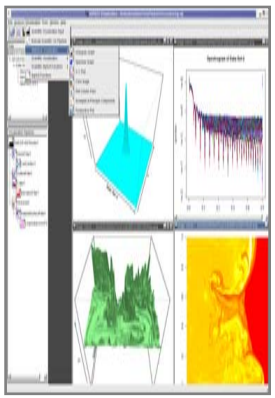
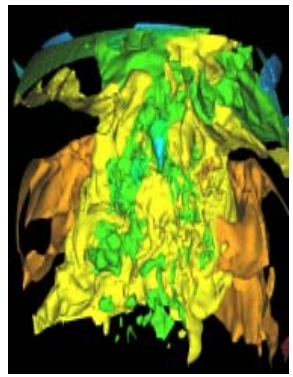
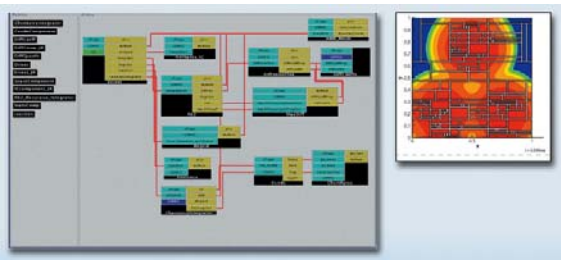
ORNL Capabilities in Support of the NVAC Visual Analytics Framework

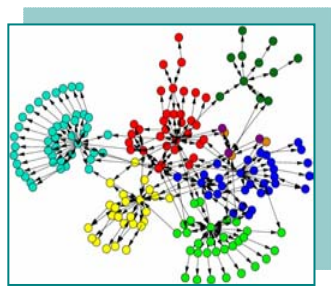


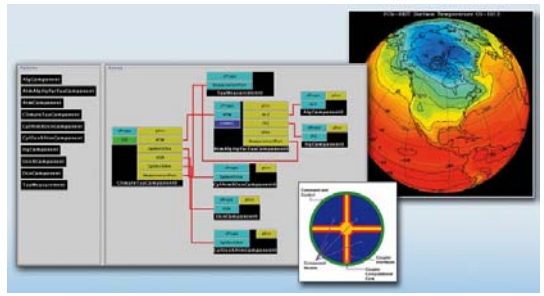


George Fann, ***James Kohl***, Robert Patton, Tom Potok and Nagiza Samatova

Oak Ridge National Laboratory







PNNL NVAC Workshop, October 28, 2004

Research sponsored by the Applied Mathematical Sciences Research Program, Office of Mathematical, Information, and Computer Sciences, U.S. Department of Energy, under contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

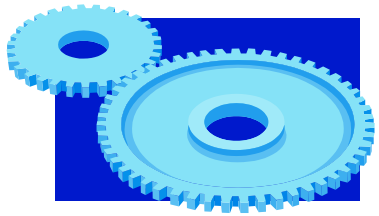
Visual Analytics Architecture…

- Sounds Like a “Hard Problem”…
  - Integration Across Agencies / Institutions…
  - Heterogeneous Databases…
  - Flexible, Extensible, Secure Services/Libraries…
  - Framework of Standards, Protocols, Utilities…
  - Collaborate Effectively, Gain Insight…
  - Scalable System Evolution…
- AAAIIIIIIEEEEEEEEEEEEEE…! ☺
  - “I have a plan…!” (Dr. Strangelove)

Work Together!

