The TeraScale Browser: Multiresolution On-Demand Dataflow

Mark A. Duchaineau

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory

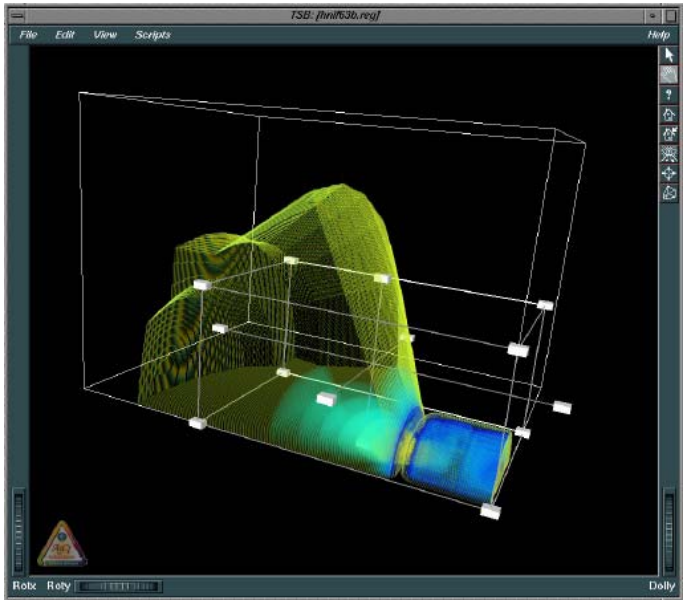
Viz Frameworks Requirements Workshop

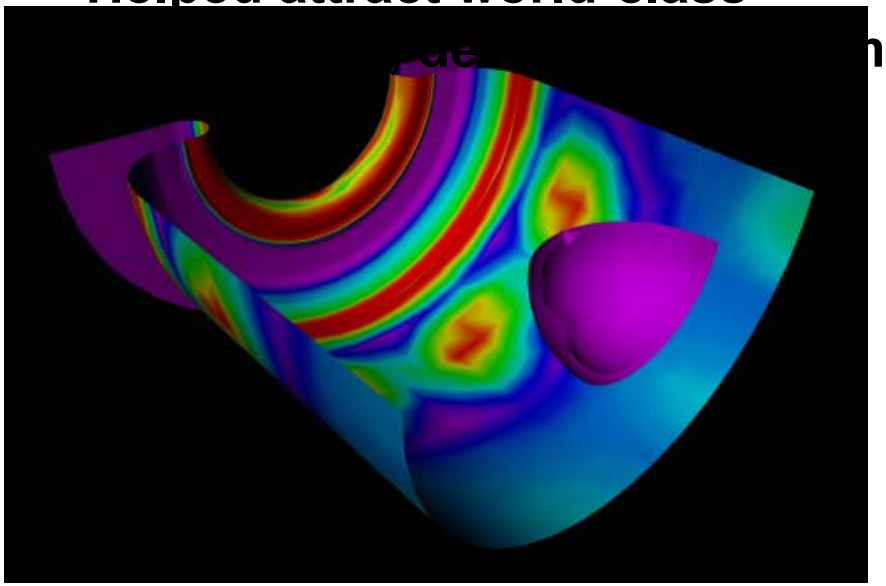
June 2-3, 2003

The TeraScale Browser deploys CASC research to help biggest ASCI runs

History:

- Early prototype "on the side" in ASCI
- Devised basic data flow, compression methods for Big Science runs
- Helped attract world-class

