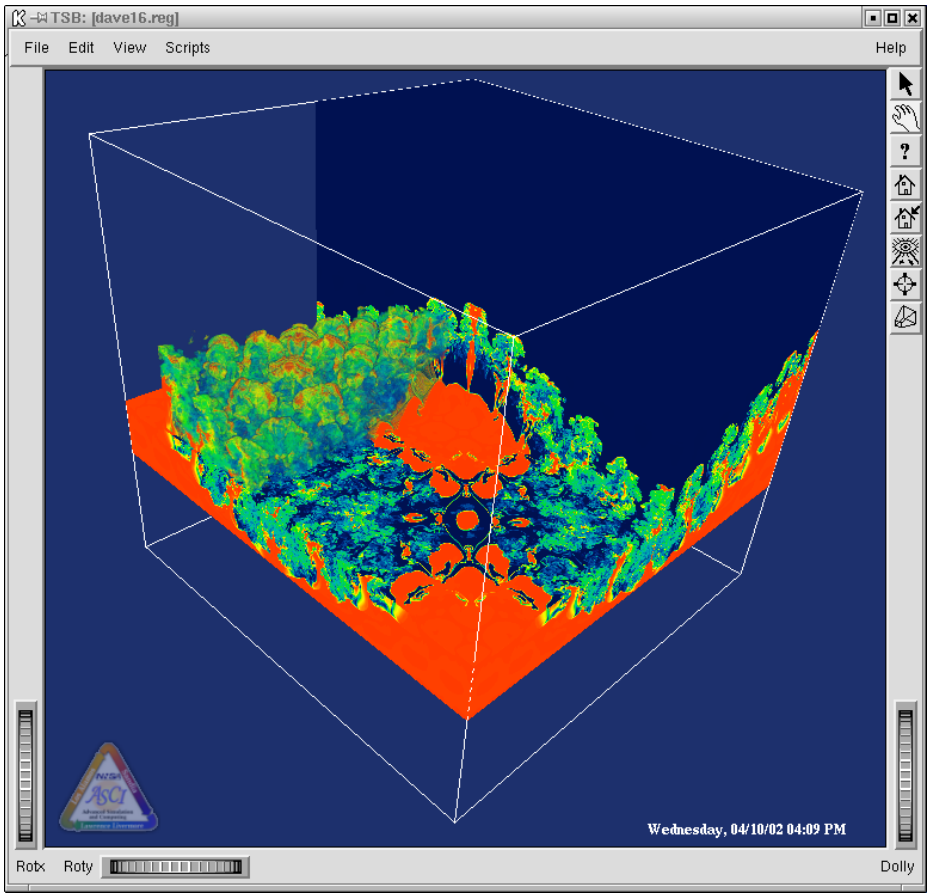
Real-time Monitoring of Large Scientific Simulations

D. E. Laney
V. Pascucci, R. J. Frank, G. Scorzelli,
L. Linsen, B. Hamann, F. Gygi

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

We would like to enable real-time monitoring of a running simulation with a desktop workstation.



Turbulent mixing:

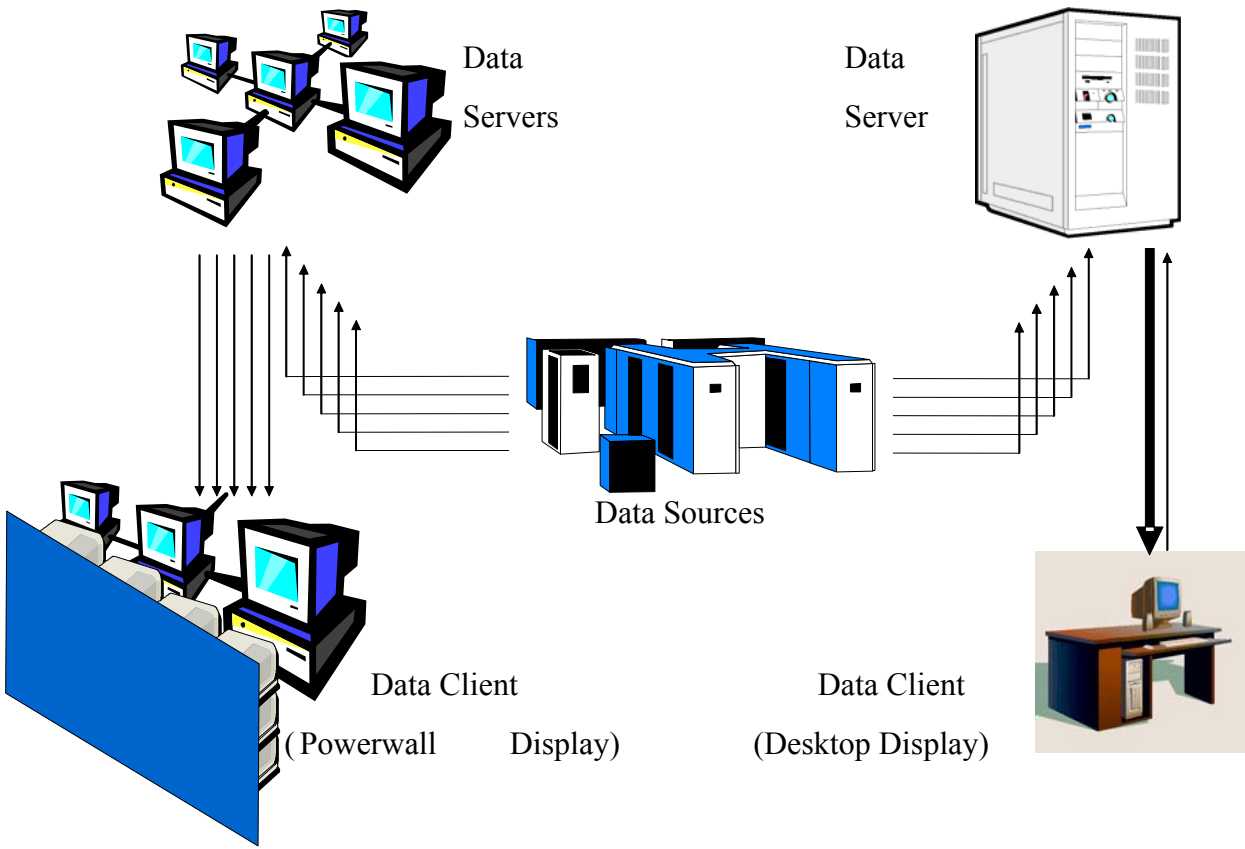
2048 ^ 3

Data post-processed for visualization

Two ideas are combined in our system.

