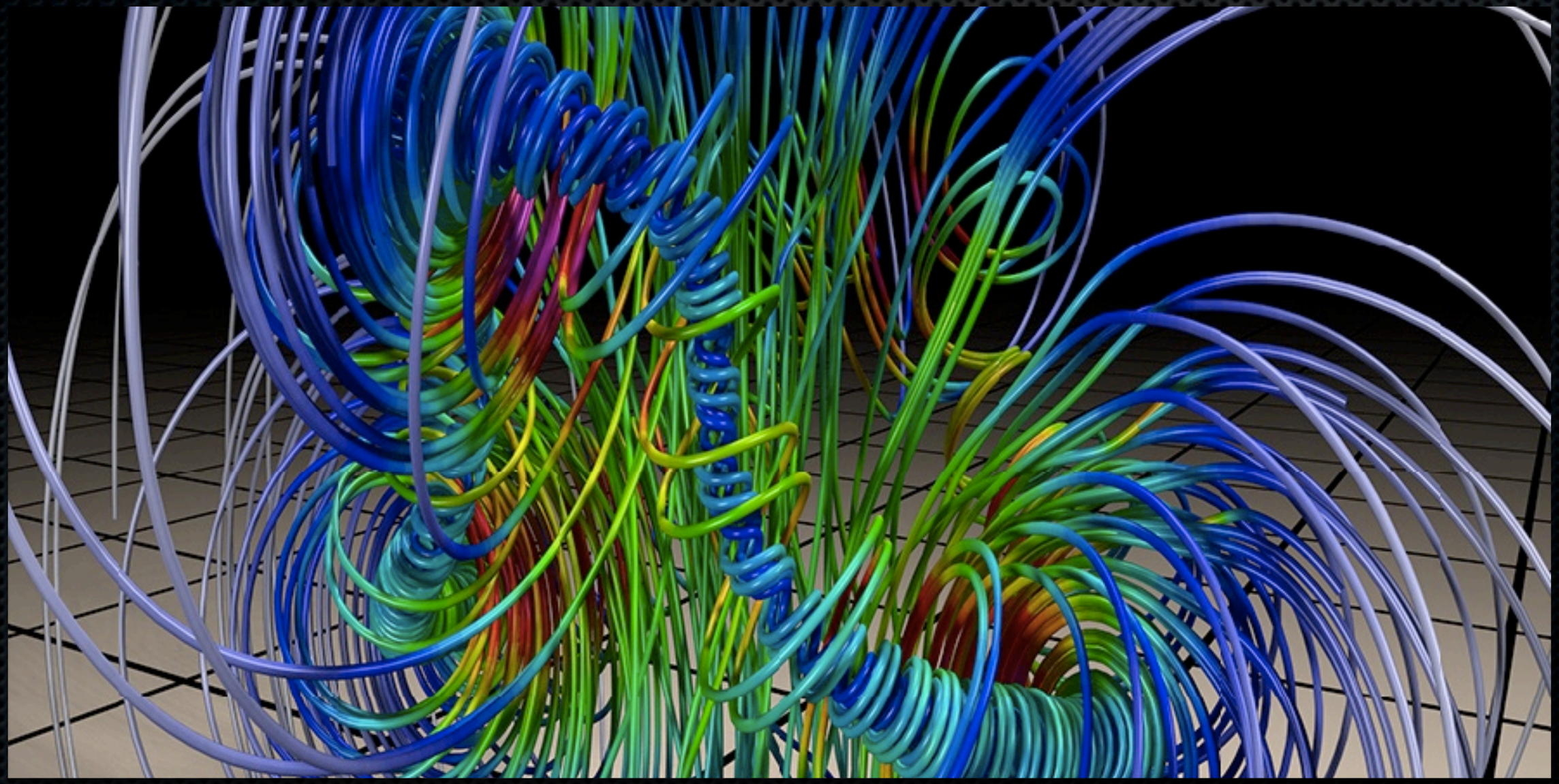
Data parallelization for all components



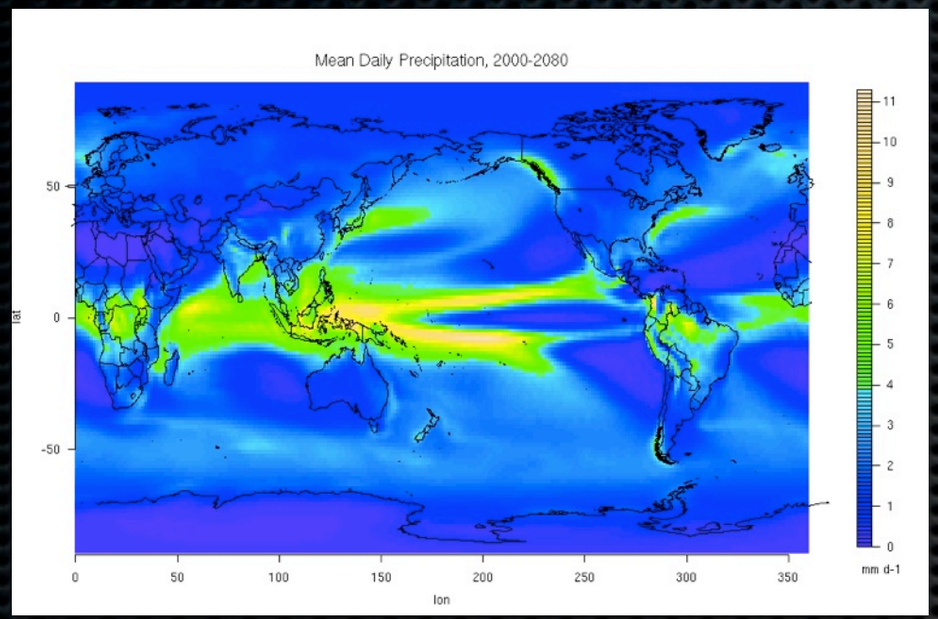
- Identical data flow networks on each processor.
- Networks differentiated by portion of data they operate on.
- "Scattered/gather"
  - No distribution (i.e. scatter), because scatter is done through choice of what data to read.
  - Gather: done when rendering
- Distribution of data is key

Data parallel streamlines/pathlines

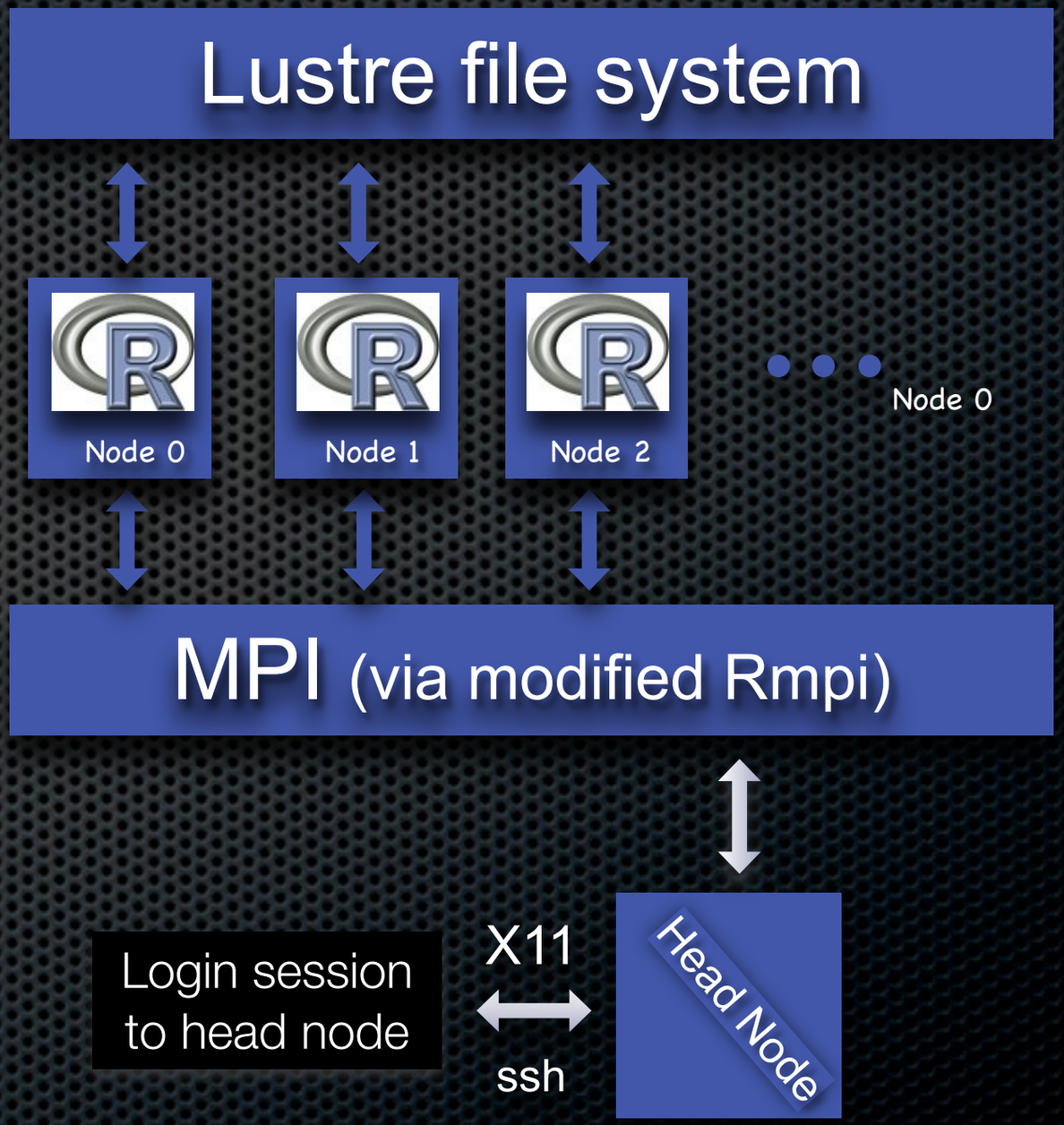
Provides understanding of large vector fields