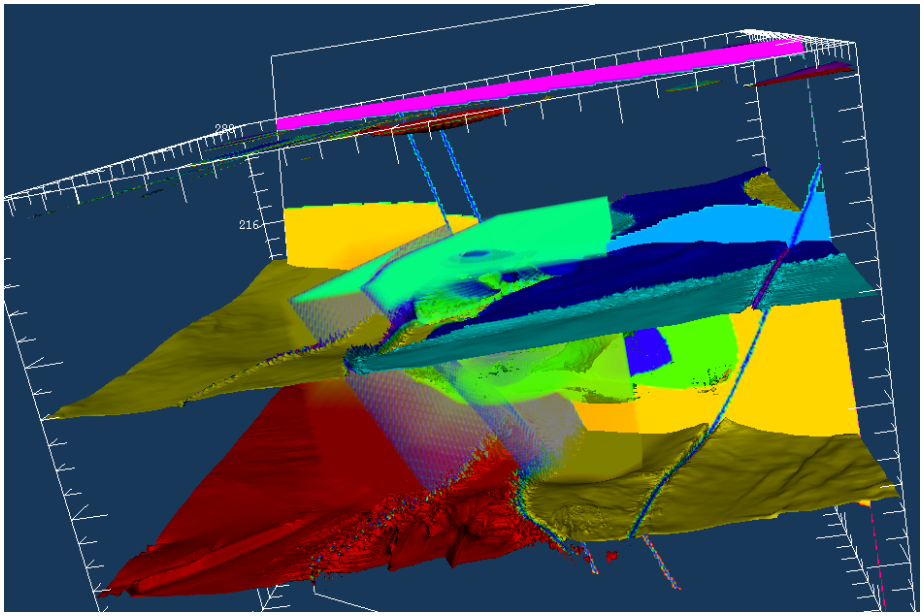




- Uses "Prepare, browse, select"
- Works on largest data sets
- Fully interactive, esp. time slider
- Common visuals
- Parallel and remote display

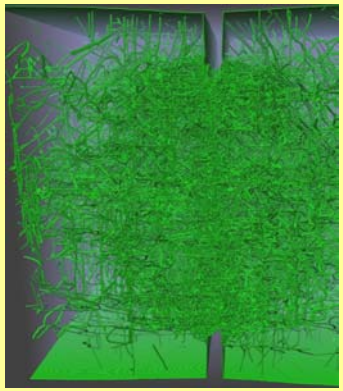
The TeraScale Browser provides quick access to common visuals

- Core visuals
  - Ortho slices
  - Iso-surfaces
  - Volume render
  - Timestep controls
  - Material boundaries
  - Arbitrary cut planes
- Tunable performance
  - Caching of slices/surfaces
  - Fast load/decompress
  - Multi-resolution
- Display anywhere
  - Wall, console, remote
  - Movie generation

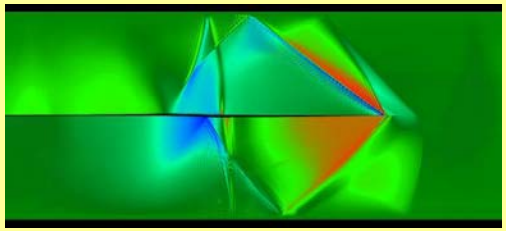


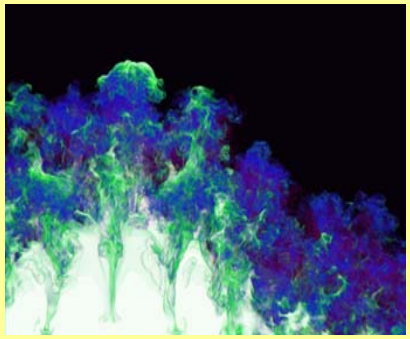
- Scripting
  - Embedded python interpreter
  - Query and set most system properties
  - Outside apps can drive and be driven by the TSB