- Multiresolution re-ordering of the data
- Real-time streaming of the reordered data.
- Monitoring via visualization of data or derived-data.

A data streaming framework must utilize a heterogeneous computational environment.



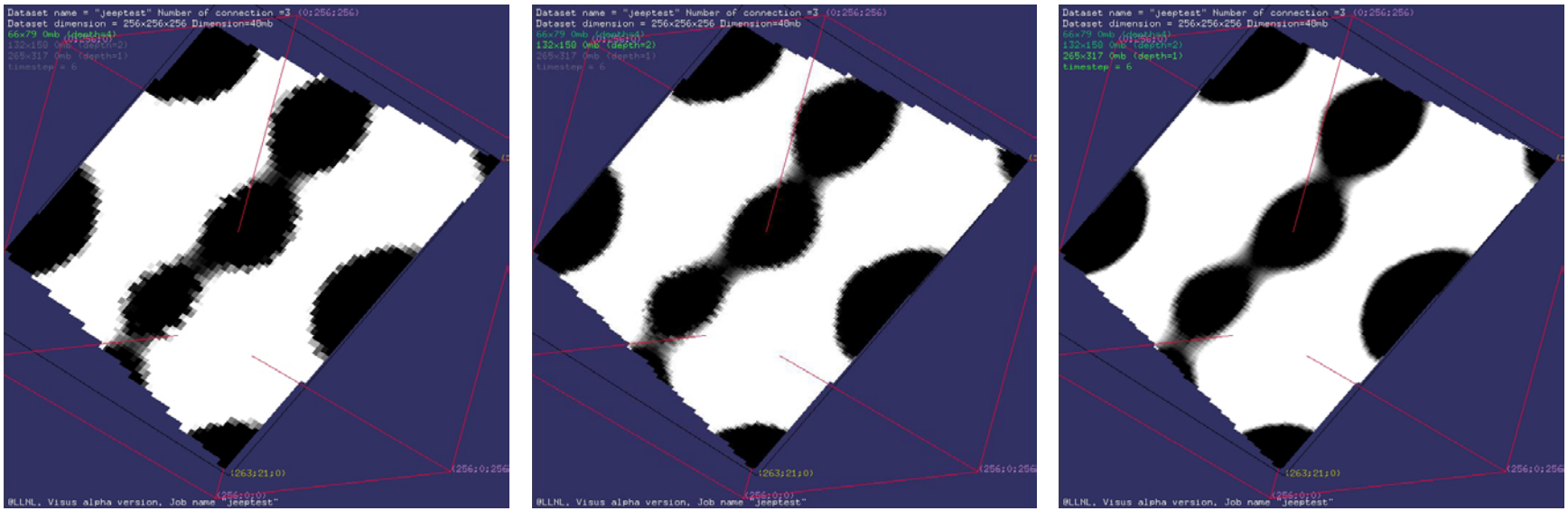
A brute force approach: send the data in the order it occurs in the simulation.

- Disadvantages
  - Bad cache coherence will slow down lower dimensional queries (slices, isosurfaces).
  - Stopping the stream early results in an unusable partial data set.
  - Requires preprocessing and delay if scalable visualization is required.

We propose a new multiresolution streaming system.

- Light weight:
  - Only requires init() and send() calls.
  - Not a problem solving framework
- Cache-oblivious out-of-core processing and data streaming.
- Simple, scalable infrastructure.

Our approach enables a coarse-to-fine construction of multi-resolution models.