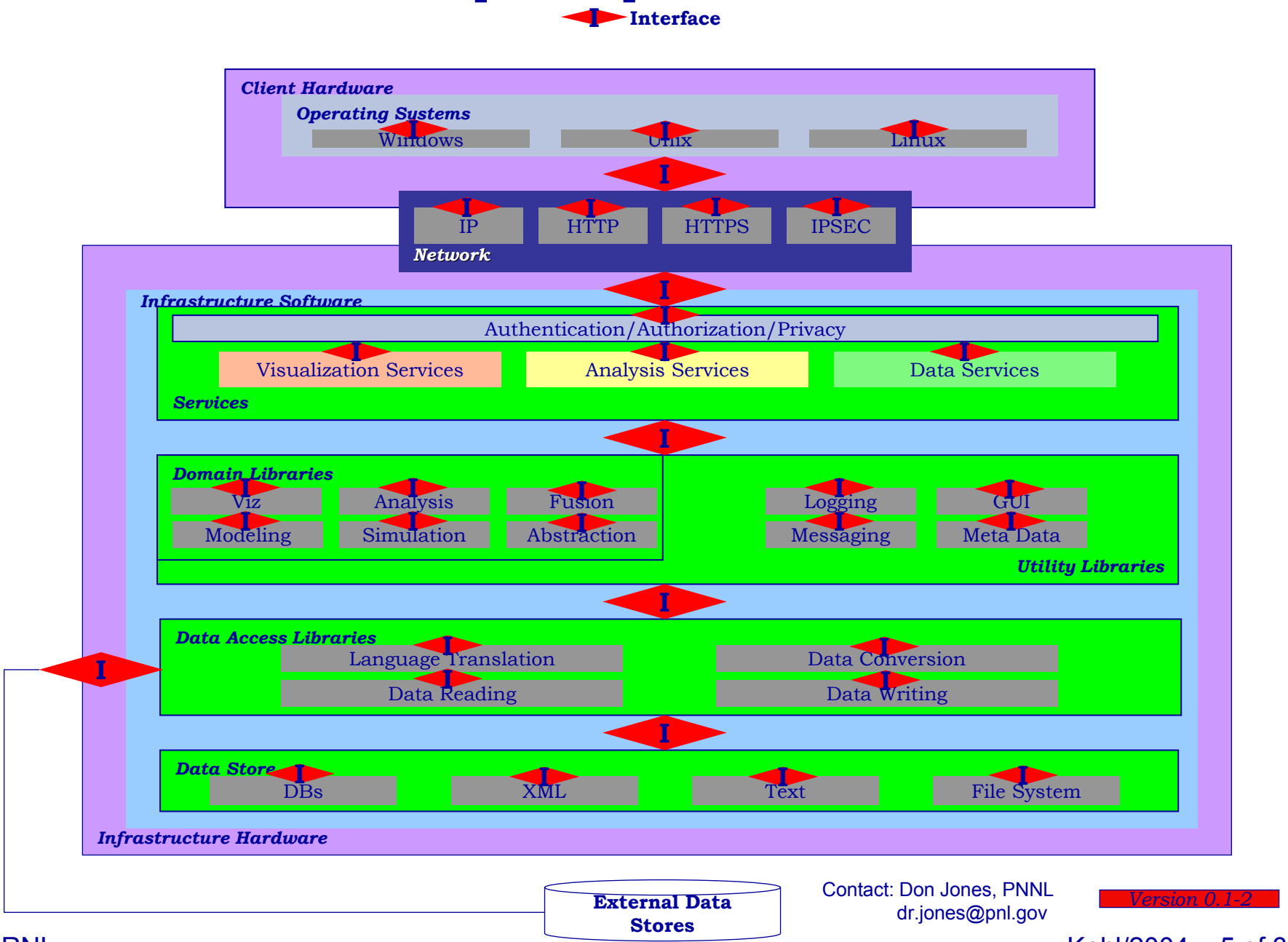
Remember
"Collaborate
or Die"? ☺

ORNL's Core Technologies
for Analysis, Processing and Visualization



- Software Components & Frameworks
  - Interfaces for Science, Interactive Visualization
- Distributed Computing & Visualization Tools
- Multiscale Image Filters, Feature Extraction
  - On CPUs, GPUs and FPGAs
- Scalable NP-Hard Graph Algorithms
- Relationships to Other DHS Projects
  - High-Perf Statistical Analysis (Parallel-R)
  - Distributed Agents (Apply to Large Data Viz)
- Scalable Visualization Infrastructure / Facility