Examples of “Big Science” datasets



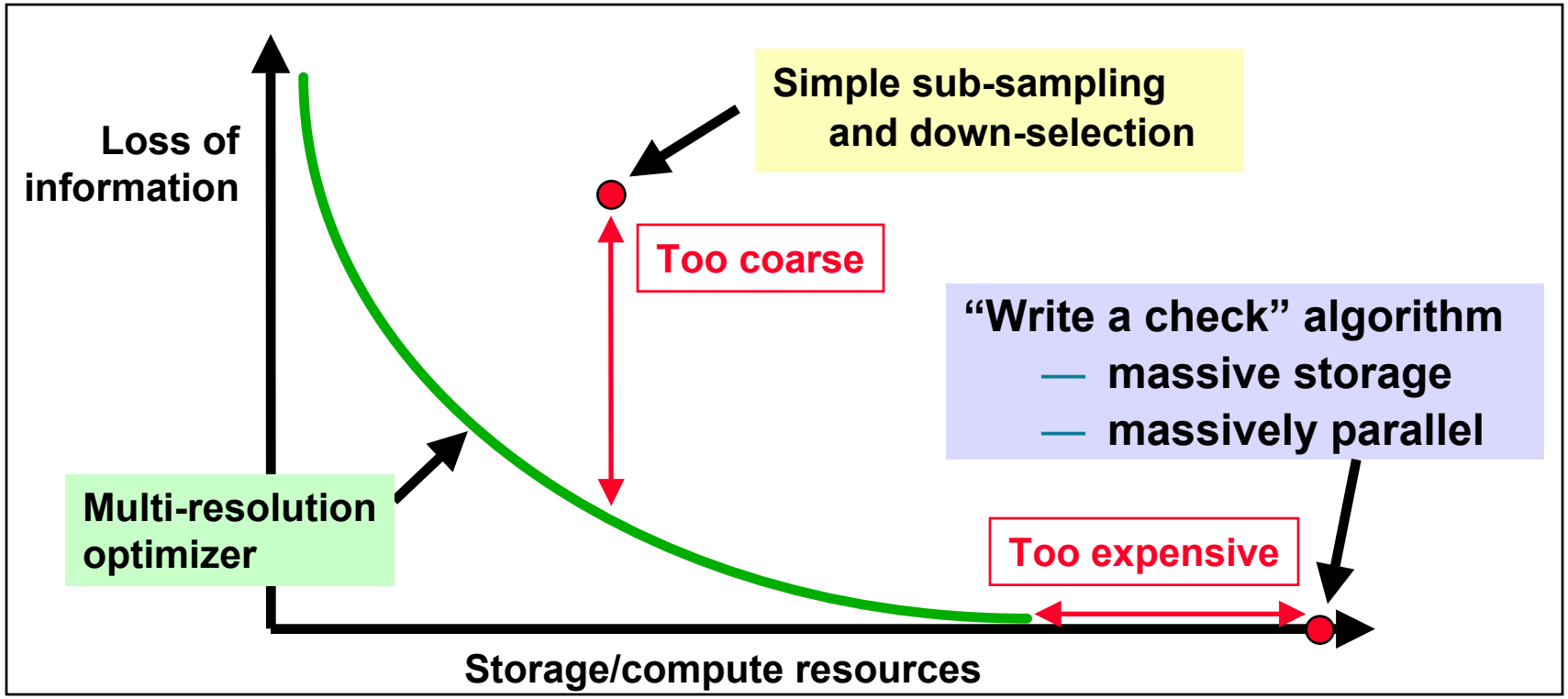
- 5120 ASCI White processors for largest run, Billion-atom 3D MD studies
- 30X compression, from 25TB to under 1TB, 10% co-process overhead