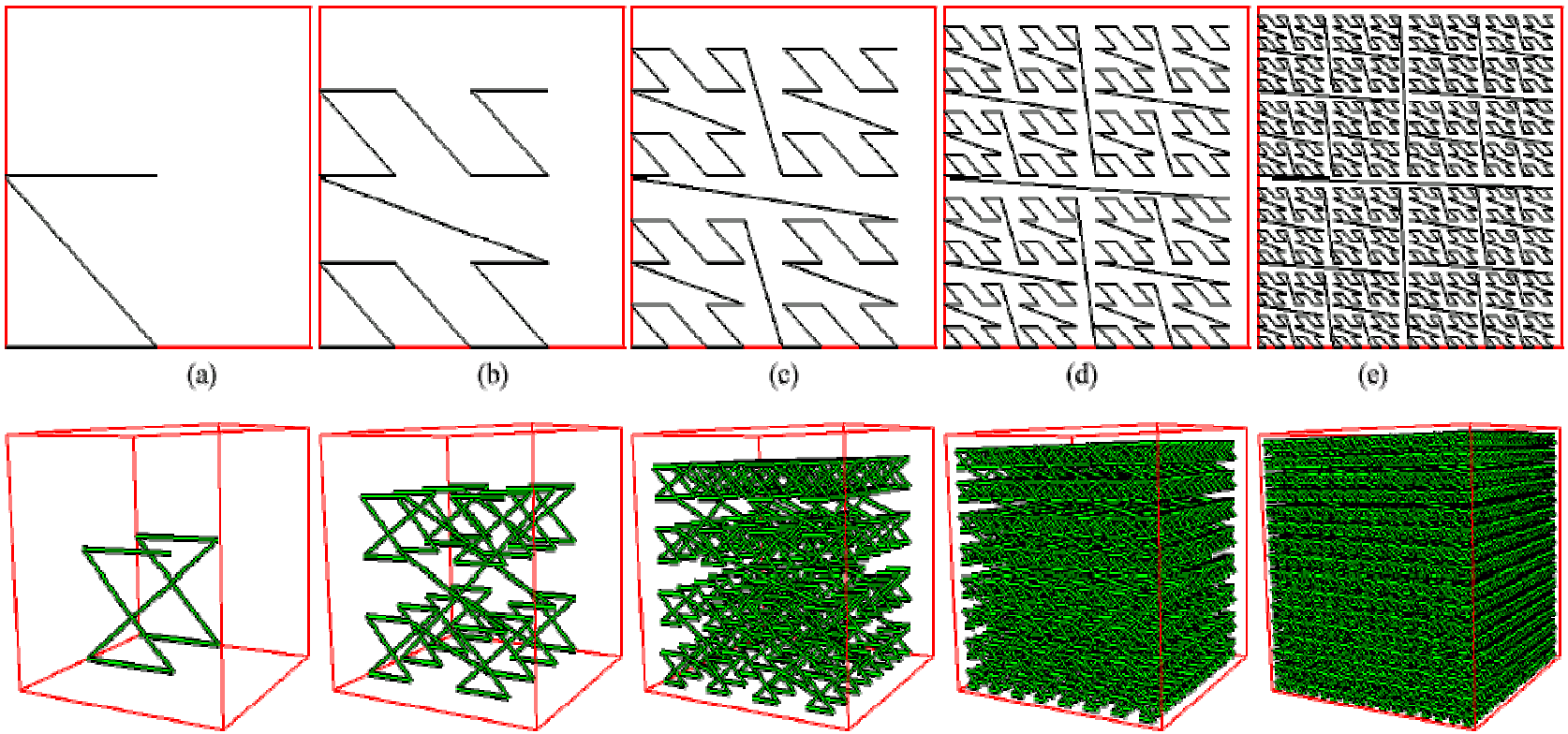


Jeep run (Francois Gygi) of 64 carbon atoms, tiled to 256^3 (final iteration). Computation on AIX system, reordered data cached on SGI server, viewed on my workstation (200MHz SGI).

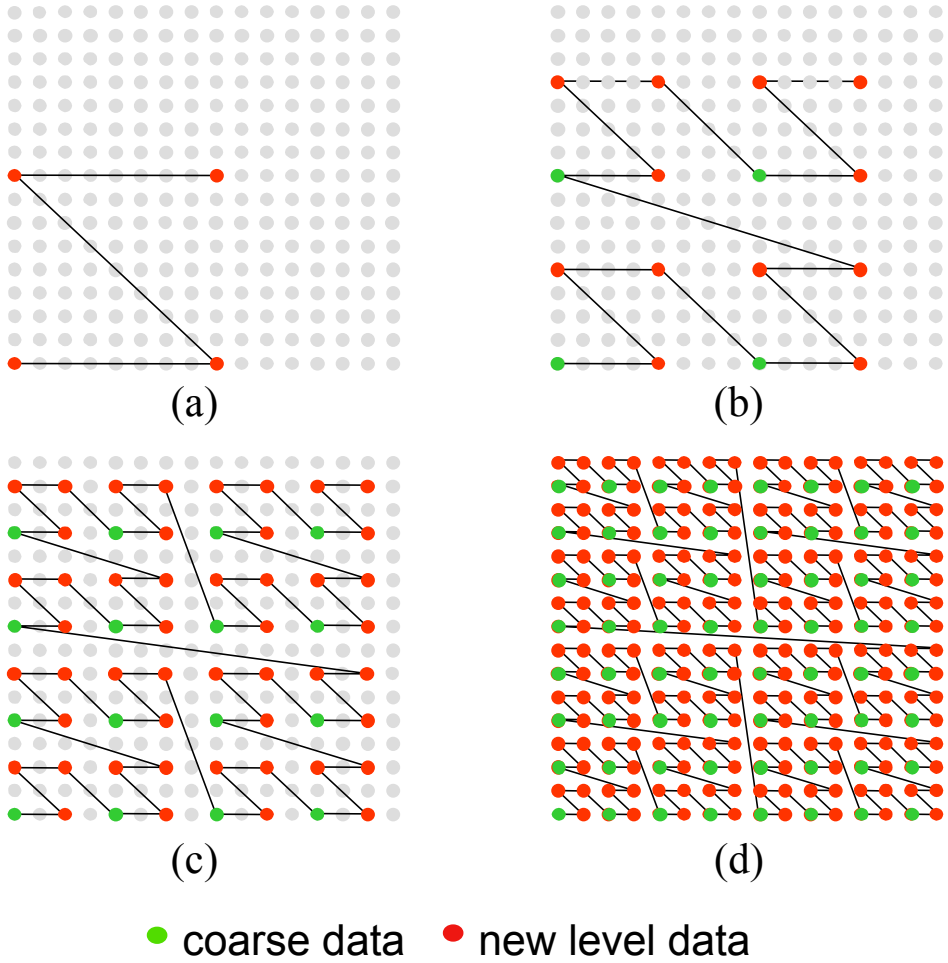
We exploit the correlation of bin/quad/oct-trees with the Lebesgue space-filling curves.

The Lebesgue curve is also known as Z-order, Morton, …. Curve.

Special case of the general definition introduced by Guiseppe Peano in 1890.