Parallel Statistical Analysis



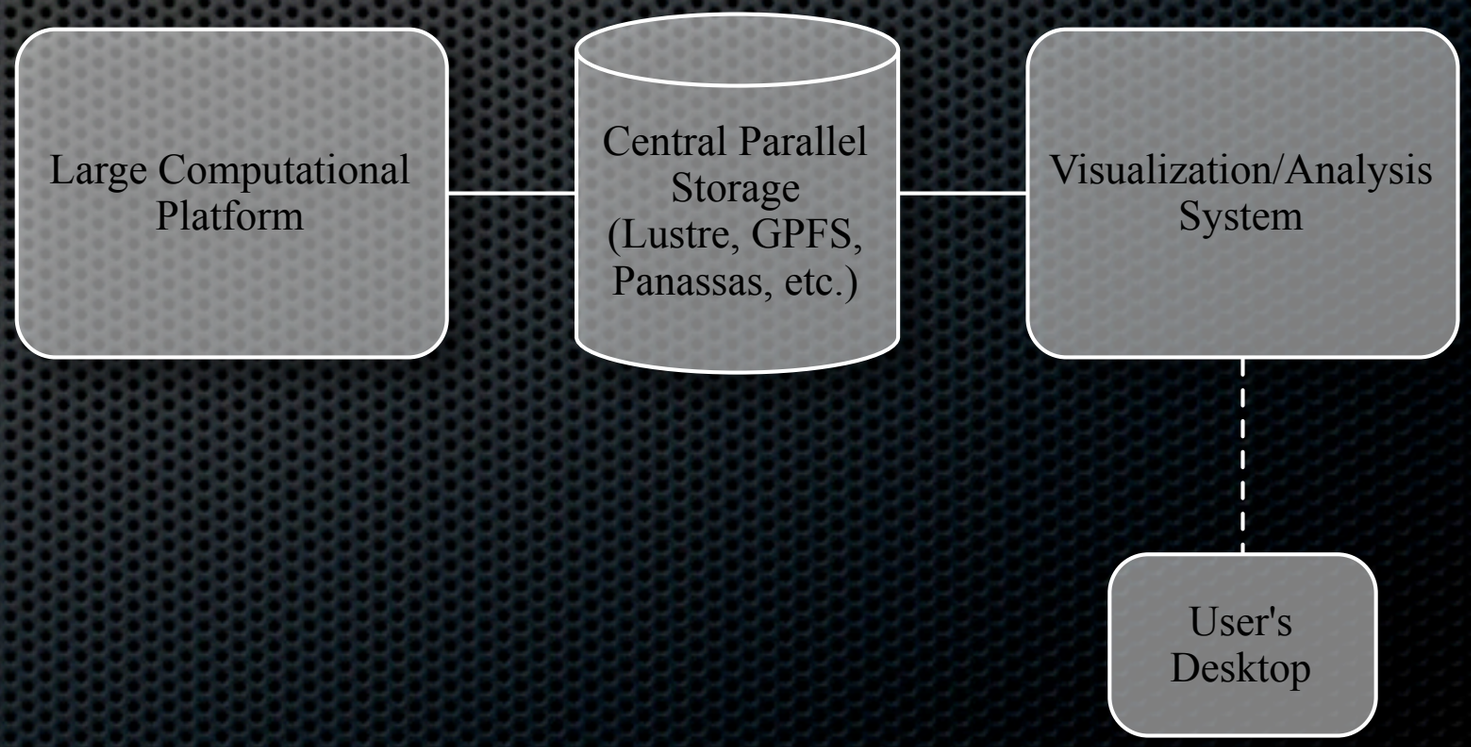
Command-line and graphics interaction with R on Node 0



- User writes data reader to bring data from parallel file system (NetCDF examples)
- User is aware of data distribution and develops data-parallel analysis methods with full capability of R on every node

Central file store

- Scalable parallel throughput
  - Lustre, GPFS, Panassas, ...
- Shared between source and destination
- Provides “zero copy” access to simulation results
- Allows easy deployment of remote visualization capabilities

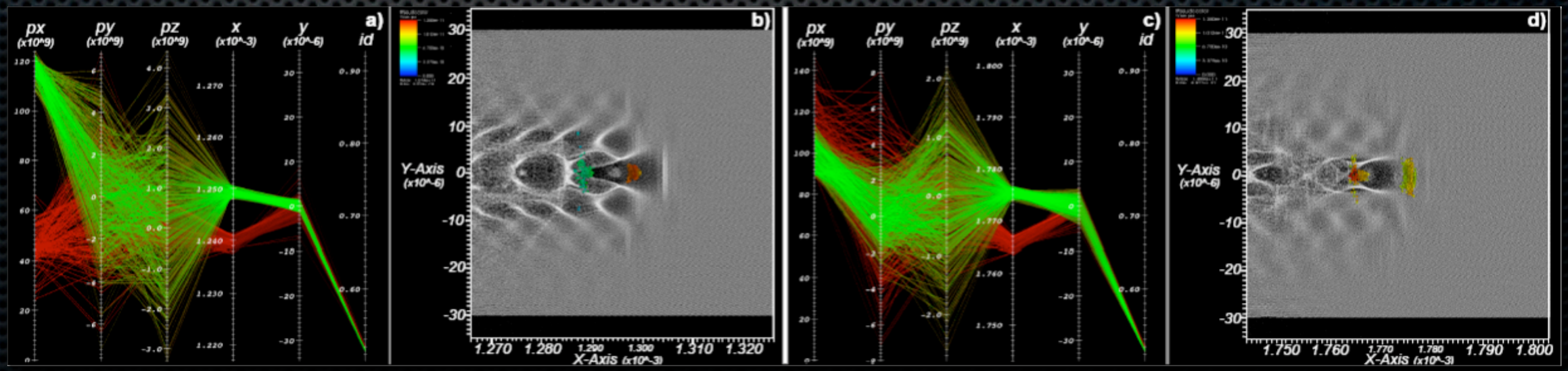


I/O management

- I/O is primary bottleneck to understanding
- Reducing “interactive” I/O is always a win
  - Pre-indexing data
  - Skip irrelevant data (slicing, view-dependence, data-directed)
  - Guided visualization

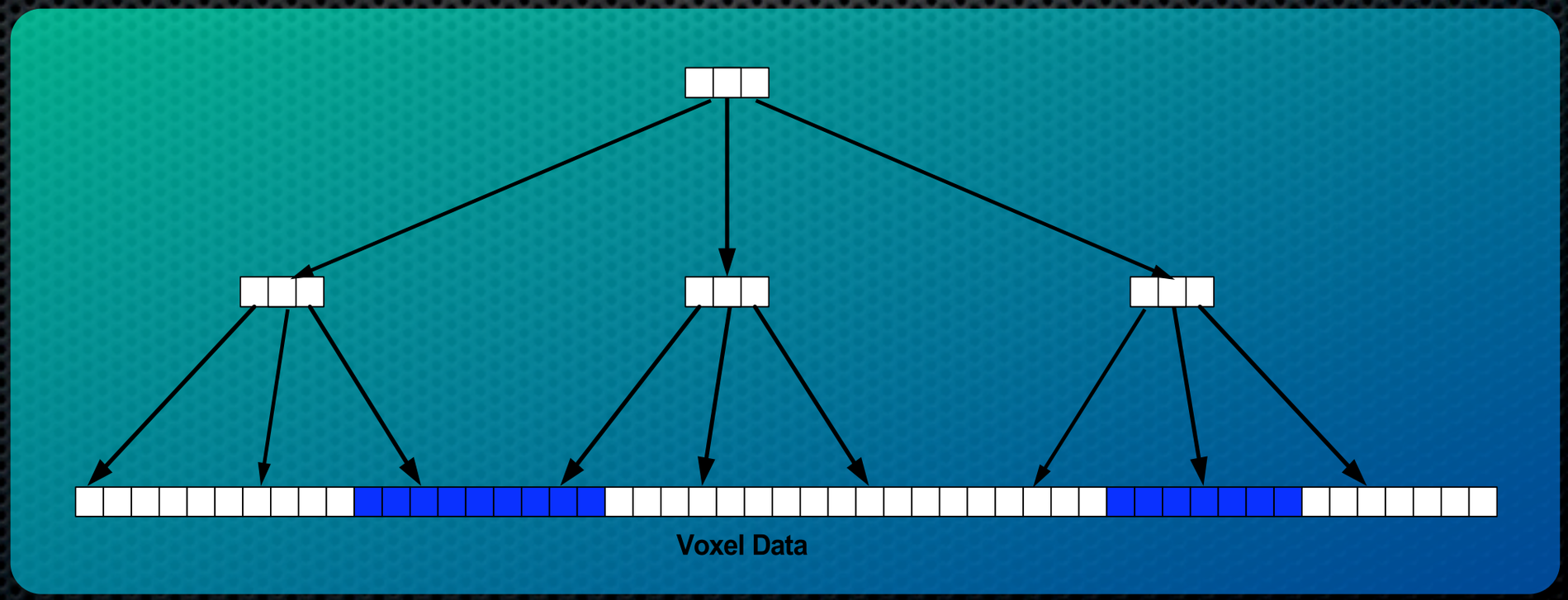
Bitmap indexing

- FastBit project from the SciDAC SDM Center.
- Accelerates interaction with laser wakefield simulations.
- Allows for interactive seeking of particles, with execution time linear in the amount of returned particles.
- *See SC08 paper.*



Fat tree data indexing

- Load-balancing among servers uses space-filling curves to break up locality within the dataset.
- Optimized data access by using a sparse search tree structure to skip irrelevant data items on top of a linear search.
- Discard unwanted data items upon distribution (data items are independent of any structural meta-information)
- Compress blocks of data items to trade memory space vs. access time, decompress on access.