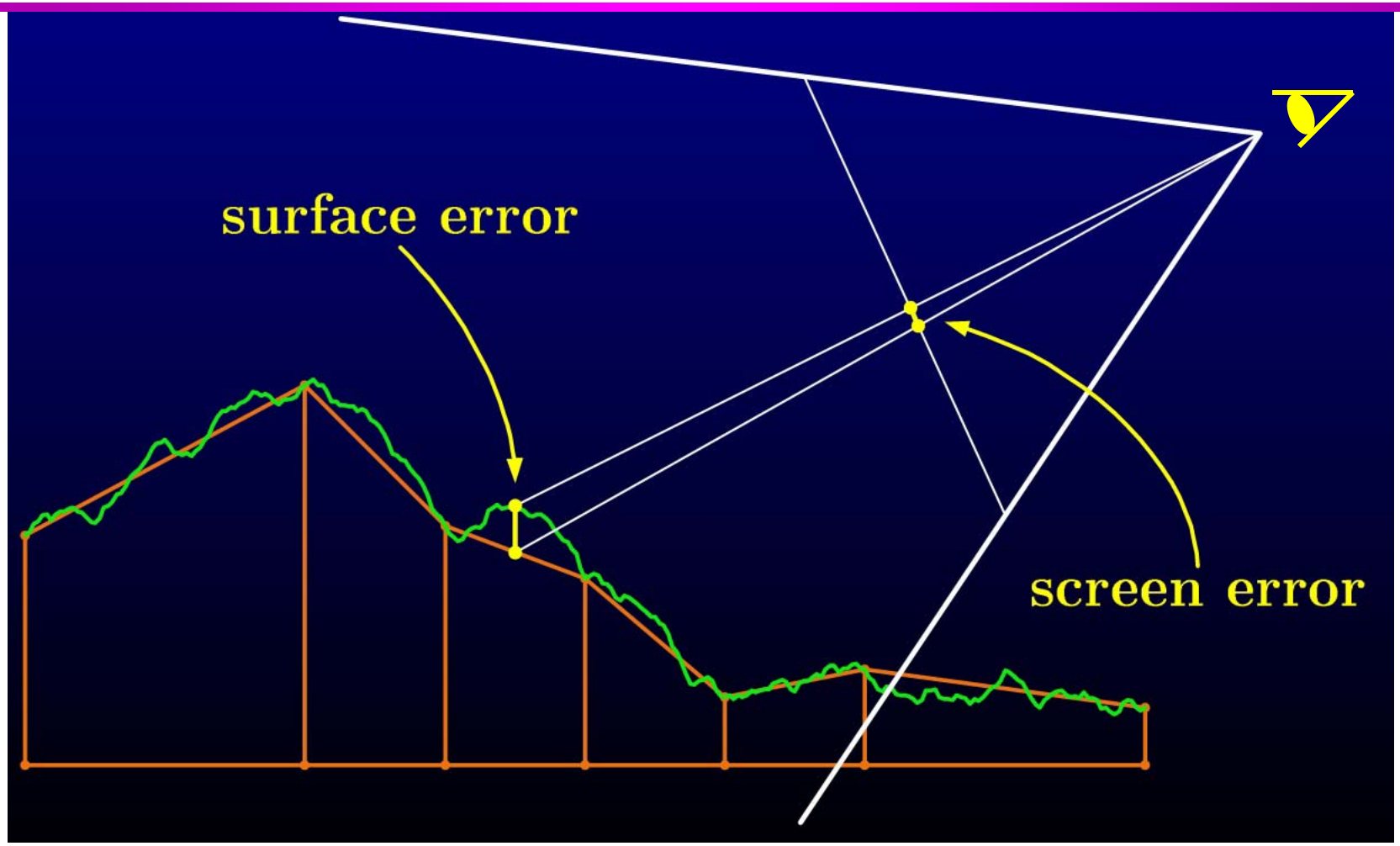


- Gordon Bell Prize, largest Richtmyer-Meshkov run
- sPPM simulation with 24 billion zones ran on 5832 processors of Blue Pacific machine at 1.18TF