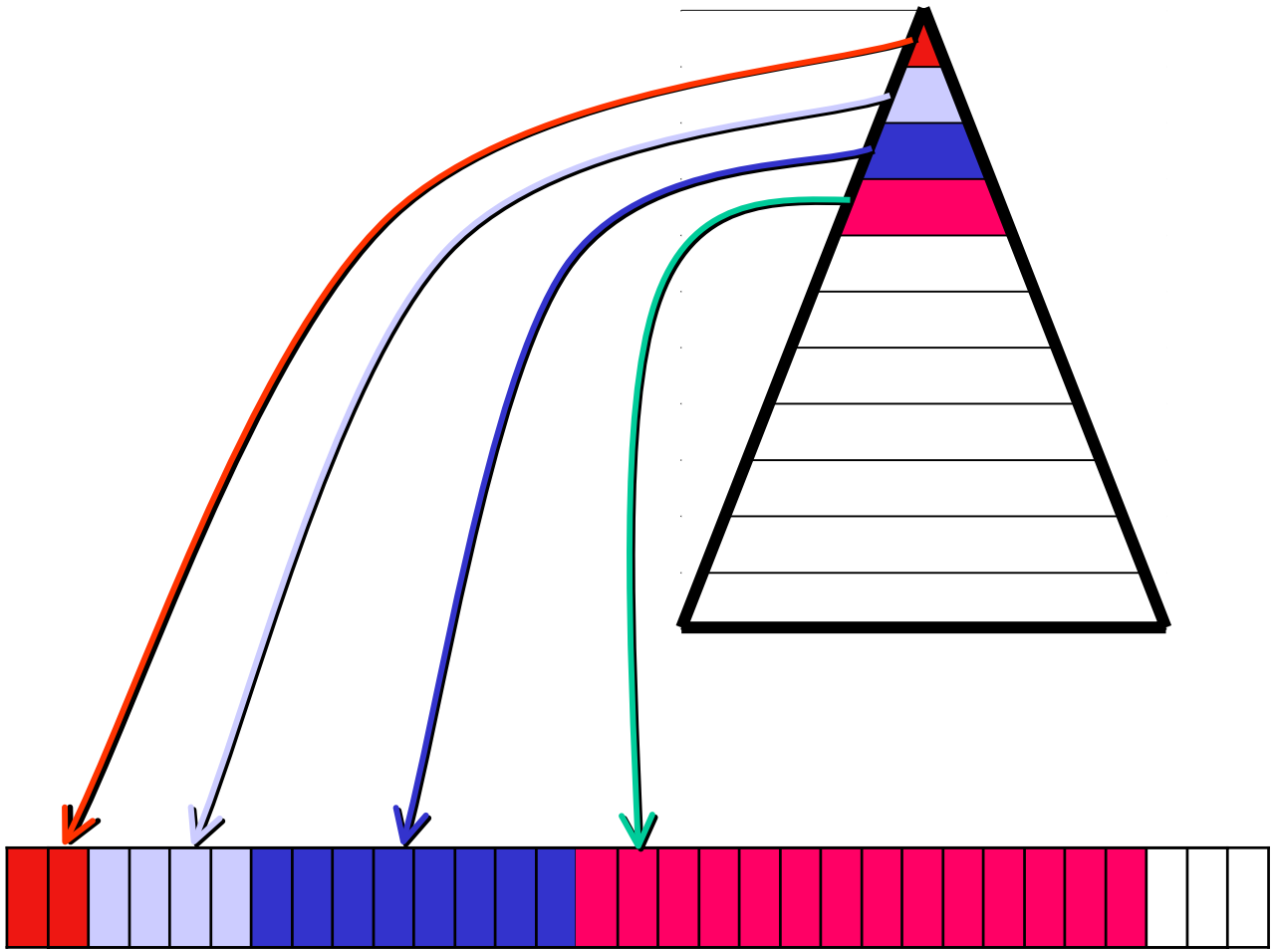


Progressive data streams enable multiresolution visualization.



- On the fly hierarchical Z-ordering
  - Embedded preprocessing
  - Fast computation of hierarchical index
  - Stream can be truncated

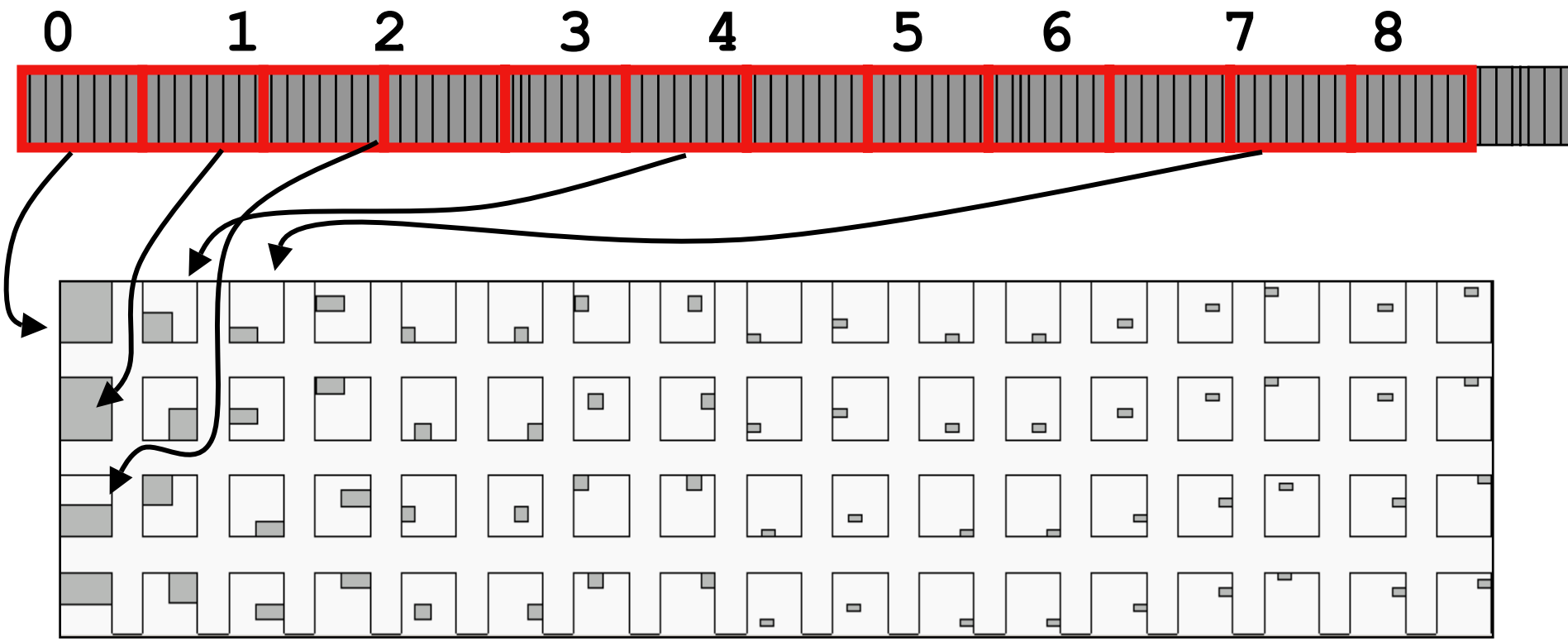
Our cache oblivious data layout optimizes visualization queries.