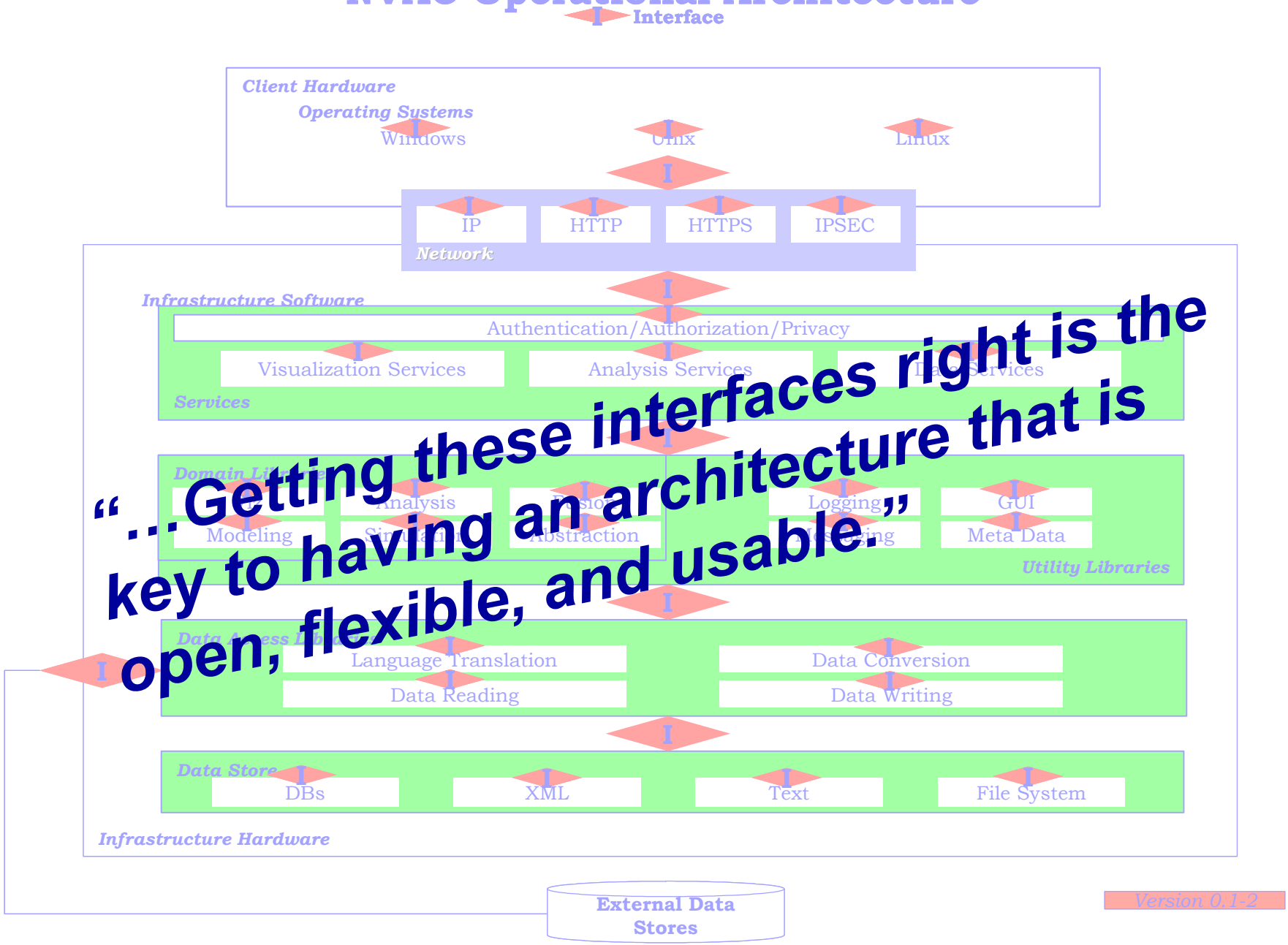
NVAC Conceptual Operational Architecture



Contact: Don Jones, PNNL
dr.jones@pnl.gov

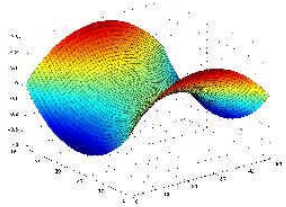
Version 0.1-2

NVAC Operational Architecture



"…Getting these interfaces right is the key to having an architecture that is open, flexible, and usable."

