

# A Scalable Viz Cache Architecture

James (“Jeeembo” ☺) Kohl

Computer Science and Mathematics Division

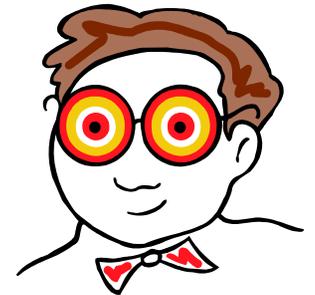
Oak Ridge National Laboratory

Tuesday, April 29, 2003

Research supported by the Mathematics, Information and Computational Sciences Office, Office of Advanced Scientific Computing Research, U.S. Department of Energy, under contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

# We're Increasingly Doomed...

- Scientific Datasets Ever More Massive:
  - Climate/T21 ~ 2 Terabytes per run...
  - TSI/Supernovae ~ up to 50 Terabytes per run...
  - Actual Petabyte dataset(s) exist? (classified 😊)
    - Genomes to Life, others...
- No “Good” Way to Fully Explore.
  - Best hi-res displays ~ 100 Mpixel?
    - Limited by human visual acuity...
  - 10,000 to 10,000,000 Bytes Per Pixel! (ouch)
  - Days/Weeks to analyze & render (one frame!)



# Current Practice



- Filter, Reduce, Data Mine, PCA...
  - Statistically-based approaches to shrink the dataset size “without losing key features”
- Pre-compute/Pre-render as much as possible
  - Terascale Browser (Randy/LLNL)
  - New Parallel Algorithms
    - E.g. Image-based rendering/occlusion culling... (OSU)
- Many isolated & distinct solutions...
  - Limited Interactivity, Interoperability...
  - Typically a single or few view paradigms...

# Need a Scalable Framework!

- Common development platform
  - Need to “standardize” component interfaces...
- Integrate wide spectrum of technologies:
  - Data Analysis & Reduction
  - Geometry Generation & Rendering
  - Image Compositing & Display
  - Interaction Paradigms & Data Navigation
- Combine Best Expertise & Best Resources
  - Utilize high-performance cluster technology

Future

Viz

Whiz...

# Proposed “Scalable Viz Cache”

- Multi-Stage Analysis & Viz Pipeline
  - Integrate Full Parallel Data Pathway
    - From Archive/Live Appl all the way to Display...
    - Modular, Pluggable Functionality (Components)
- *Suites* of Computational Clusters
  - Not “just” local memory hierarchy...
    - Distributed Memory and Parallel Disk Cache
- Multi-Resolution Interactive View Navigation
  - Heterogeneously, across intermediate forms
    - Zoom In & Out through Pipeline Stages
      - Levels of Detail ~ *Overview* ↔ *Actual Data Elements!*

# Viz Cache Architecture



Not all combinations  
make sense...

Zoom Source Selection...

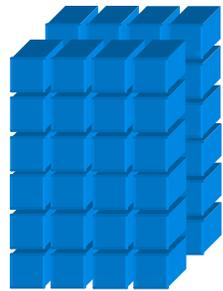
**Mesh  
Transformation?**

Heterogeneous Intermediate  
Data Formats!

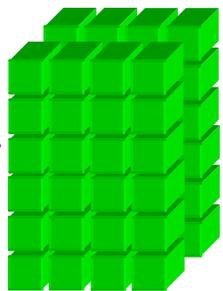
**View  
Continuity?**

Direct Viz of Stat By-Products...

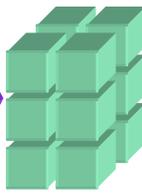
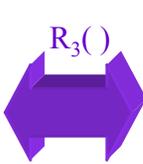
(Spatial  
Correlation)



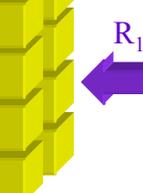
Petabyte or  
Terabyte  
Appl Data  
  
(Archive  
or Live  
Simulation...)



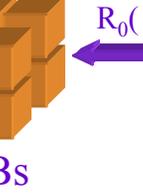
Reorganized  
Data For Viz  
  
(Spatial or  
Temporal  
Striping?)



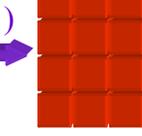
100s of GBs  
  
(Dim  
Reduct?)