Ultimately, it's about "understanding"

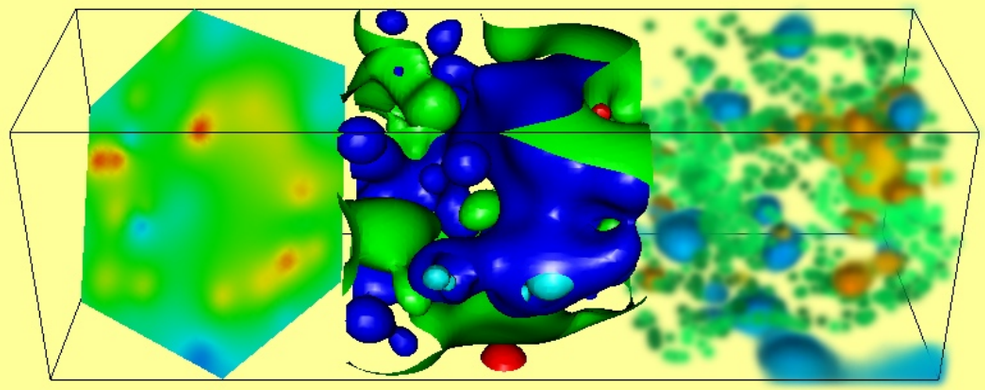
The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' (I found it!) but 'That's funny ...'

– *Isaac Asimov*

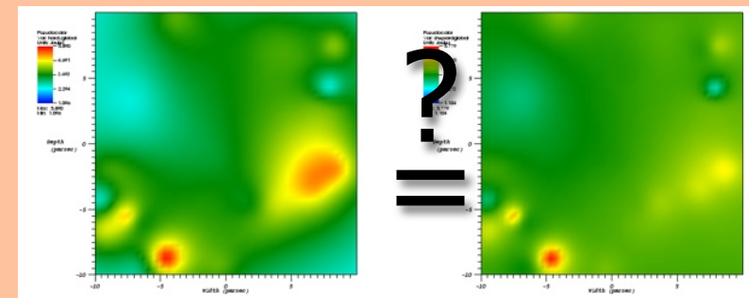
The purpose of computing is insight, not numbers.

– *Richard Hamming*

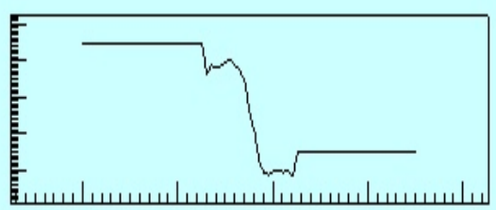
Analysis is more than just rendering pixels



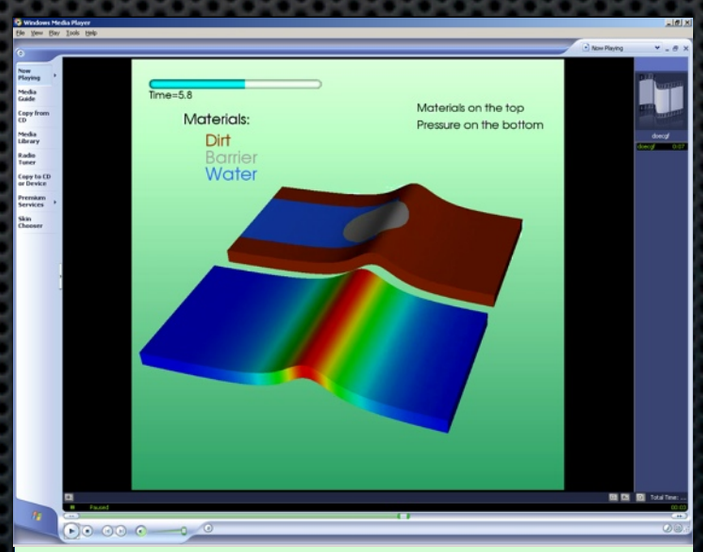
Data Exploration



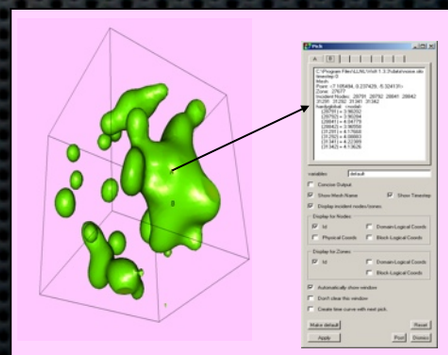
Comparative Analysis



Quantitative Analysis



Communication



Visual Debugging

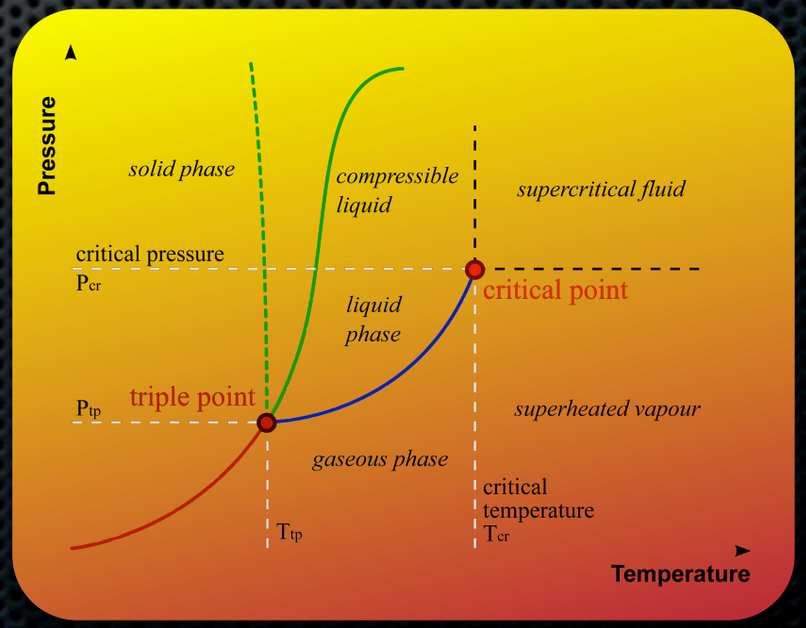
Data scale limits scientific understanding

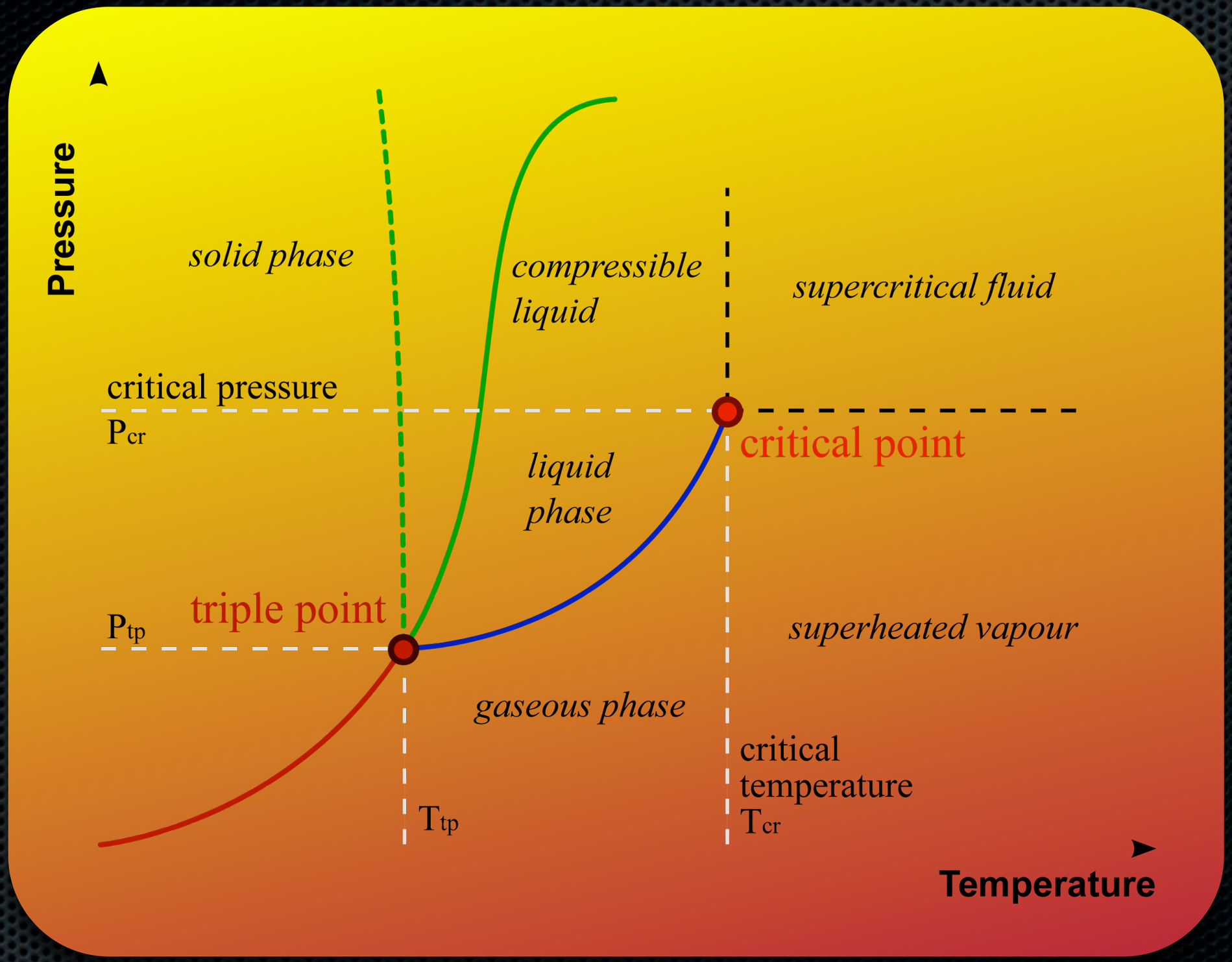
- Spatial resolution increasing
  - Too many zones to see on screen (without high-res Powerwall)
- Temporal complexity increasing
  - Manually finding temporal patterns is tedious and error-prone
- Multivariate overload
  - Climate simulations have 100-200 variables
- Issues of data models and domain-specific data
  - E.g. Multigroup radiation fields