- could not store 2.4TB of data without compression
- Wavelet library reduced space 2X & time 10x

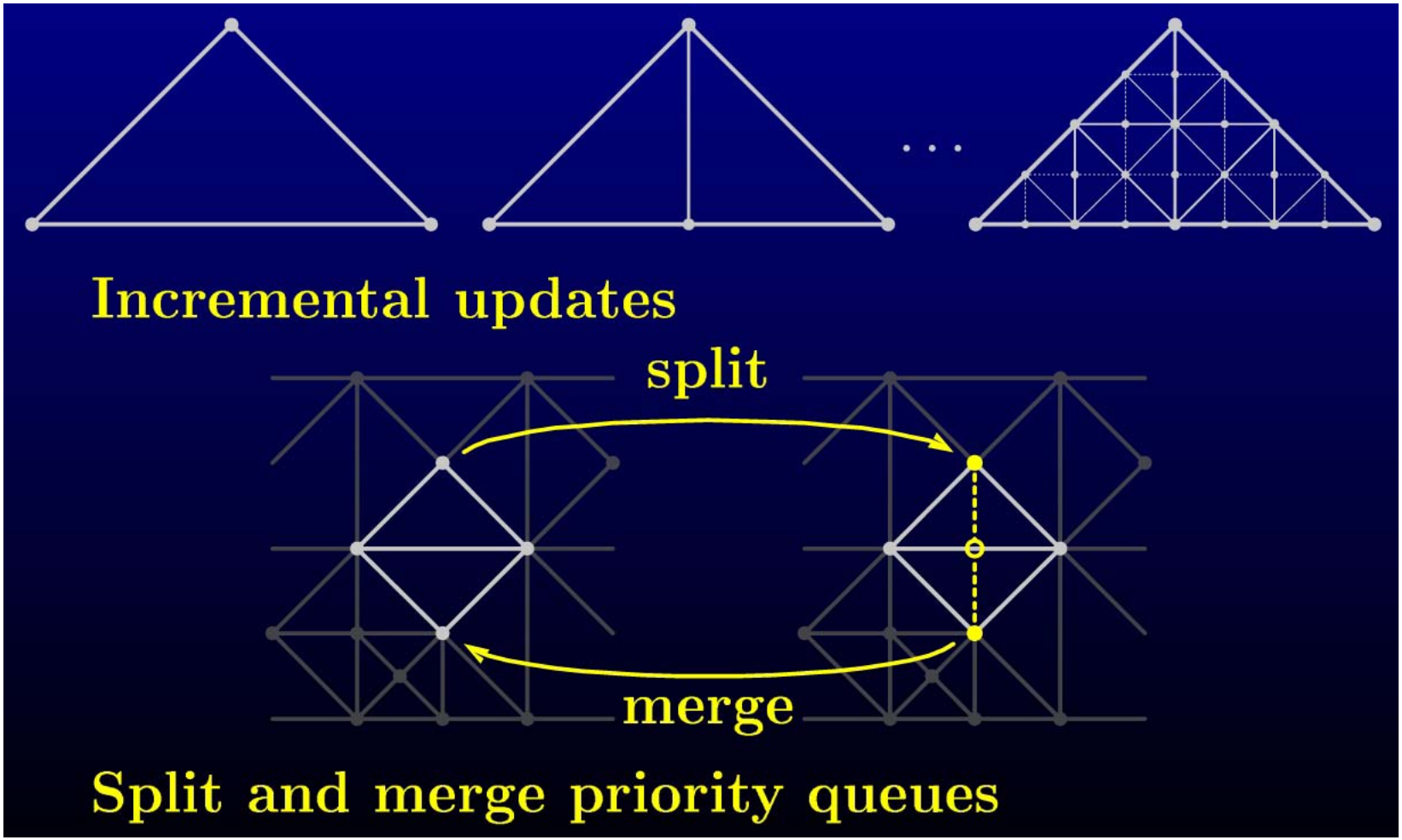
Terascale science requires orders-of-magnitude better visualization data flow