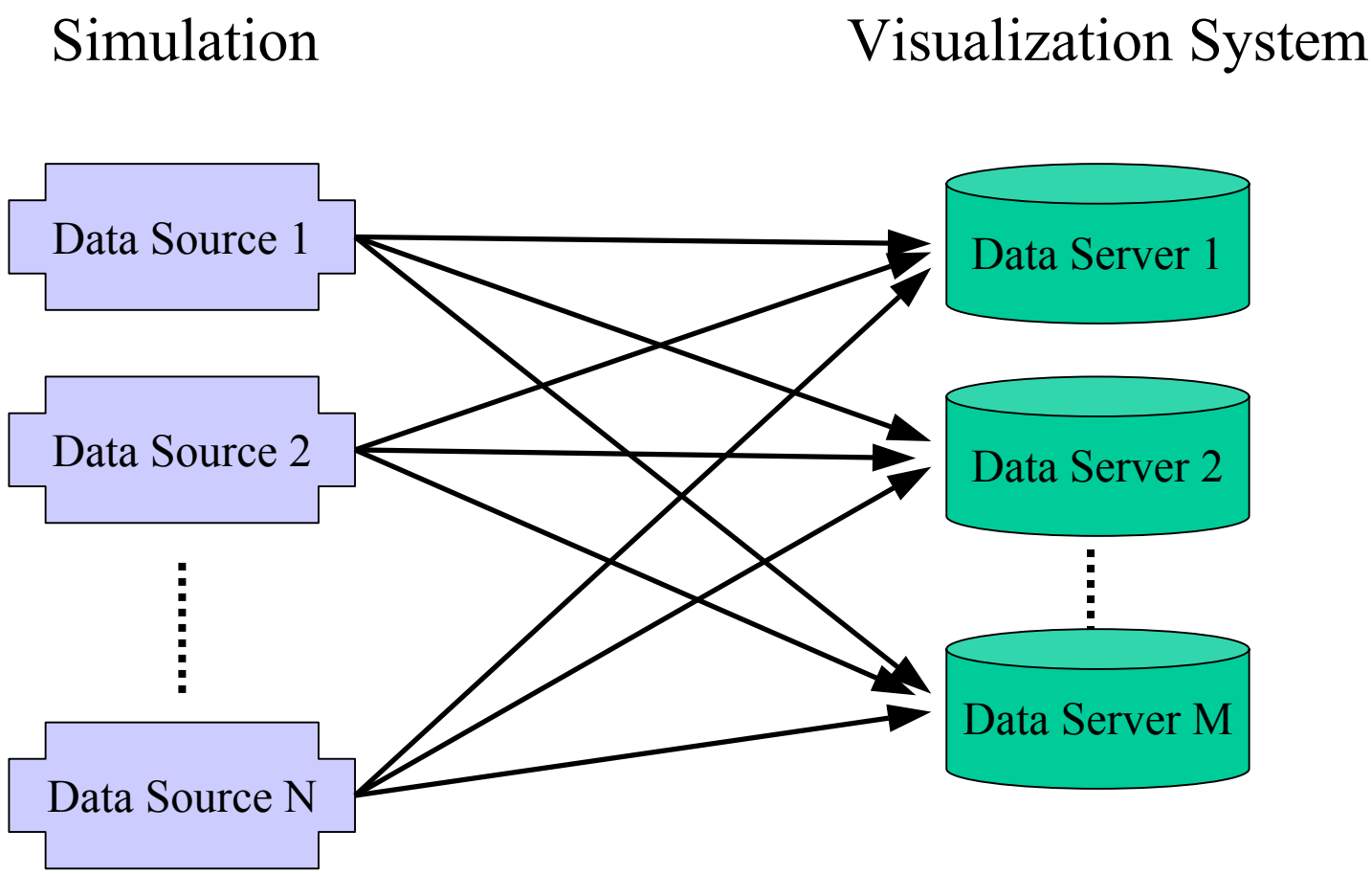
- Data is laid out by level and broken into blocks.
- Simple load balancing on visualization side.
- Fast slicing, isosurfacing, and volume rendering.

The data layout generates increasingly local coverage.

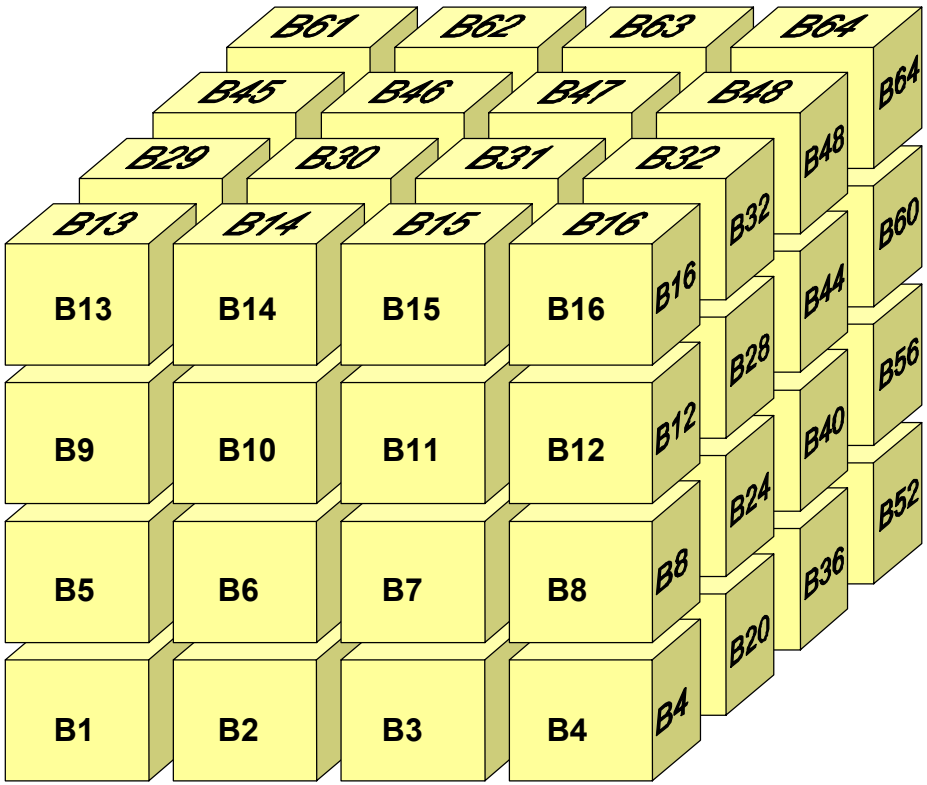
Distribution in the grid of each constant size block of data



We implement immediate streaming of simulation data to the storage location.