Employ “Info Vis” methods for added insight and Guided Analysis

- Scatterplots
- Trees
  - Topology
  - Contour trees
- Histograms
  - Multidimensional
- Multivariate binning

- Parallel coordinates
- Provides more insight into multivariate relationships

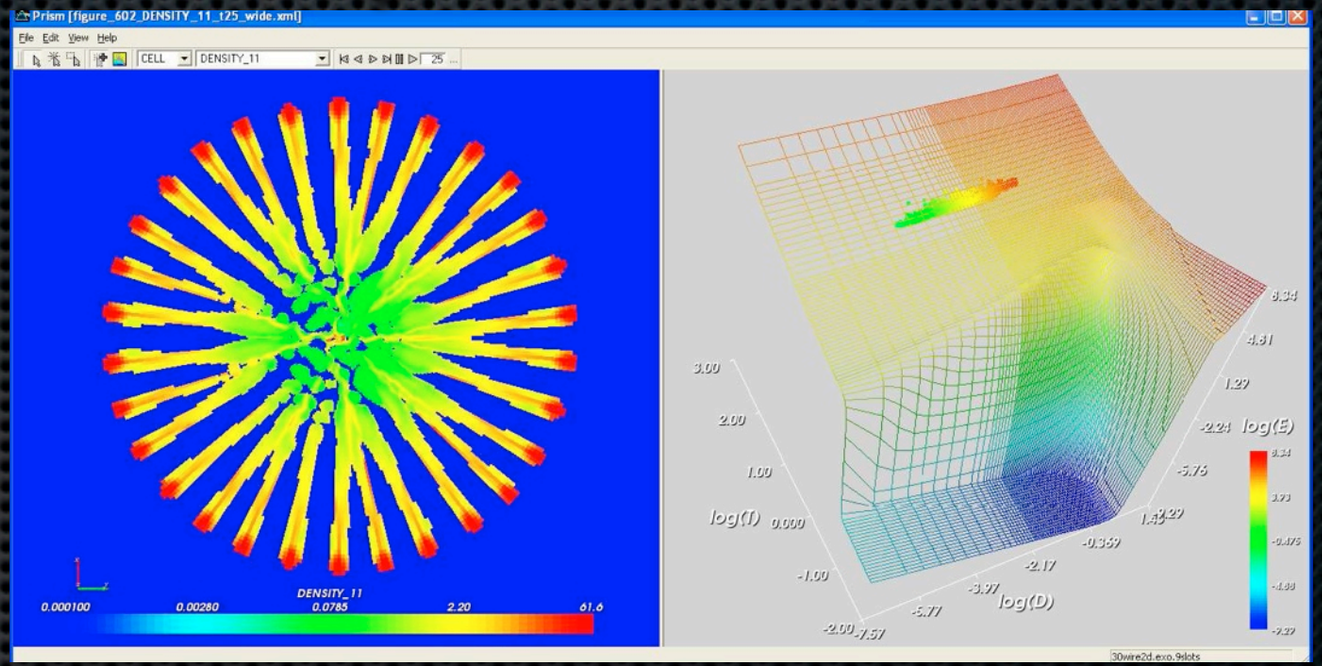


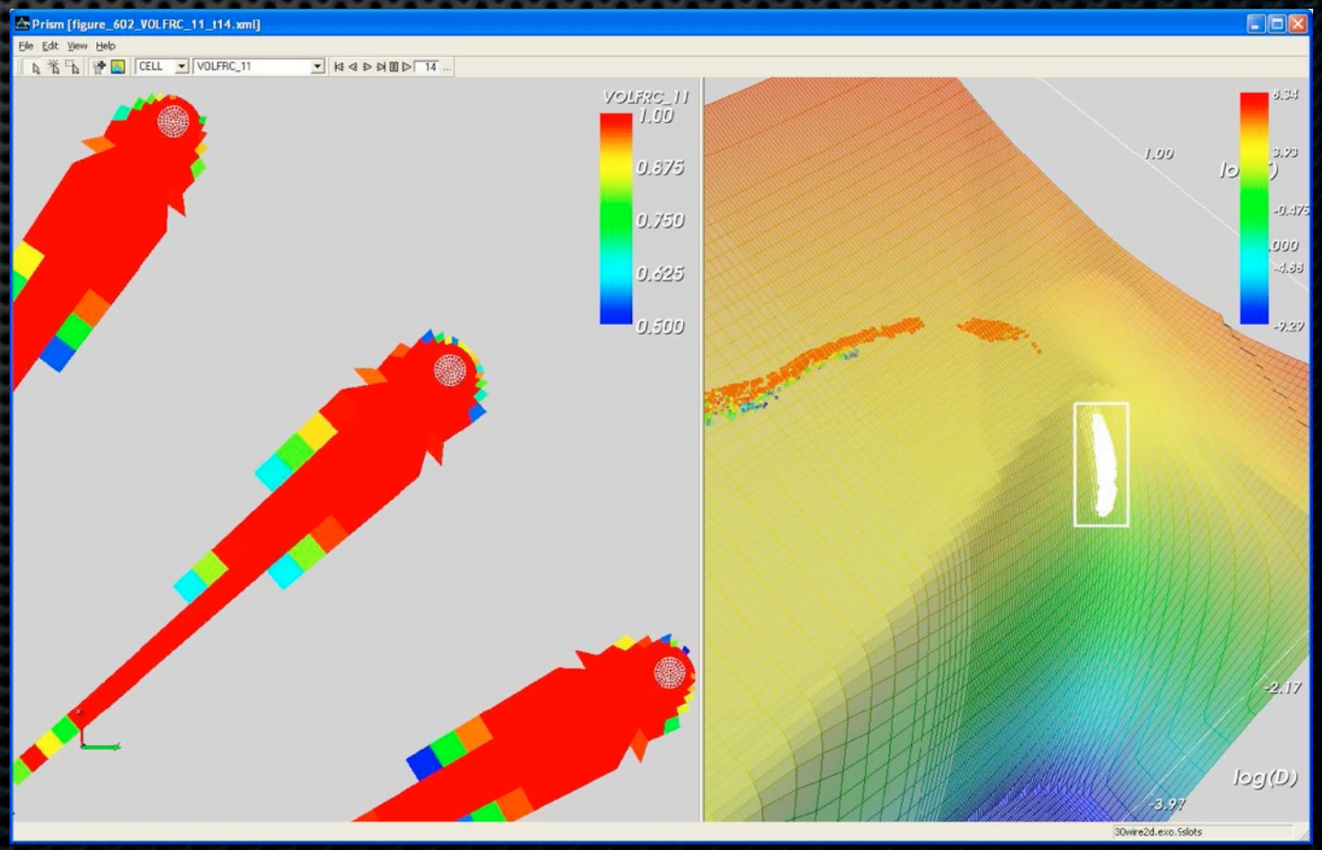


Example: Phase space

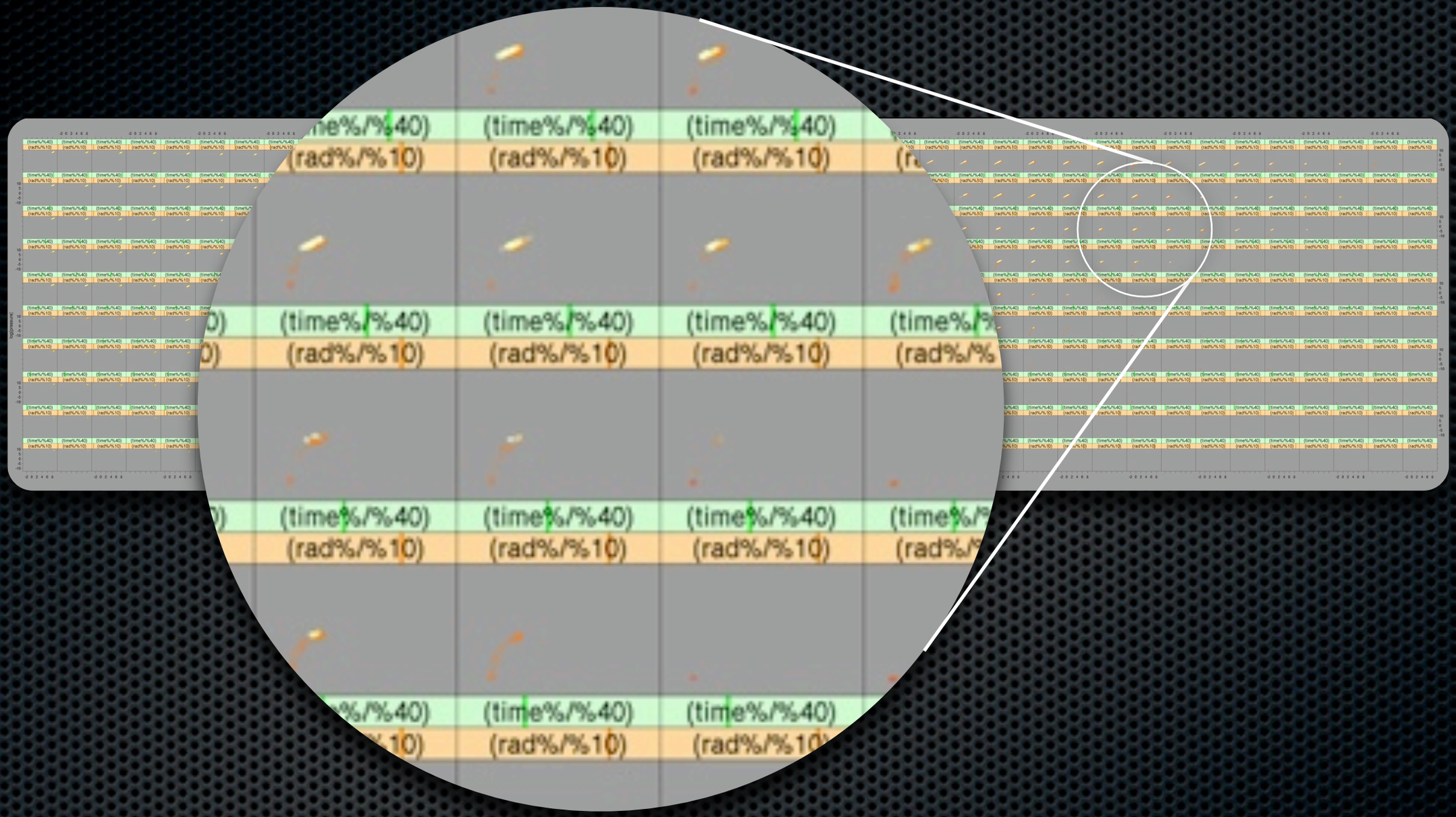
Example: Phase space

- Link together phase space depiction with traditional spatial visualization



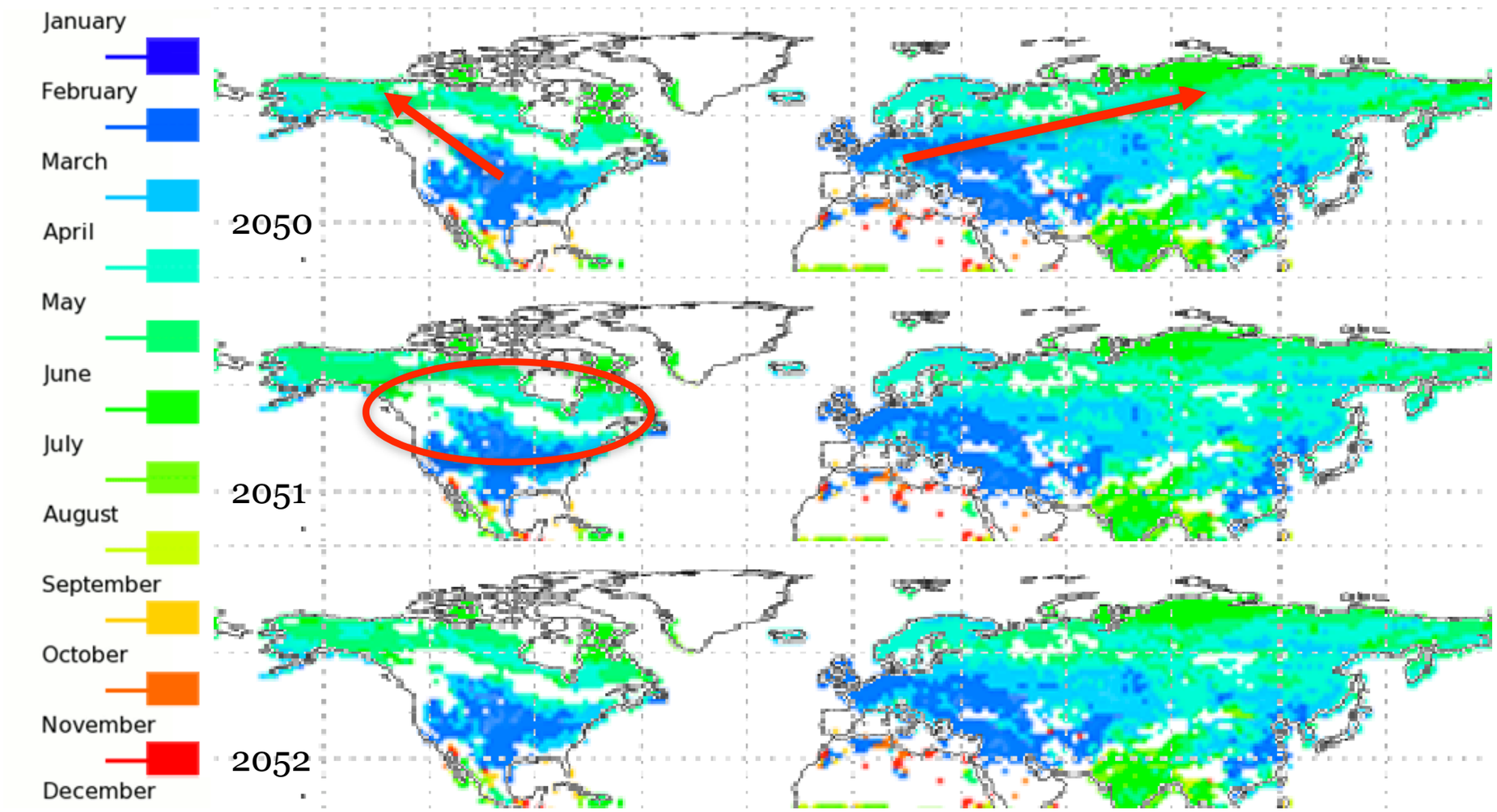


Courtesy David Rogers and Pat Crossno of Sandia



Higher dimensions

- Supernova simulation
- 5 variables plotted at once
- Radius, time, pressure, density, entropy