ROAM maximizes the displayed accuracy, which changes with the viewpoint

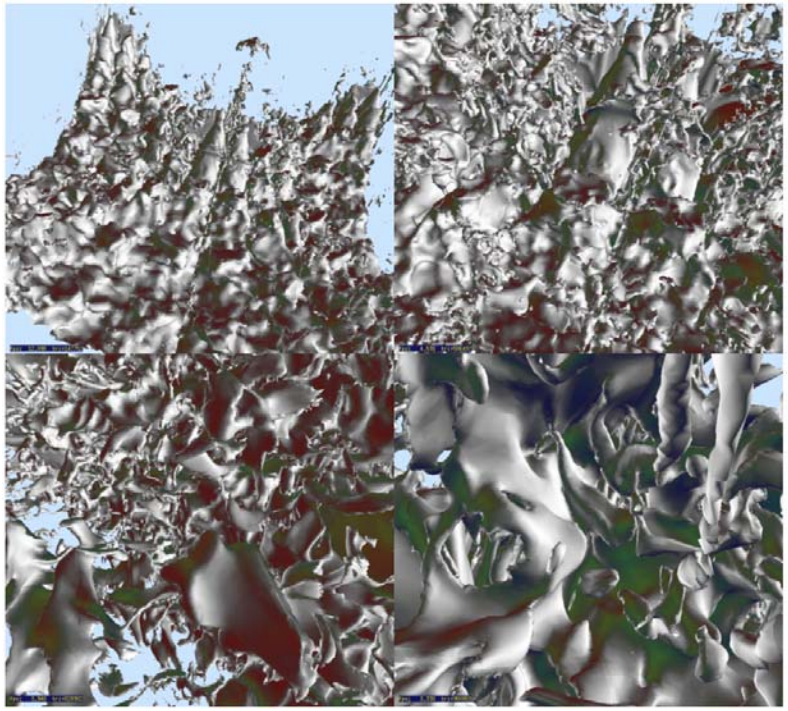


A ROAM mesh adapts accuracy by performing a few splits and merges each frame

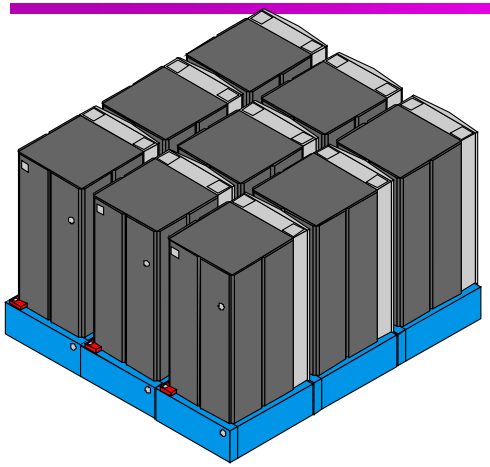


View-dependent 490M Triangle Surface

- Data details
  - Isocontour of a 8B zone dataset
  - 490M triangles in the surface (5.5GB compressed)
  - 1 hour of preprocessing on desktop PC
  - 526MB of final disk space
- Algorithmic features
  - Implicit data storage reduction
  - Out-of-core rendering
  - Demand driven I/O
  - View dependent display
- Host
  - Dell Laptop 1Ghz CPU
  - nVidia GeForce Go graphics
  - Sustained 10fps



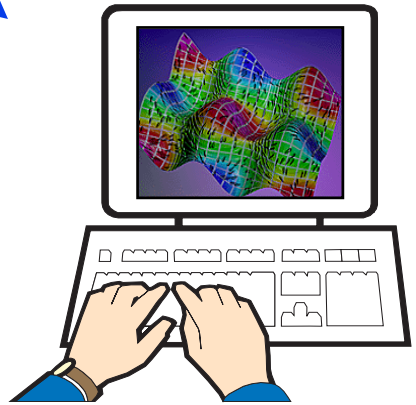
The ROAM optimizer produces a “trickle” of data during interaction with server-side geometry



ROAM geometry server

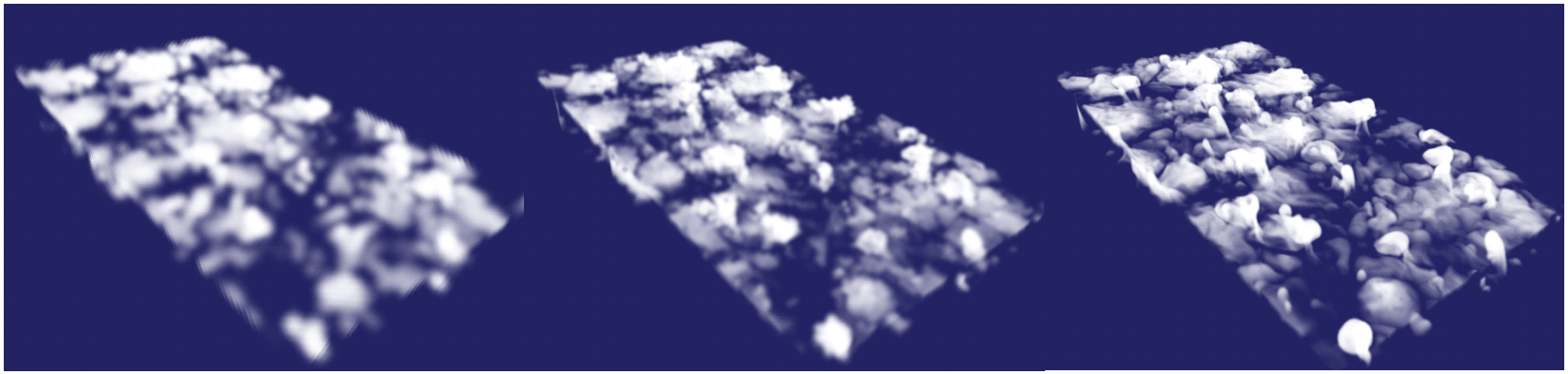
Split/Merge operations (3% of scene)