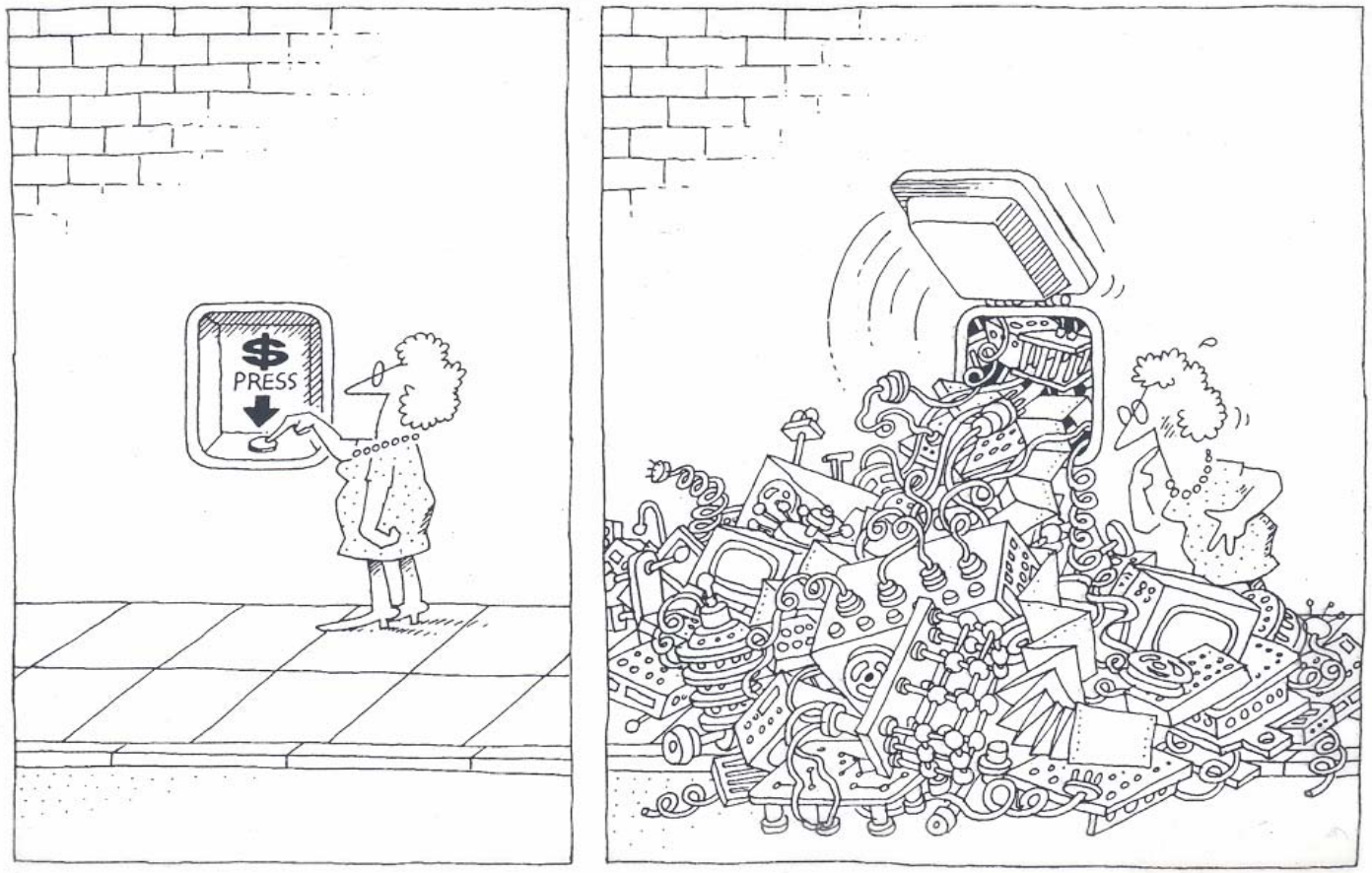
CCA
Common Component Architecture



Software Infrastructure Substrate

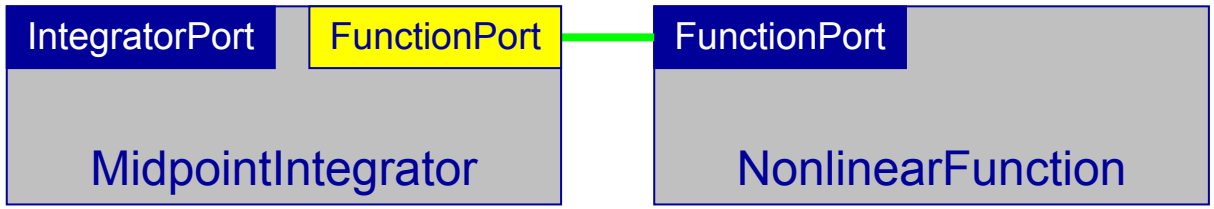
- SciDAC Common Component Architecture
  - Specification for Integrating Disparate Software
    - Manages Evolving Code Module / Library Complexities
  - Several Run-Time Execution Environments
    - “Frameworks” Provide Parallel / Distributed Sandboxes
    - Remote Communication “Bridges” Under Development
  - Platform for Defining Interface Standards
    - Code Re-use, Interoperability, Version Management
      - (Just a mechanism, “standards” require *agreement*...!)
    - Already Being Applied ~ Chemistry, Fusion, Viz (DiVA)

Why Components?



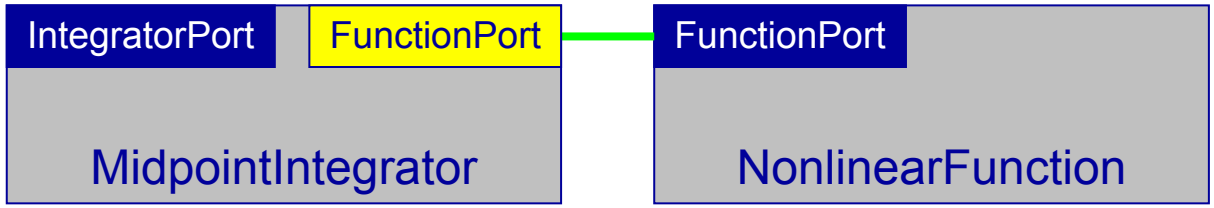
The task of the software development team is to engineer the illusion of simplicity *[Booch]*.