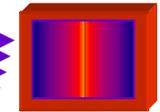
10s of GBs  
  
(Sampling?)



GBs  
  
(PCA?  
FFT?)



100s of MBs  
  
Parallel  
Rendering  
  
(ParaView,  
TSB,  
Ensignt,  
Custom?)



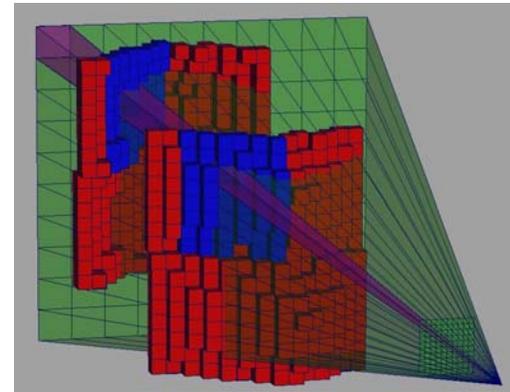
Display  
  
Image  
Comp.



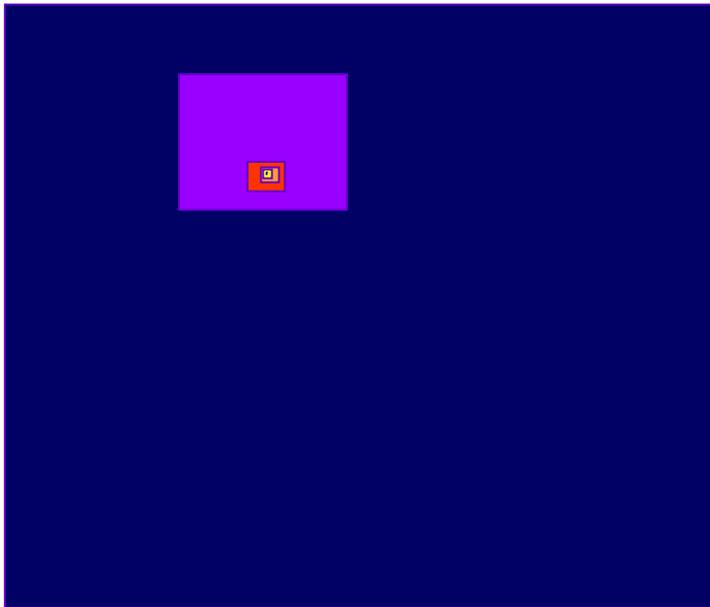
Reduction & Filtering

**Processing and  
Bandwidth Requirements?!**

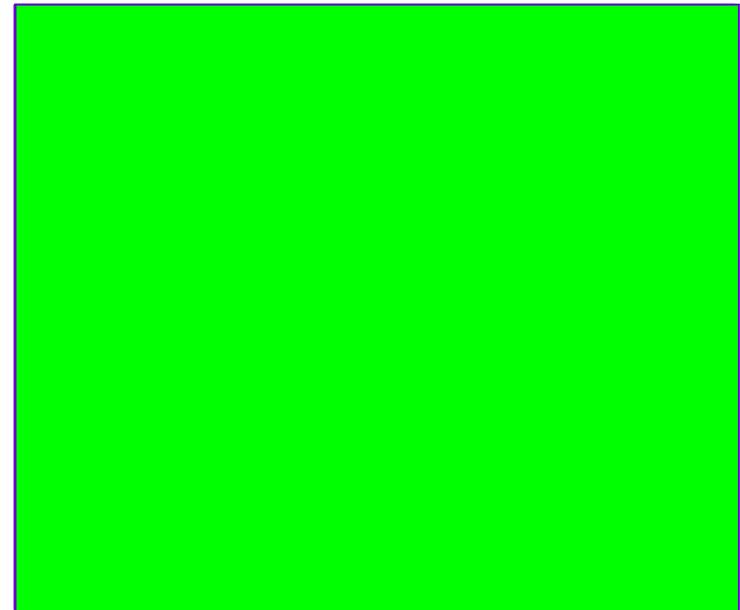
# Multi-Resolution Navigation



- “Initial” View ~ High-Level Overview
  - Zoom & Navigate through Hierarchy of Details