ROAM-enabled graphics chips

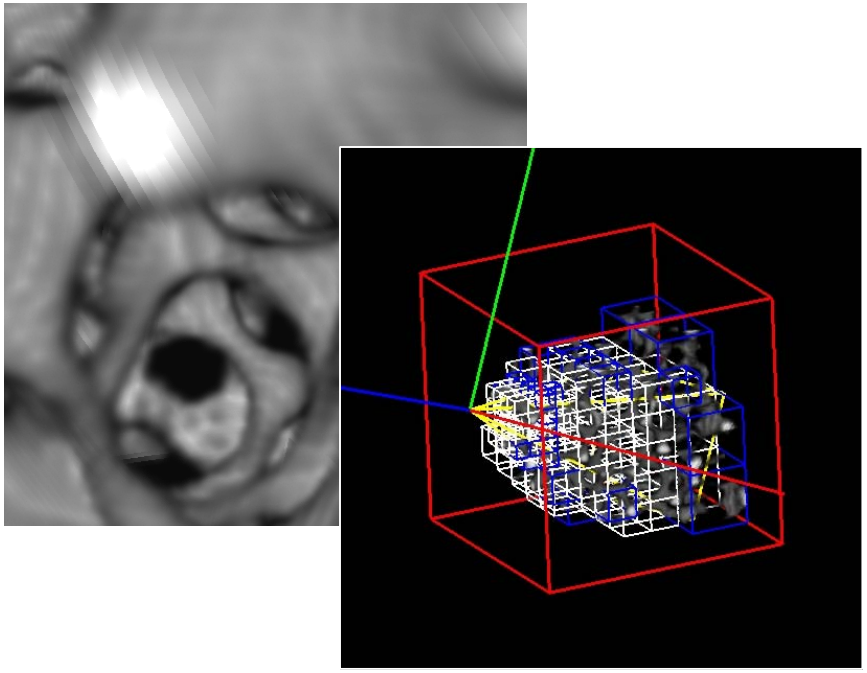


Viewpoint updates

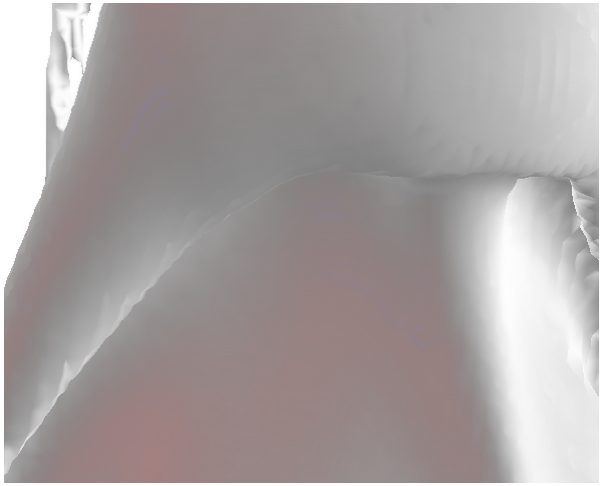
We optimize per view in a resampled-block hardware volume renderer



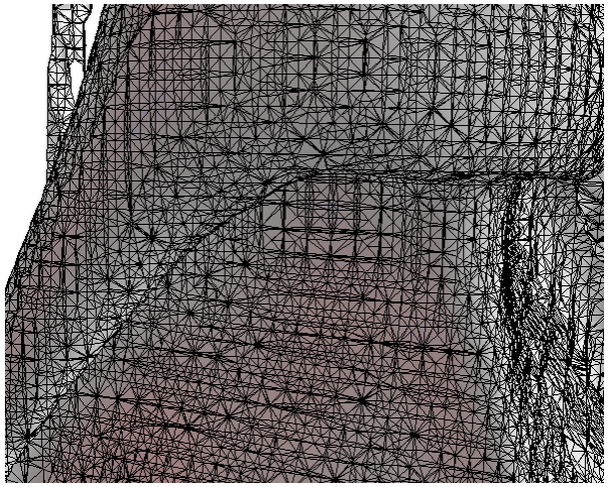
- Hardware-texture based volume visualization
- Octree decomposition of space and resolution
- Adaptive rendering, driven by Viewpoint, Error, Rendering Budget
- Efficient Error Calculation, reduced compositing error
- Cache-friendly paging



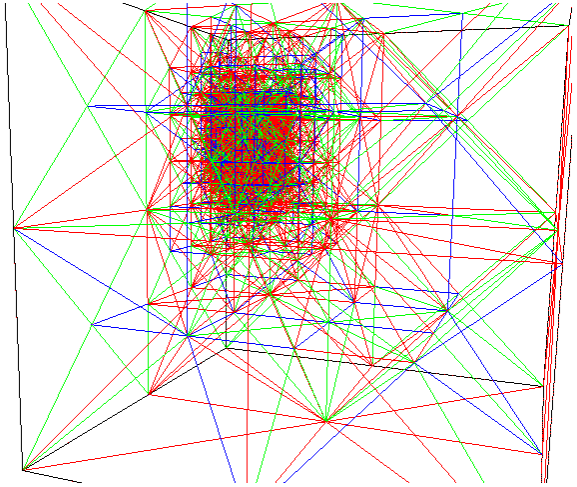
Interactive view-dependent isocontouing used 3D ROAM split-merge