CCA Concepts: Components



- Opaquely encapsulate implementation details
  - Expose only well-defined port *interfaces*…
- Components provide/use one or more ports
- Components include some code which interacts with a CCA framework
  - Describes internally implemented ports…
  - Dependencies for required external ports…

CCA Concepts: Ports



- All component interaction is through well-defined *port interfaces*
  - In OO languages, a port is a class or interface
  - In Fortran, a port is a bunch of subroutines or a module
- Components may *provide* ports ~ implement the class or subroutines of the port (“Provides” Port)
- Components may *use* ports ~ call methods or subroutines in the port (“Uses” Port)
- Links between ports denote a *procedural* (caller / callee) relationship, ***not dataflow!*** (no arrows… ☺)
  - e.g., FunctionPort could contain: *evaluate(in Arg, out Result)*

CCA Concepts: Frameworks

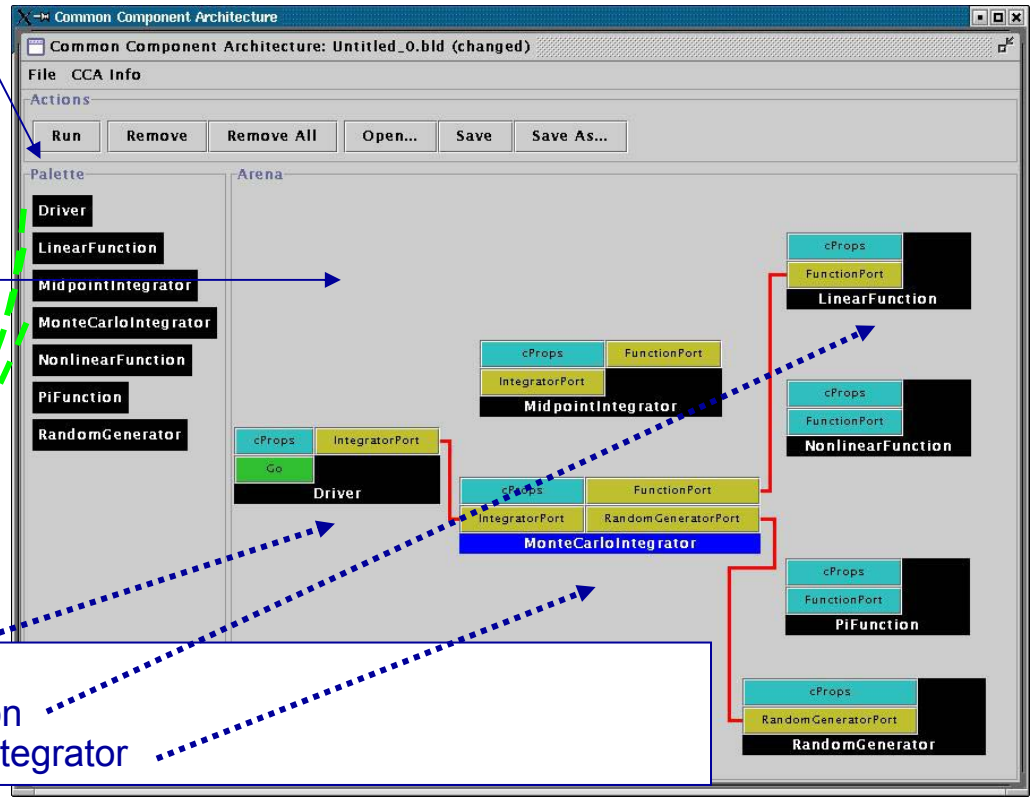
- The framework provides the container to “hold” components and compose them into applications
- Frameworks allow connection of ports without exposing component implementation details
- Frameworks provide a small set of standard services to components

- *Currently:* specific frameworks support specific computing models (parallel, distributed, multithreaded…)
- *Future:* full flexibility through integration or interoperation
  - Framework “Bridges” for hybrid control flow combinations…

User Viewpoint: Loading and Instantiating Components

- Components are: code + metadata
- Using metadata, a **Palette** of available components is constructed
- Components are instantiated by user action (i.e. by dragging from **Palette** into **Arena**)
- Framework calls component's constructor, then setServices(svc)