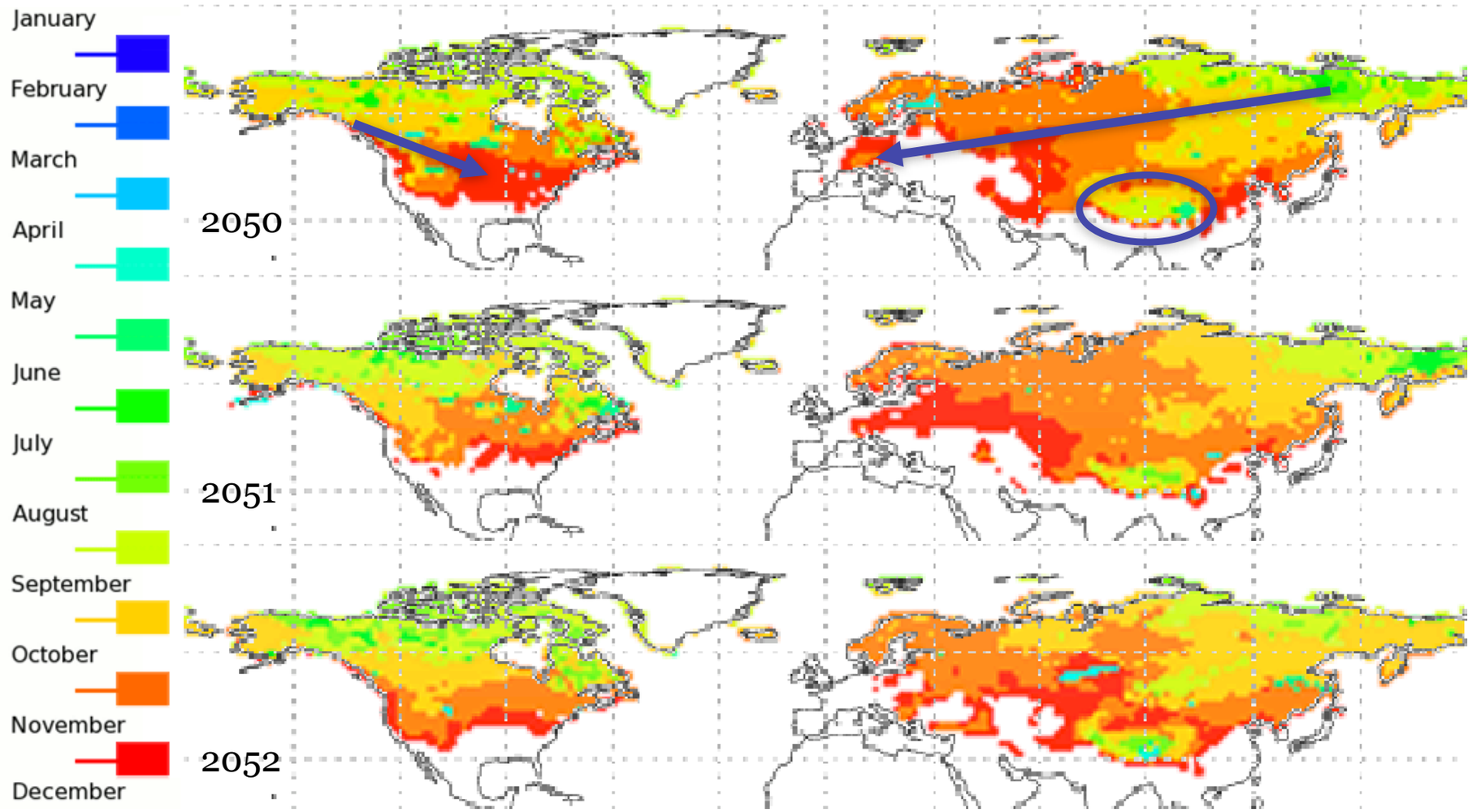


"Green-Up": Northern Hemisphere colored by month of event in variable ELAI

Query: `ELAI:[-.4-.4]*T[.4-max]?*`

Temporal summaries

- See IEEE Vis 08 paper



"First Snow": Northern Hemisphere colored by month of event in variable FSNO

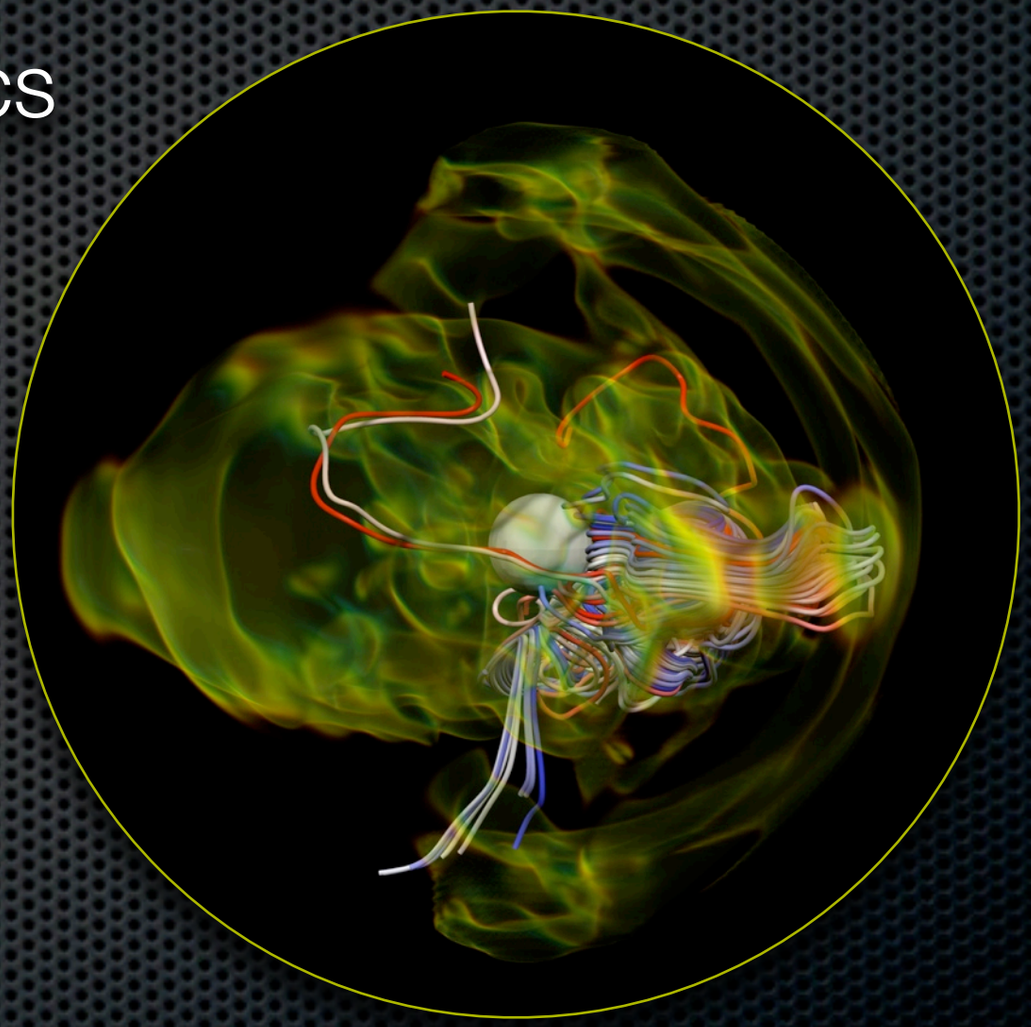
Query: `FSNO:???[min-.07]*T[.07-max]?*`

Temporal summaries

- See IEEE Vis 08 paper