Global Context



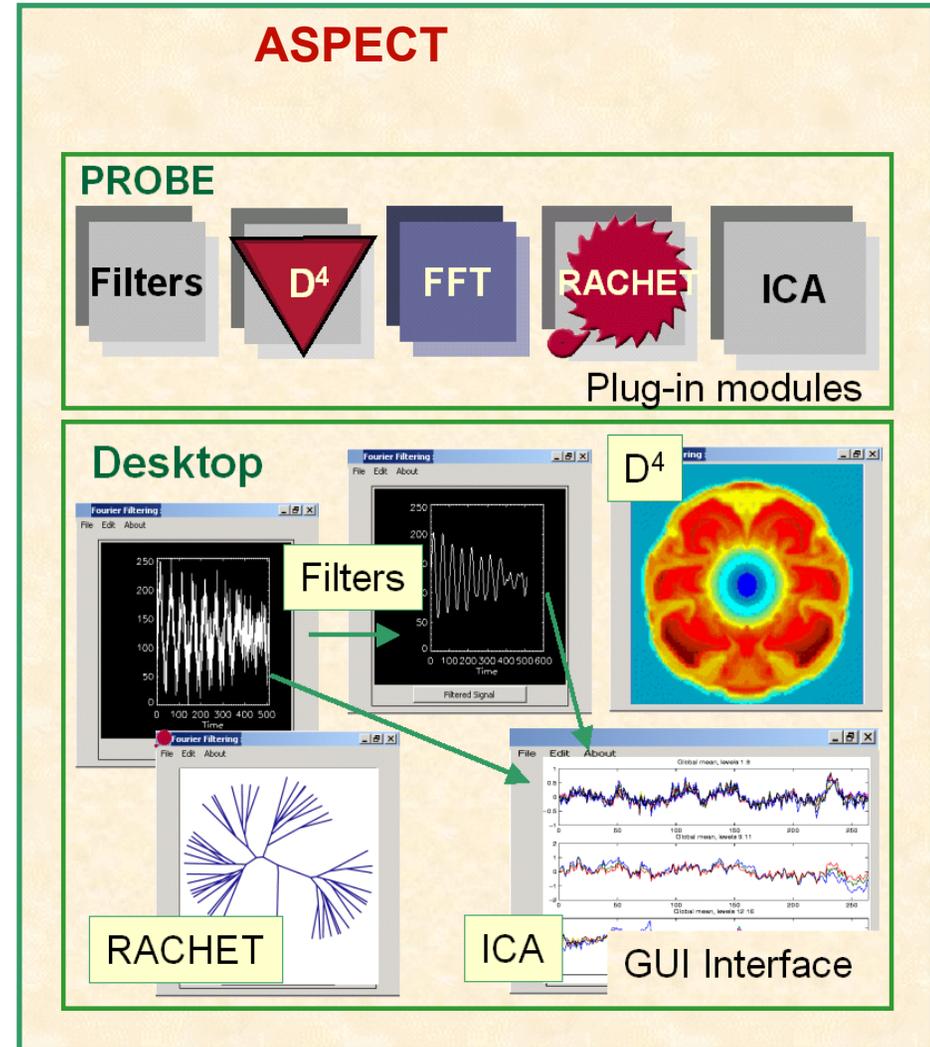
Navigation / Display

# Data Analysis Paradigms

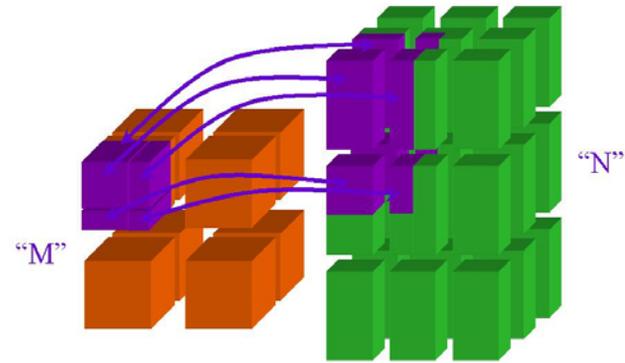
## ASPECT: Adaptable Simulation Product Exploration and Control Toolkit