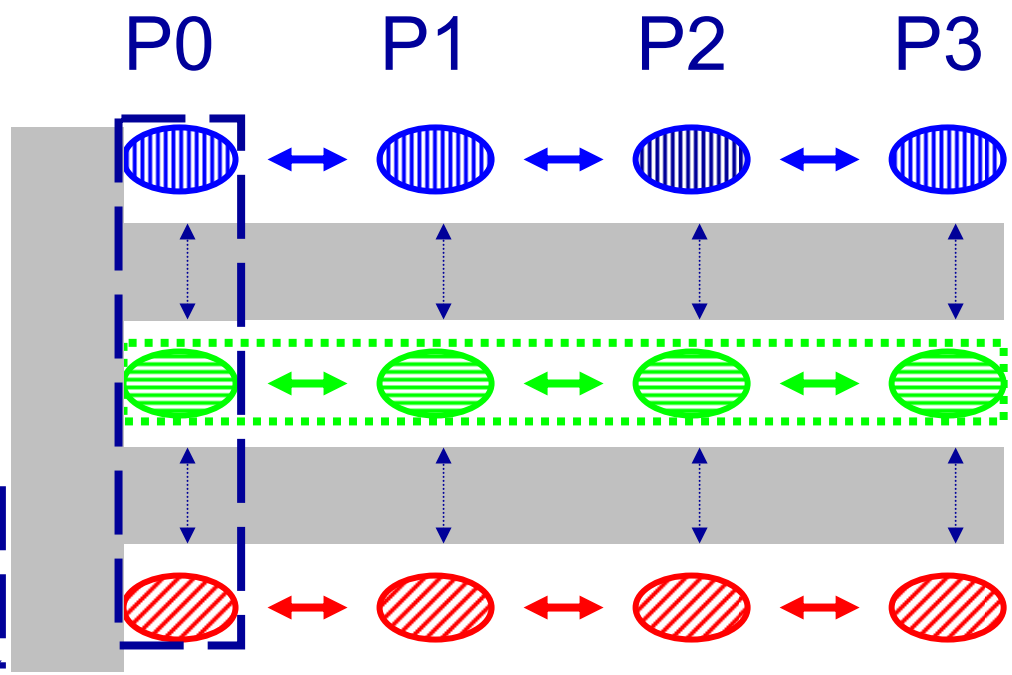
- Details are framework-specific!
- Ccaffeine currently provides both command line and GUI approaches



create	Driver	Driver
create	LinearFunction	LinearFunction
create	MonteCarloIntegrator	MonteCarloIntegrator

CCA Frameworks Stay "Out of the Way" of Component Parallelism

- Single component multiple data (SCMD) model is component analog of widely used SPMD model
- Each process loaded with the same set of components, wired the same way
- Different components in same process "talk to each" other via ports and the framework
- ***Same component in different processes talk to each other through their favorite communications layer (i.e. MPI, PVM, GA)***



Components: Blue, Green, Red

Framework: Gray

MCMD/MPMD also supported

Other component models ignore parallelism entirely

Interfaces, Interoperability, and Reuse

- Interfaces define how components interact…
- Therefore interfaces are key to interoperability and reuse of components!

- In many cases, "any old interface" will do, but…
- Achieving reuse across multiple applications requires *agreement* on the same interface!

- "Standard" or "community" interfaces facilitate reuse and interoperability ~ critical mass…
  - Typically domain specific (climate, chemistry, fusion, viz)
  - Formality of "standards" process varies…
  - Significant initial investment for long-term payback

End-to-End Distributed Computing

Scalable High-Resolution Visual Display Facility

Front-End User Interface and Display (Desktop, PowerWall, reCAVE, Big Bertha…)

CUMULVS:
Collaborative Interactive System For Visualization, Steering, Coupling and Fault Tolerance

Run-Time Monitoring and Control Middleware Visualization, Steering, Fault Tolerance

Concurrent Programming Model:Servers/sMPPs (PVM, MPI, OpenMP…)

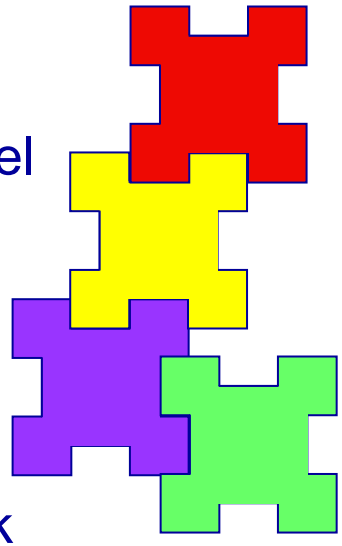
Harness:
Adaptive Reconfigurable Distributed Virtual Machine Environment built on High-Performance Clusters

High-Performance Server Parallel/Distributed Execution Environment

Harness

Flexible Robust Server, Parallel Operating Environment

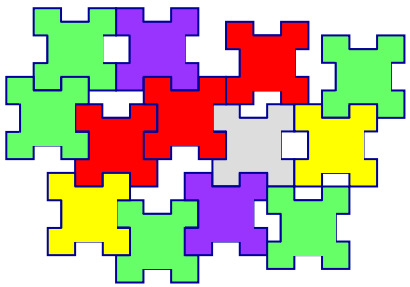
- Parallel “Plug-In” Software Components (a la CCA)
- Dynamically Adapt to Changing Program Needs
  - Not Forced to “Fit” into Monolithic Infrastructure
- Distributed Peer-to-Peer Control
  - No Single Point of Failure, Robust Protocols
    - Consistent Global State / Database Updates in Parallel
- Share Resources and Migrate Applications
  - Servers, Parallel Computers, Clusters, PCs…
  - Merge and Split Virtual Machine Sets
    - Owner Controls Access to CPU, Storage and Network



Harness Features

Self-Assembling Virtual Machine

Parallel Plug-in Software Components Provide the Capabilities:
Network, Programming Model,
Numerical Libraries,
Front-End Tools / GUIs.