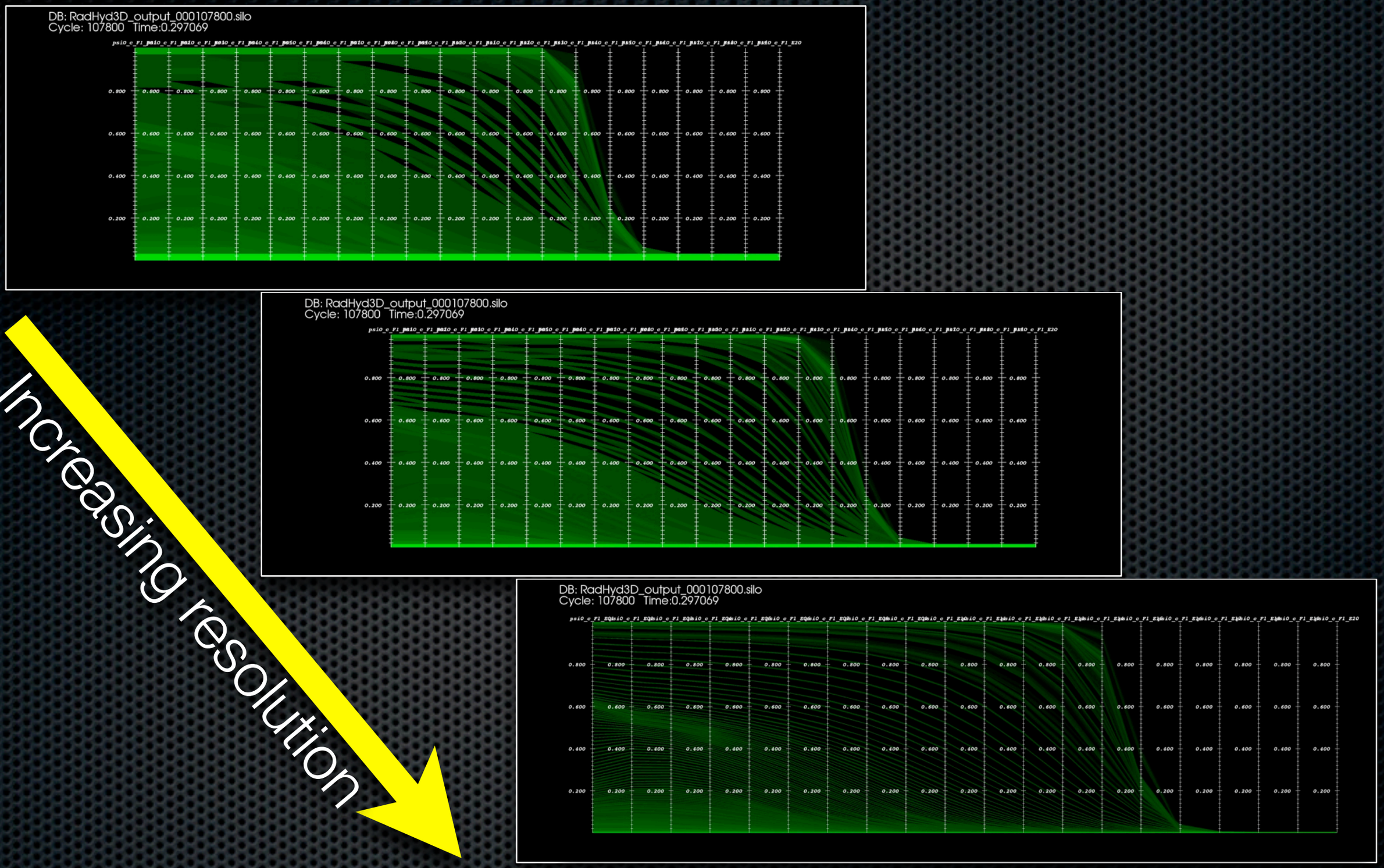
Supernova neutrino radiation field

- Historically, just hydrodynamics
- Now:
  - Magnetic field (vector field)
  - Neutrino radiation
    - Multiple neutrino “flavors”
    - Anisotropic distribution
    - 20 radiation “energy groups”



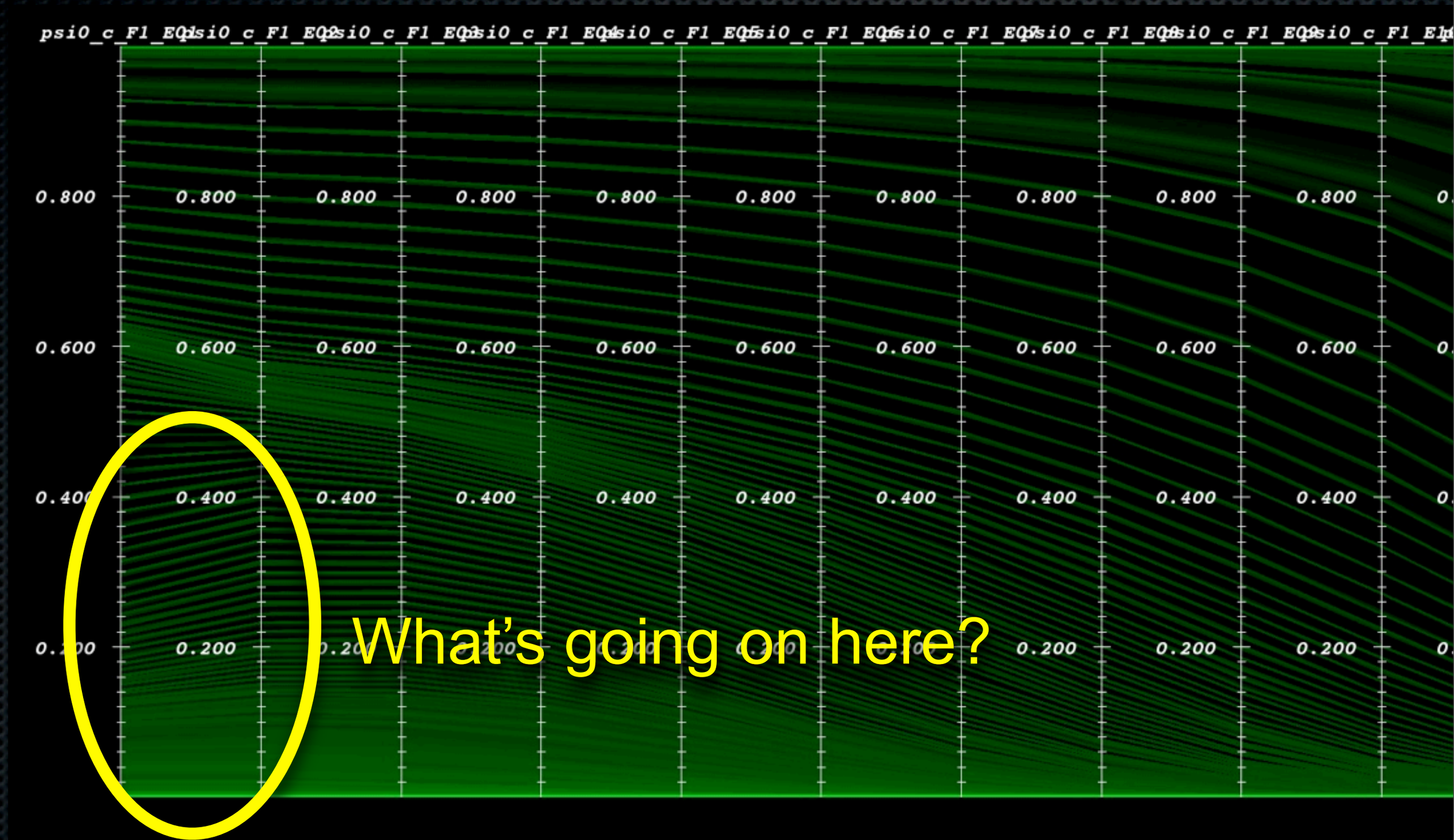
$$f(r, \theta_1, \phi_1, t, \theta_2, \phi_2, flavor, \lambda) = occupancy$$