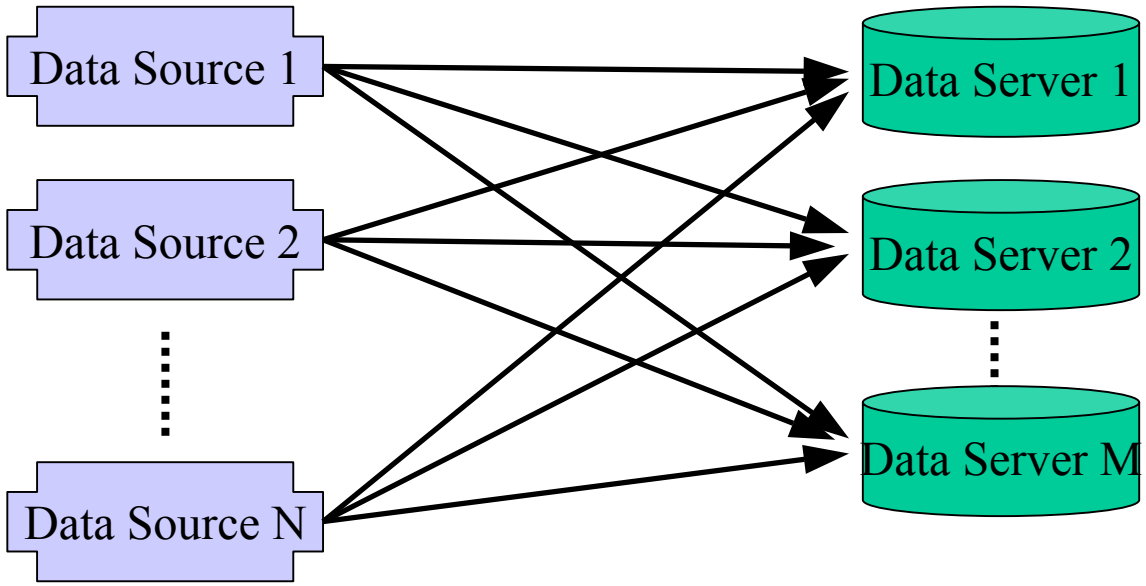


We exploit the data decomposition used in the simulation.



- Takes advantage of simulation load balancing
- One Data Source per compute node
- Each compute node connected to all Data Servers

Data Servers are buffering and caching components.

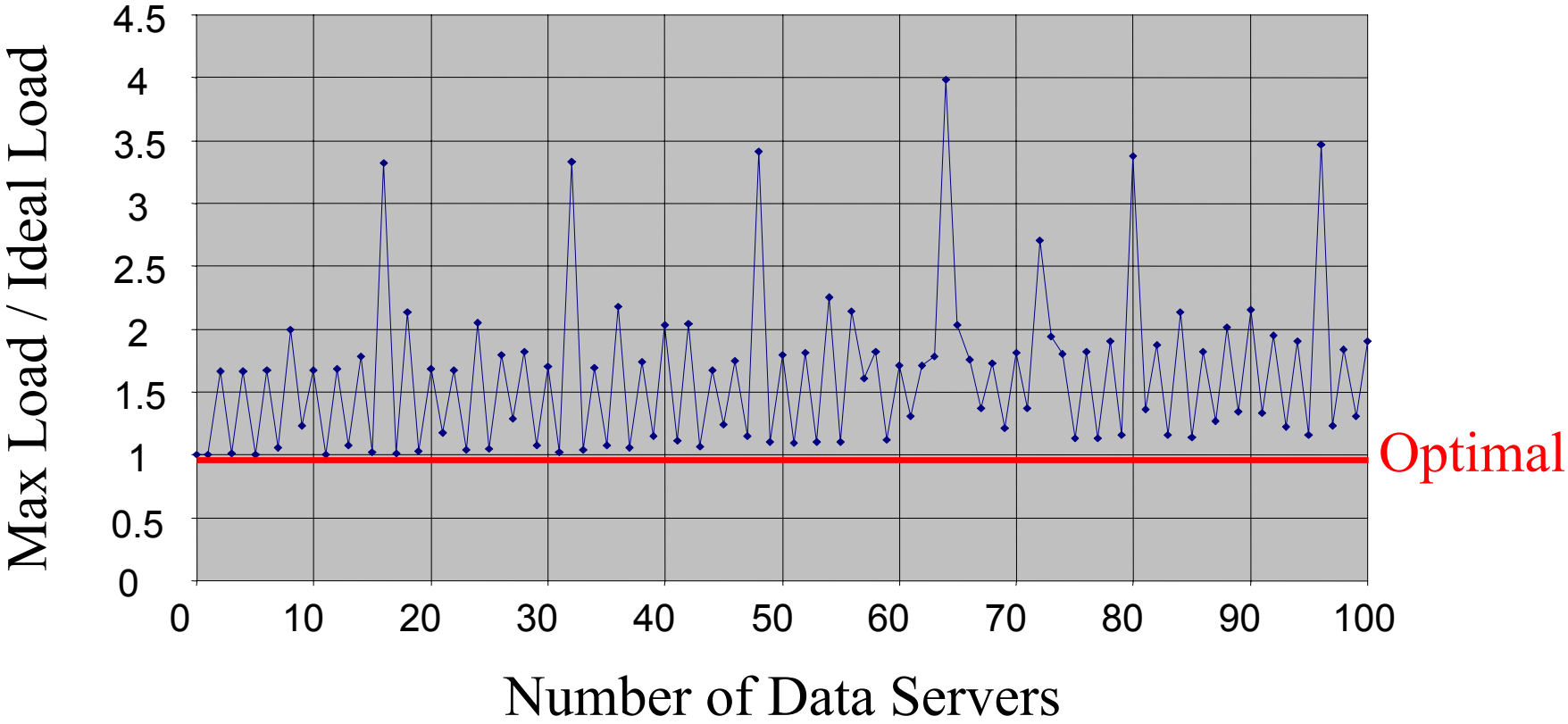


- Cache data and respond to visualization queries
- Filter and buffer data to produce multi-layer configurations

We achieve load balancing with a static data decomposition.