Fault-Tolerant
MPI Extensions
~ "FT-MPI"

Long-Running Applications
Often Exceed the MTBF for Large
High-Performance Computing Systems!
→ *Distributed Control Algorithms Maintain*
Global State in the Presence of Faults and Failures

Robust Fault-Tolerant Environment

Multiple Tools, Versions
and Parallel Paradigms
Used Simultaneously

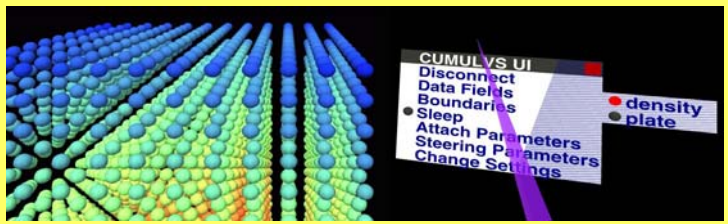
PVM PAPI MPI Pablo

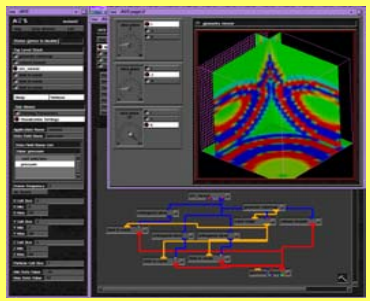
CUMULVS

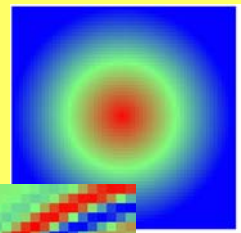
Interacting with Live Data Applications

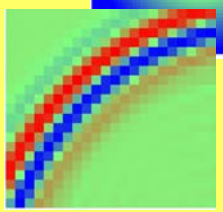
Visualization

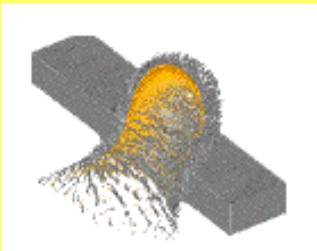
Dynamic Attachment of
Multiple Independent Viewers







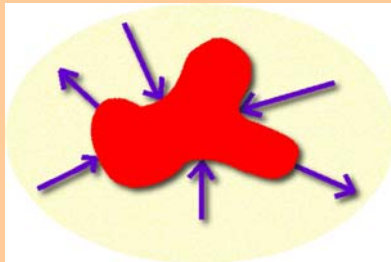




Steering

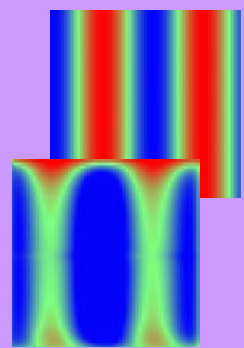
Multiple Simultaneous "Steerers"
→ Updates Applied in Parallel Unison
→ Algorithmic or Model Parameters,
or Truncate Runs Gone Awry…