Eight discretized “spatial” dimensions!



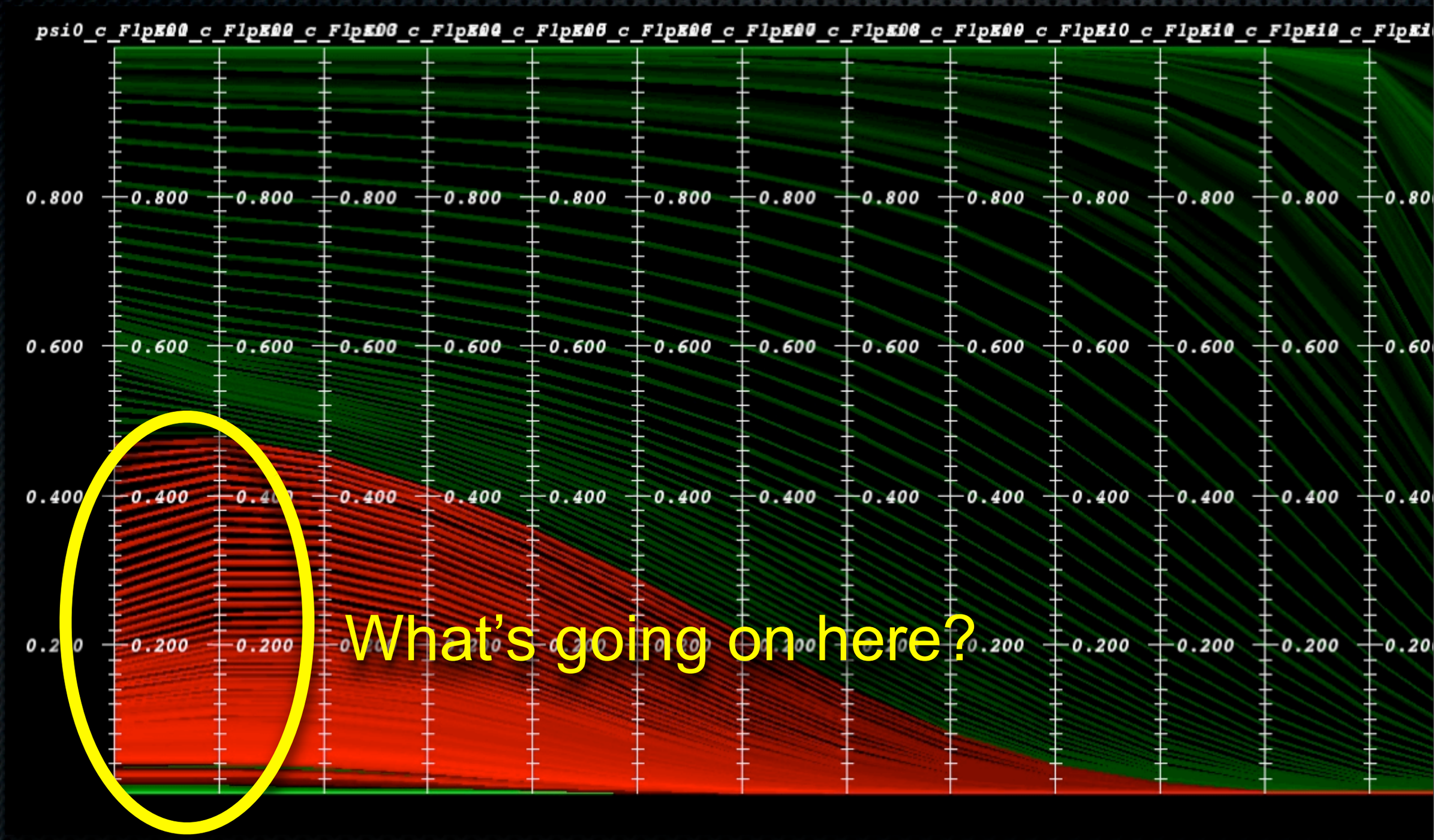
Parallel coordinates summary view

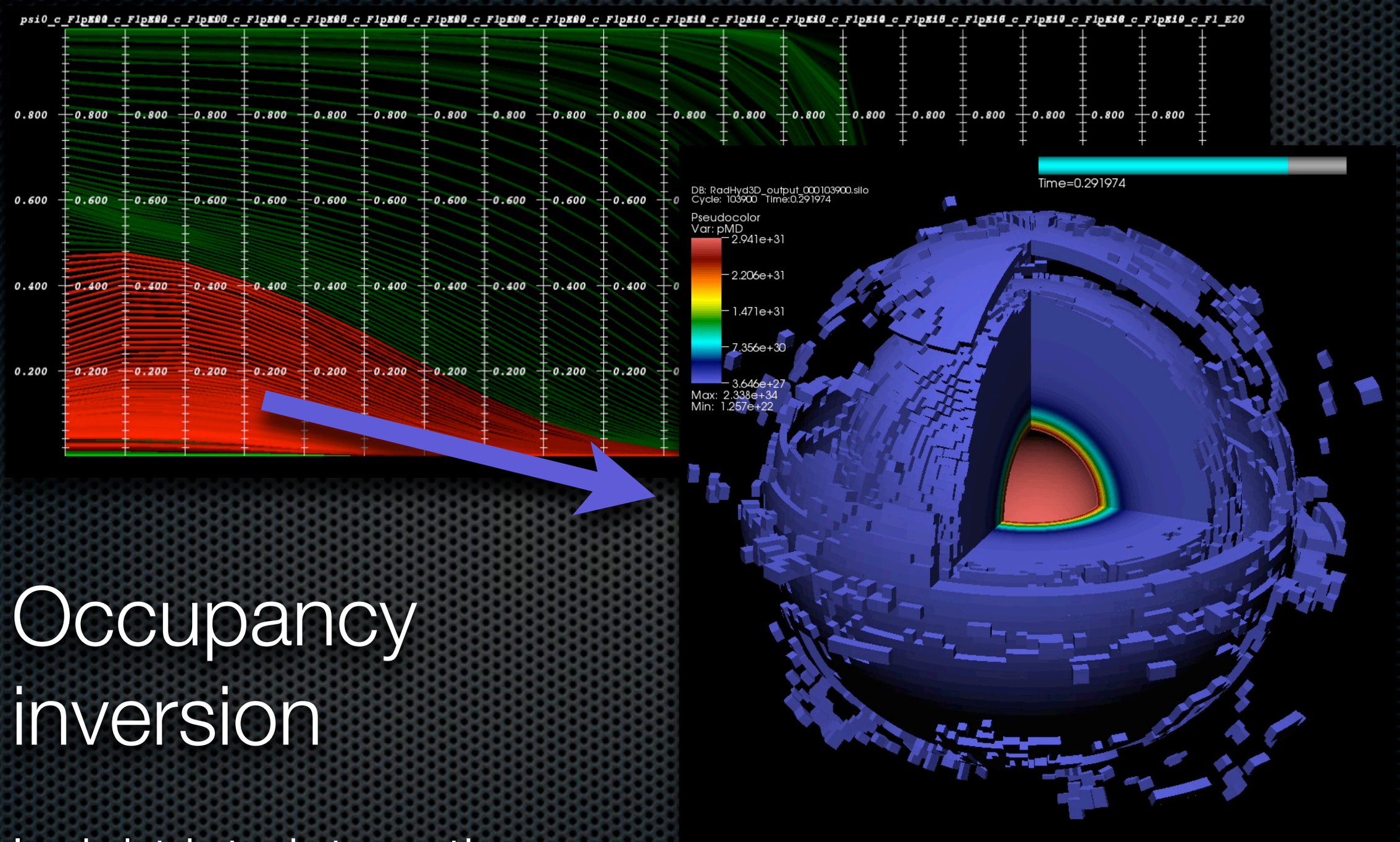
Parallel coordinates summary view



What's going on here?