N. Samatova, G. Ostrouchov, I. Watkins

- Already Component-Based
    - XML Module Interfaces
  - Existing GUI, Infrastructure
    - ParaView hooks...
  - Variety of Filters/Reductions
    - RACHET Clustering
    - Time-Series Subsetting
  - Data Wrangling Expertise
    - MPI-IO: HDF, NetCDF...
- (Not Scalable, Needs Caching 😊)



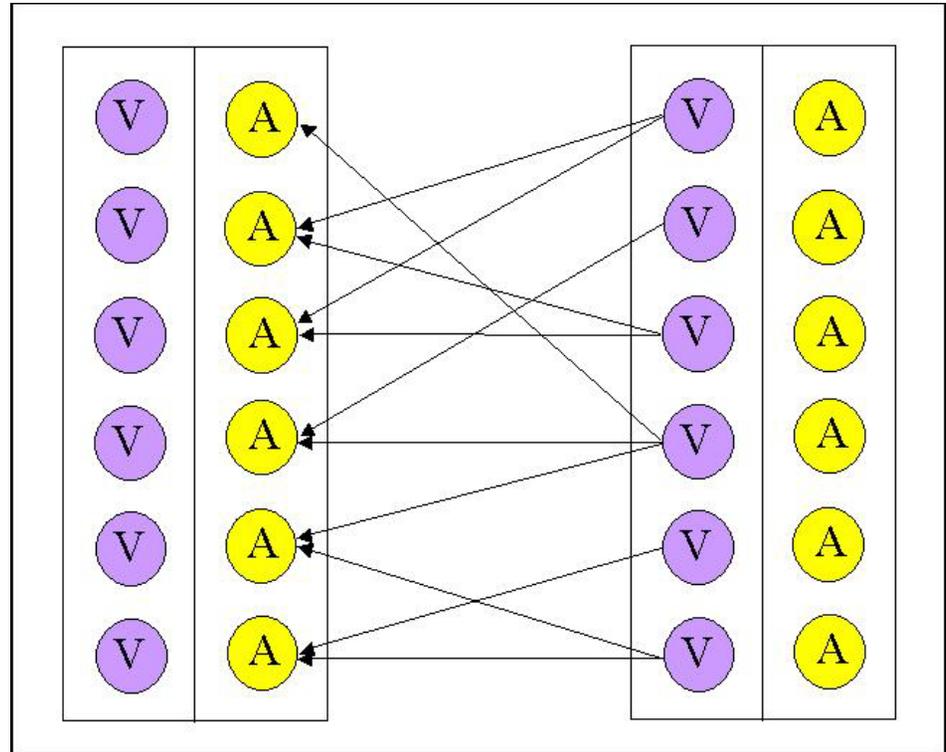
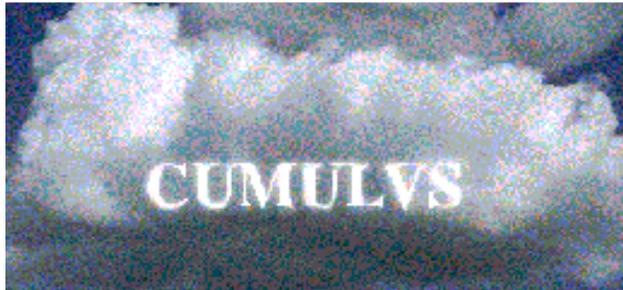
# Parallel Data Movement



- Common Component Architecture (CCA)
  - “MxN” Parallel Data Redistribution Effort
    - General Parallel Data Exchange (not just decomp)
    - Various Synchronization Options
      - One-Shot vs. Periodic (streaming?)
    - Bridges Gap Between Distinct Frameworks
      - Matches well with pipelined stages of Viz Cache...
  - Build on existing (culminating 😊) infrastructure
    - MxN uses ORNL’s CUMULVS system
      - Interactive Viz, Steering, F.T.