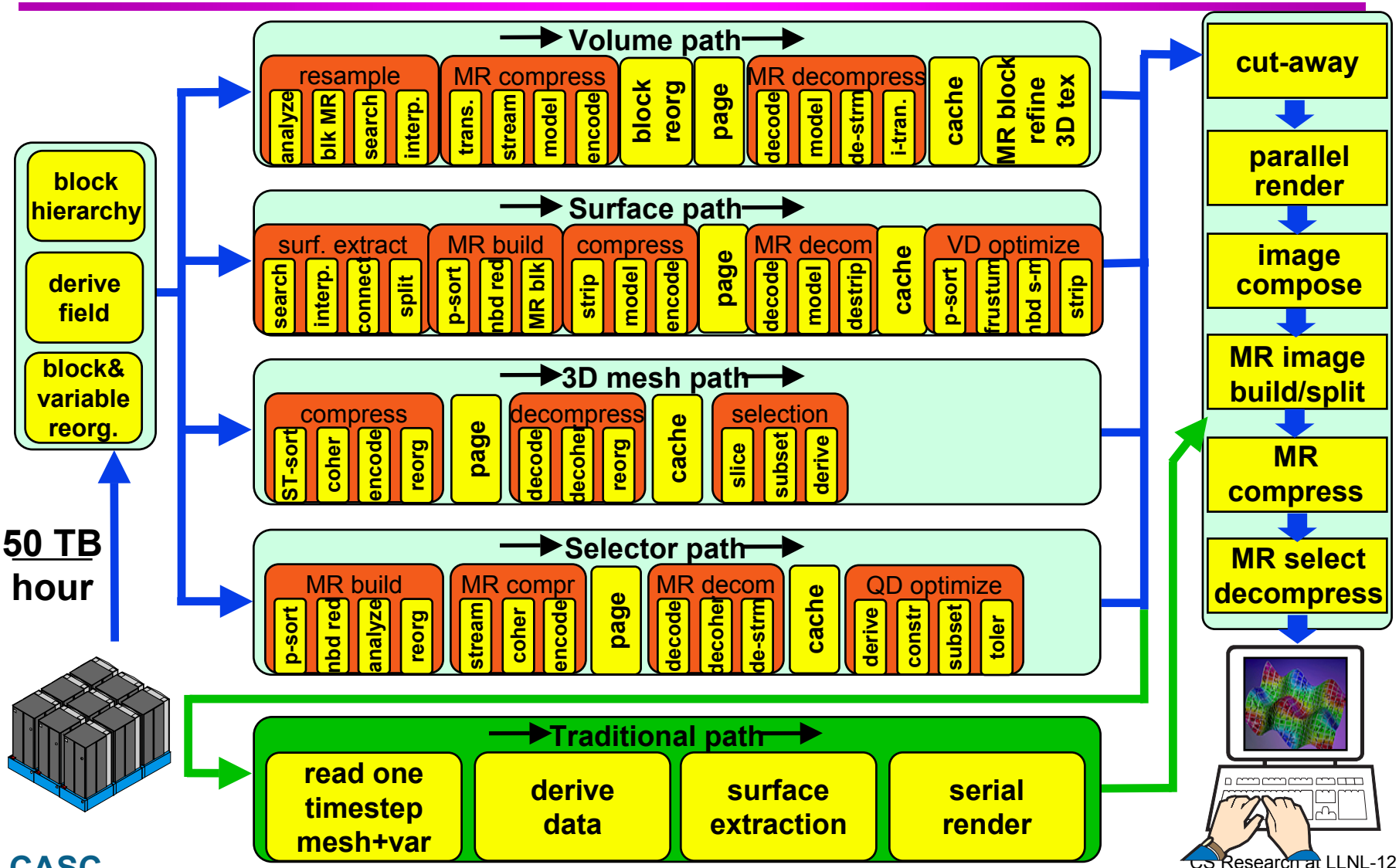


Shaded surface



Triangle mesh



Adaptive mesh

The software dataflow for big-data interaction