- No contention
- No data duplication
- Streaming:
  - D = I/d mod N
    - D is Data server, I is HZ index of sample, d is chunk size
- Visualization queries:
  - Choose number of Data Servers **wisely** →

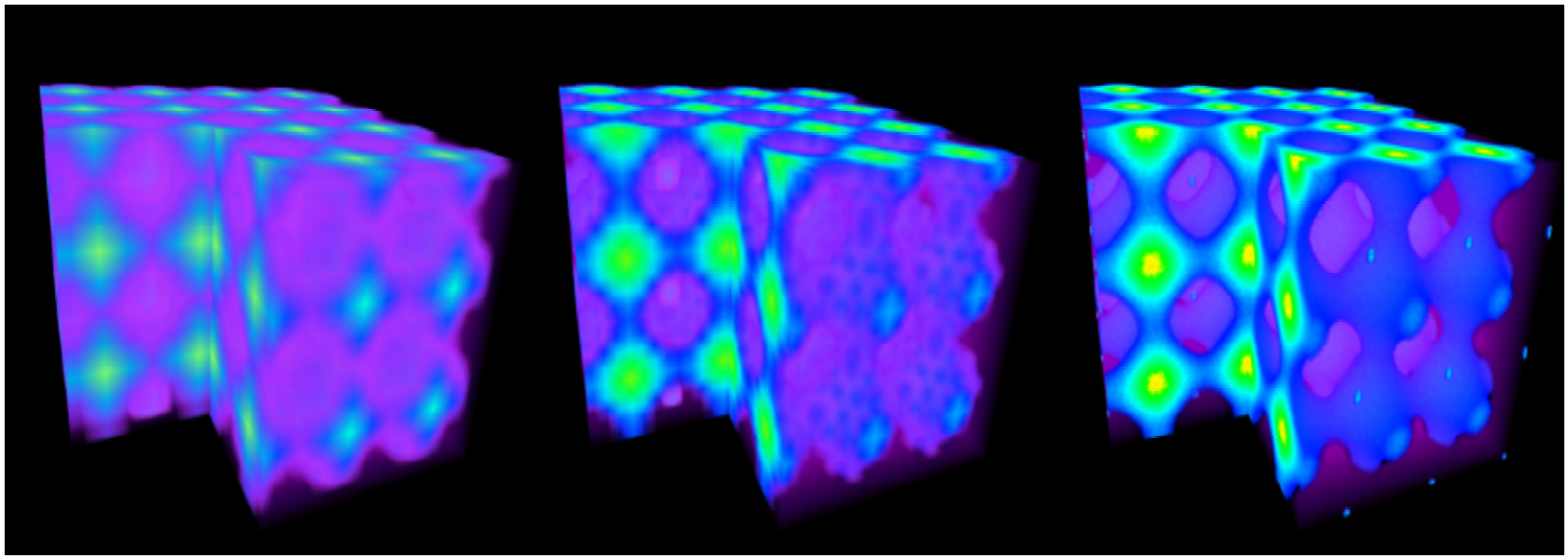
The load balancing of visualization queries depends on the total number of data servers.



The measured streaming time is small compared to the compute time.

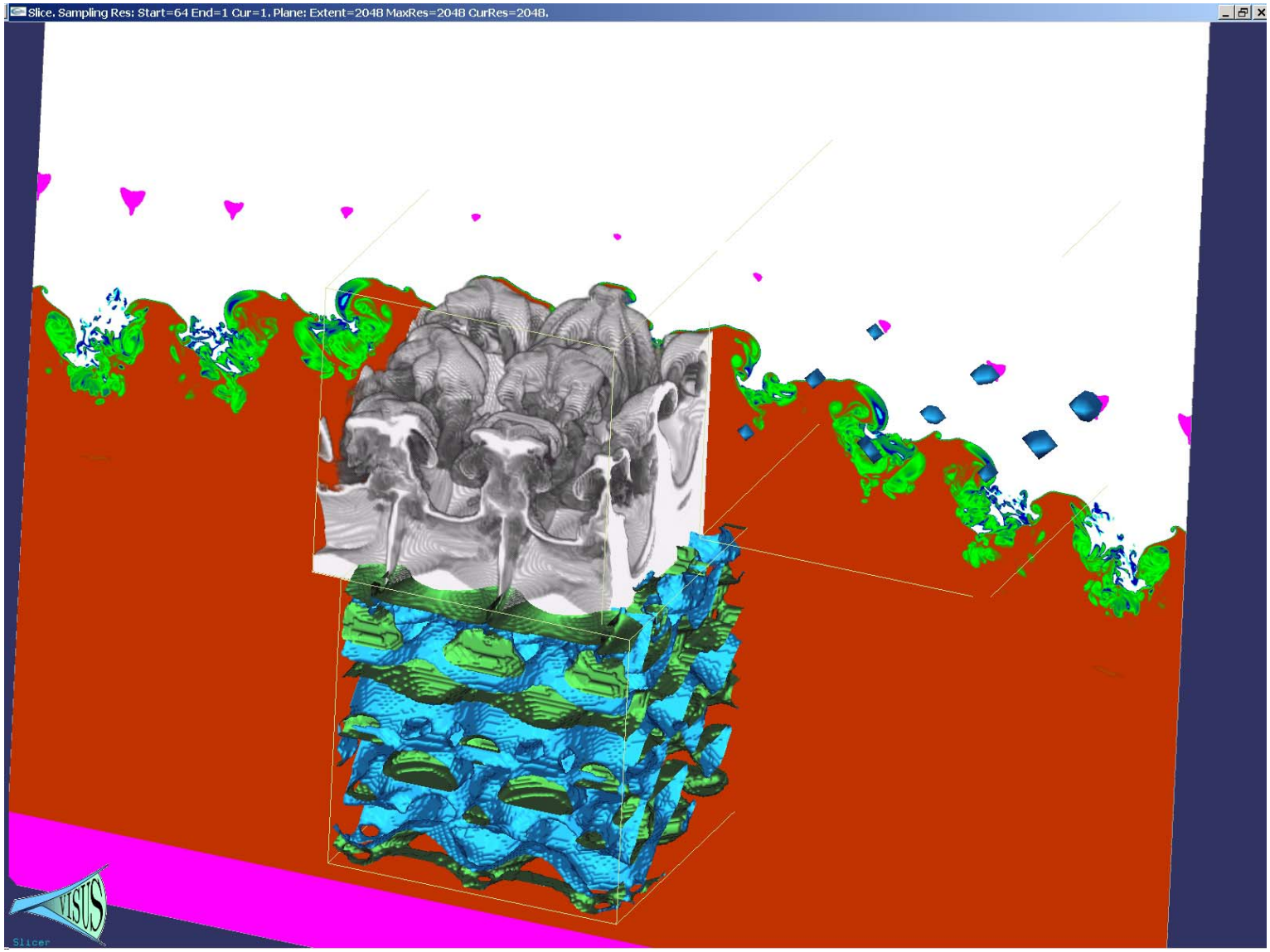
Number of Data Sources	1	8	64
Number of Data Servers	1	3	3
Total Domain Size	128^3	256^3	512^3
Equivalent simulation time/time-step	270s	736s	4224s
Send Time	5.31s	7.03s	38.6s