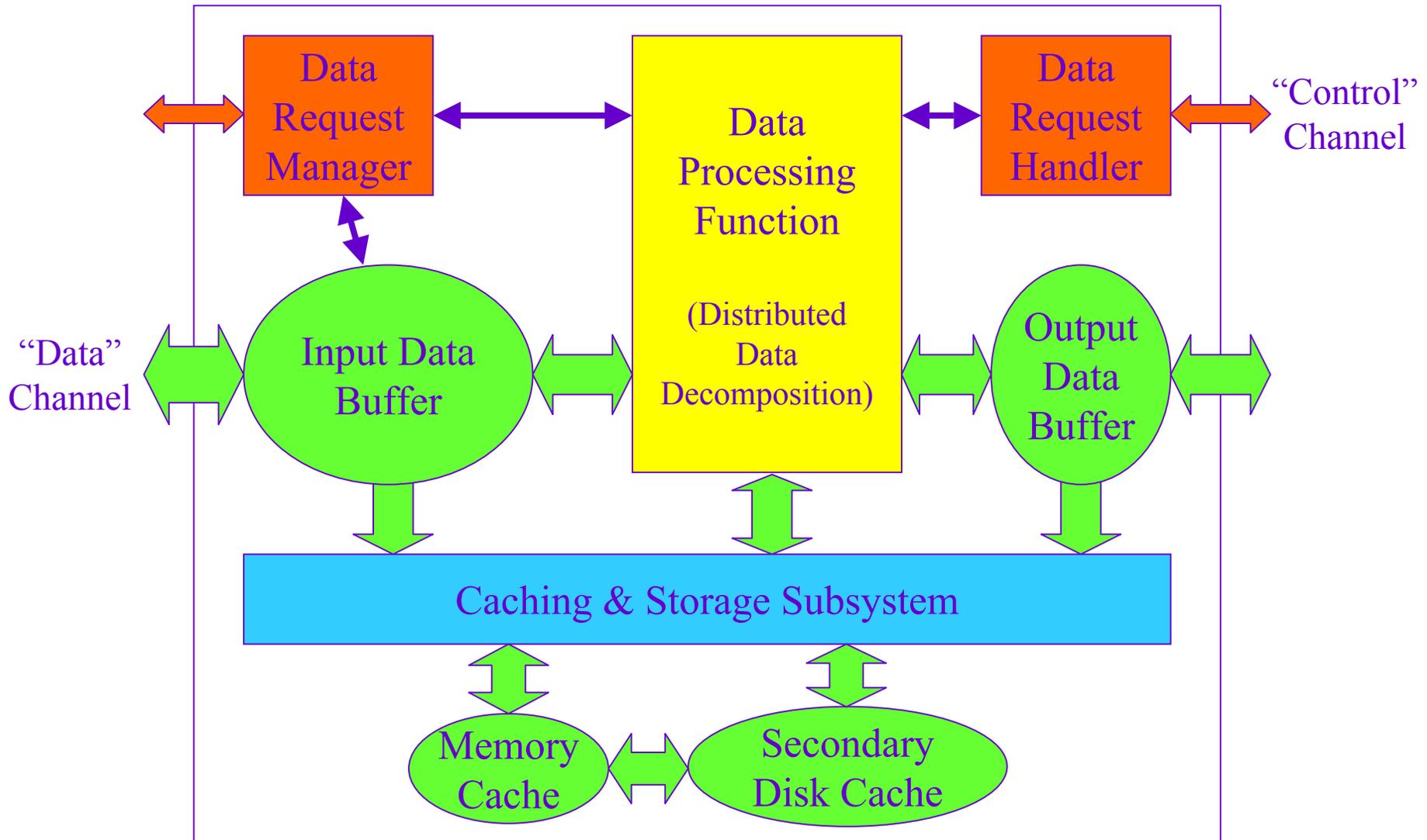
CCA “MxN” → “M x N x P x Q x R”!