Parallel coordinates summary view



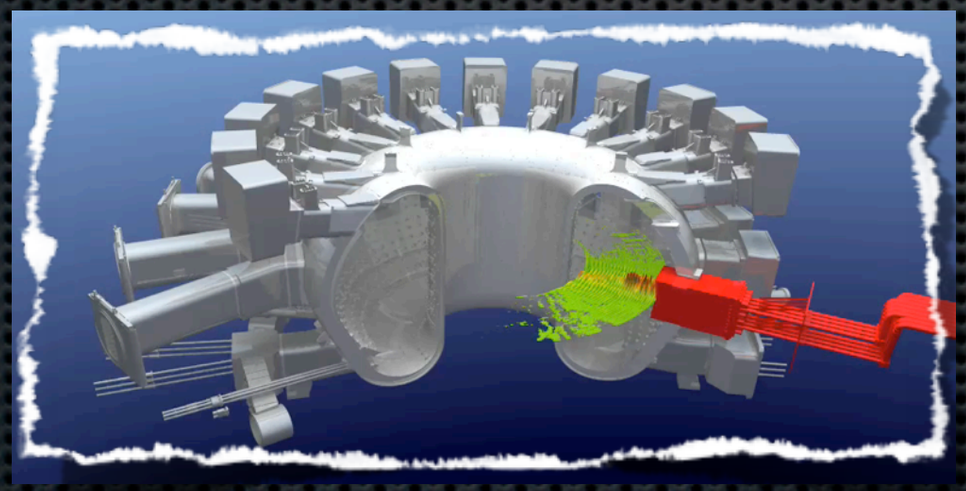


Occupancy inversion

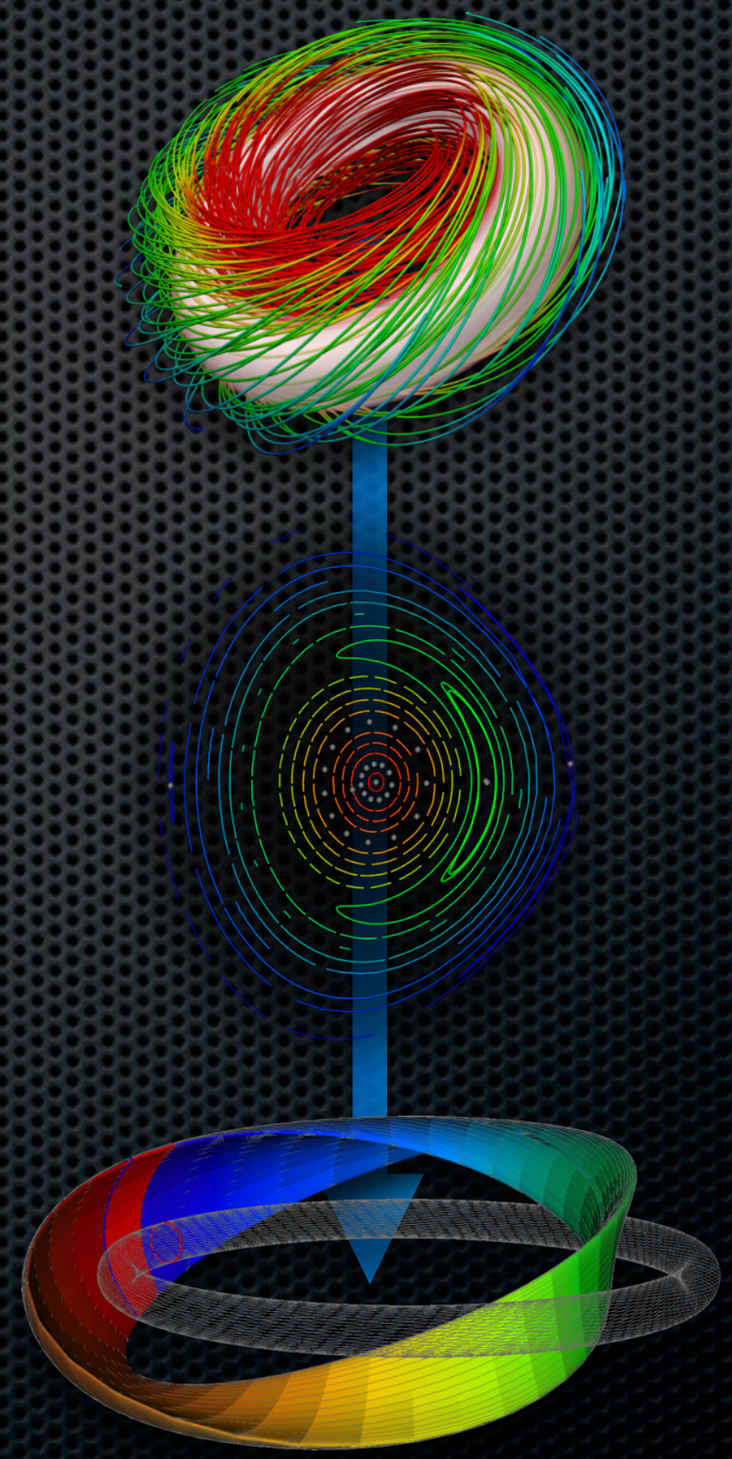
Insight into interesting calculation feature

Magnetically Confined Fusion

- Complex magnetic fields for confinement
- Topological analysis
  - Magnetic field line winding
  - Poincaré plot generation
  - Island extraction



Combined visualization of simulation results and CAD model of ITER



HPC Implications

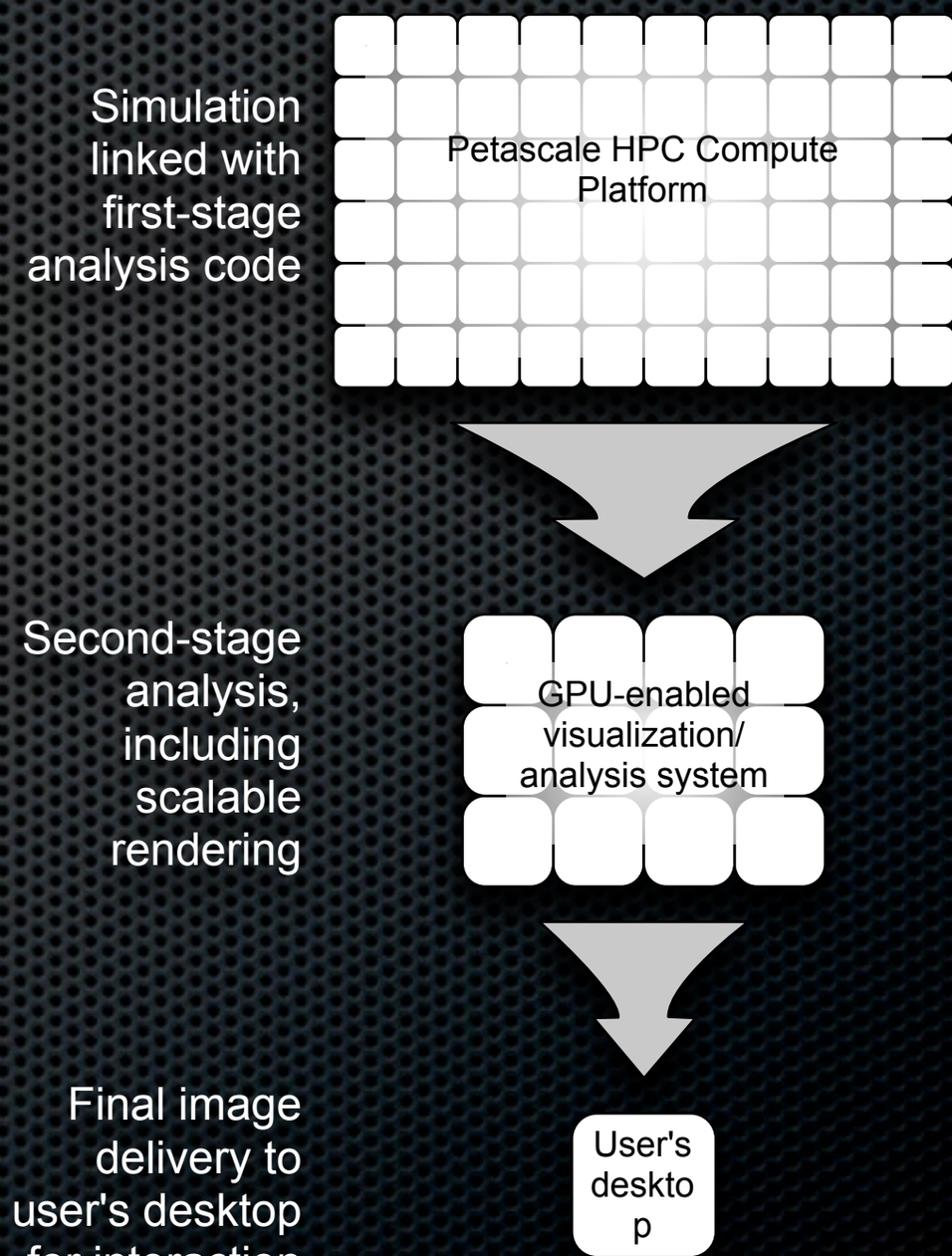
- Many techniques straightforward to parallelize:
  - Histograms
  - Statistical binning
  - Scatterplots
  - Parallel coordinates with 2D histograms
  - Rendering (see Jim Ahrens' talk)
- Others need more work:
  - Compositing
  - Multiaxis parallelization (spatial, temporal, multivariate)
- I/O optimizations scalable to HPC systems:
  - FastBit, fat trees
  - Leverage I/O architectures for data reduction

HPC Implications

- Can run bulk of analysis on HPC systems
- GPUs are of secondary importance
- Interactivity requires rapid queue turnaround
- Bursty I/O reads is a different mode of operation

Future work

- Deploying and harnessing “three-tiered” institutional infrastructures.
  - Direct coupling to sim
  - “In-situ” analysis
- Workflow management
  - Entire process (compute, storage, analysis, archiving) – Kepler
  - “Packaging” analysis capability (VisTrails)
- Alternate client technologies
  - Dashboards/portals
  - Web-based delivery



Session “take-home” messages

- I/O is our primary gating factor.
- “Vis clusters” are not sufficient for the petascale.
- It’s much more than just rendering.
- Analysis is very domain-specific.
- Pushing analysis/vis to the HPC system requires policy changes.