A progressive data stream enables visualization on desktop workstations.

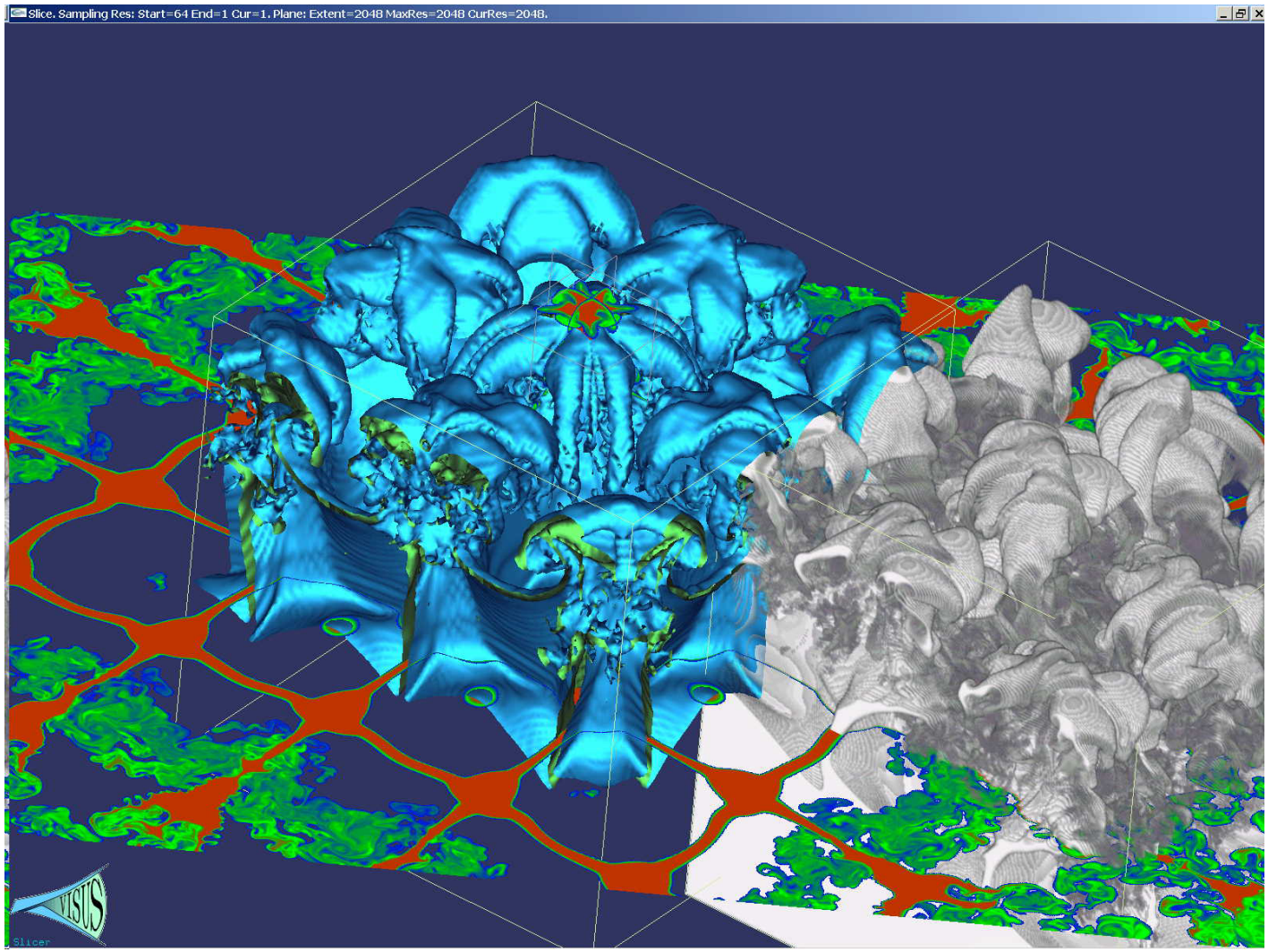


Progressive refinement (left to right) of the volume rendering of the electron density distribution.

Data servers handle localized queries.



On a desktop machine you can explore the finest resolution data available.



~ 8 billion values

Future Work

- Extend to non-power-of-two grid sizes
- Extend to more general mesh structures
- Include fast compression between data source and data server
- Enable multiple layers of data servers to handle larger simulation sizes.