# Using CUMULVS Technology



- Each Layer is:
  - Application to Front-End Layers...
  - Viewer to Back-End Layers...
- Need “Callbacks” vs. Static Data Allocation

# Pipeline Stage Component Organization



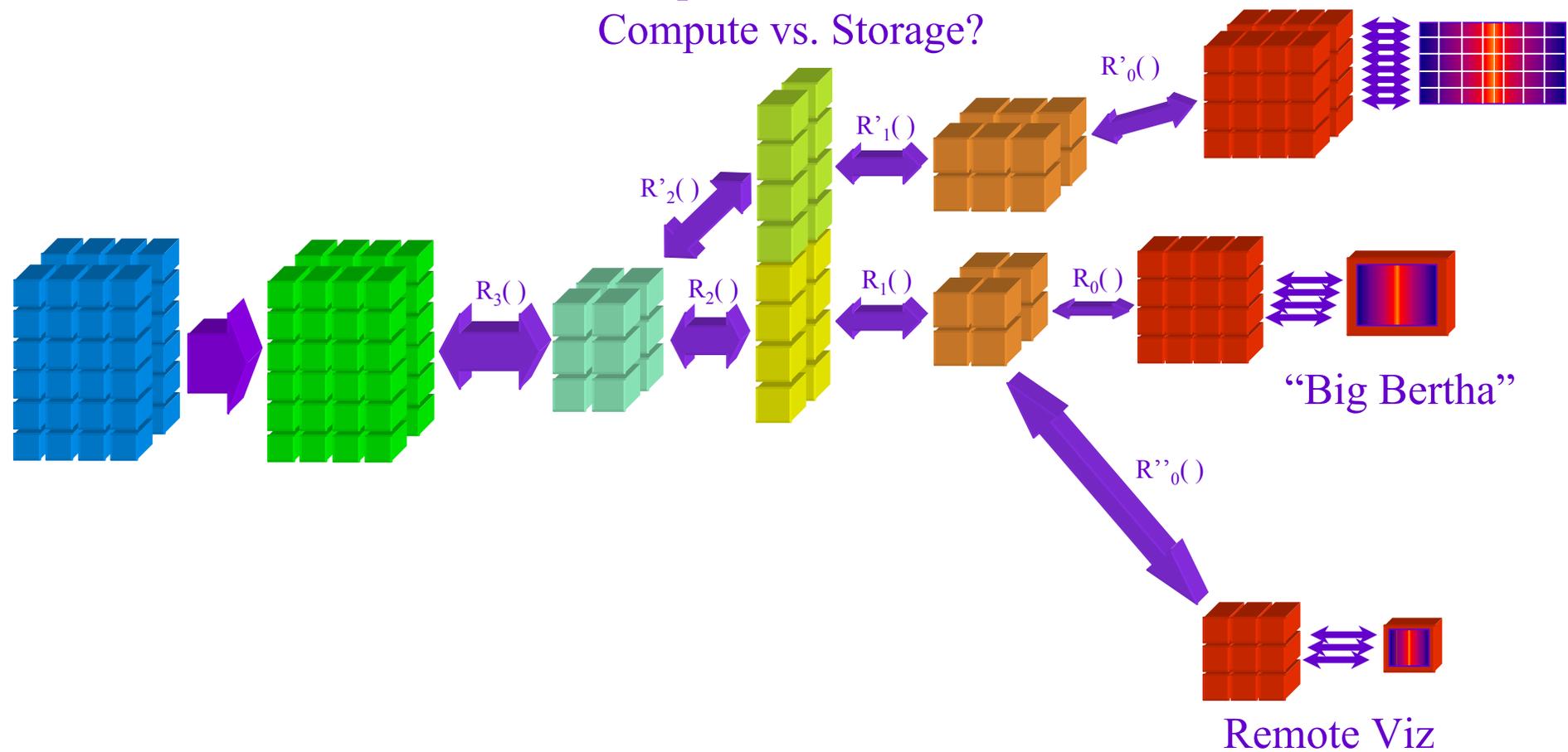
# Viz Cache Research Issues...

- Front-End Navigation & Control
  - Interface/GUI that Preserves Global Context
- Caching Schemes!
  - Critical to Interactive Performance
    - Need for Simulator, Computational Complexities
- Parallelizing Data Analysis Functions...
  - Currently Mostly Serial
- Parallel Rendering
  - Integration of New and Existing Solutions

# Flexible Viz Cache Configurations

Split Resources:  
Compute vs. Storage?

PowerWall





# Viz Cache Summary

- Modular Architecture for Scalable Data Analysis and Visualization
- Interactive, Multi-Resolution Zooming and Navigation through Intermediate Data Forms
- Flexible Framework for Sharing Expertise Across Several Disciplines
- Show Me The Money! 😊
  - Very Simple Prototype/Simulator Under Development...