Explore
Non-Physical
Effects



Model Coupling

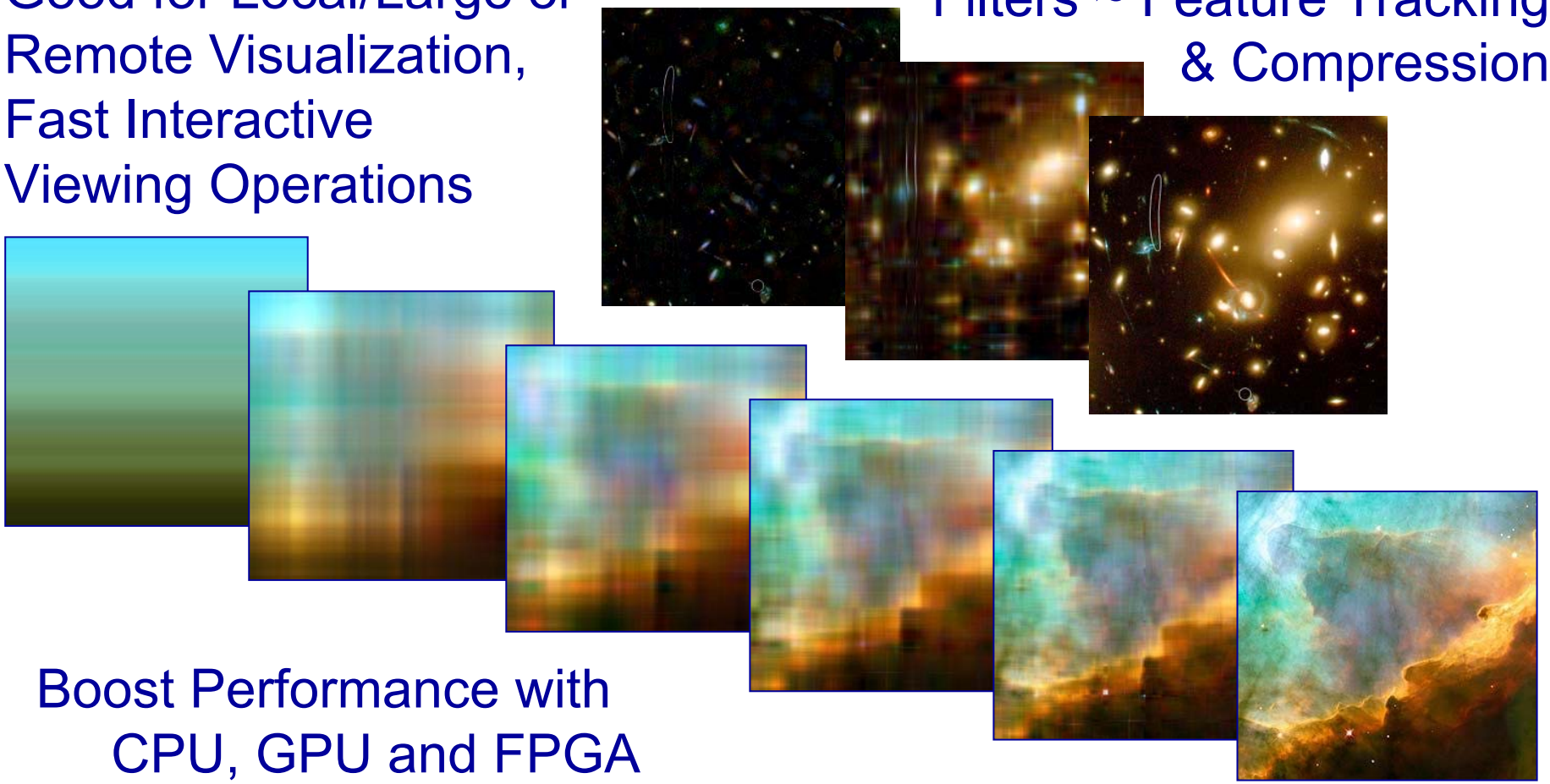
Natural Extension of Protocols
→ Models Share Parallel Data
→ Parallel Data Redistribution
(CCA "MxN")



Parallel Multiscale Compression and Image Reconstruction

Good for Local/Large or Remote Visualization, Fast Interactive Viewing Operations

Filters ~ Feature Tracking & Compression



Boost Performance with CPU, GPU and FPGA

Contact: George Fann gif@ornl.gov

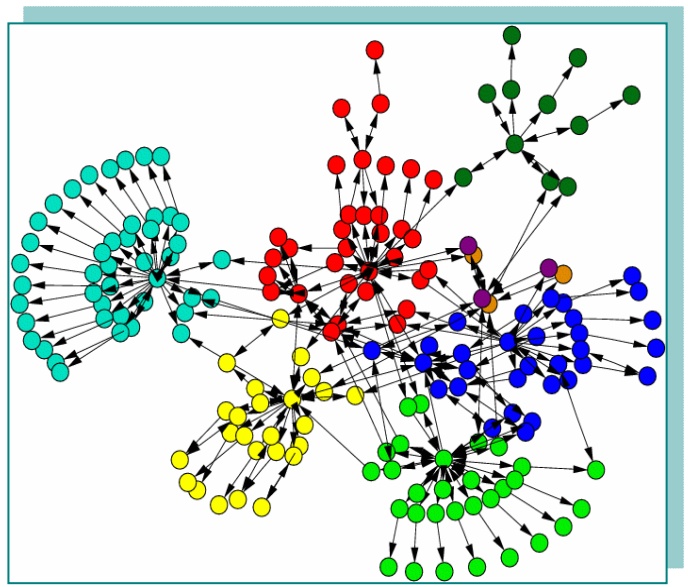
Intelligent Queries over Semantic Graphs

Intelligent query processing and advanced information analysis presents a significant computational challenge in DHS.

→ Impose Semantic Structure on Relational Graphs…!

Example Queries beyond Google:

- Identify a minimum group of people that are related to all other people (**Minimum Vertex Cover**);
- Discover a suspicious pattern of interest in the DB (**Sub-graph Isomorphism**);
- Find the largest group of cities such that every two would be affected by a disease spreading from one to the other, or enumerate all such groups (**Maximum or Maximal Clique**);
- Extract group of people common to two or more suspicious organizations, including all relations between them (**Maximum Common Subgraph**).

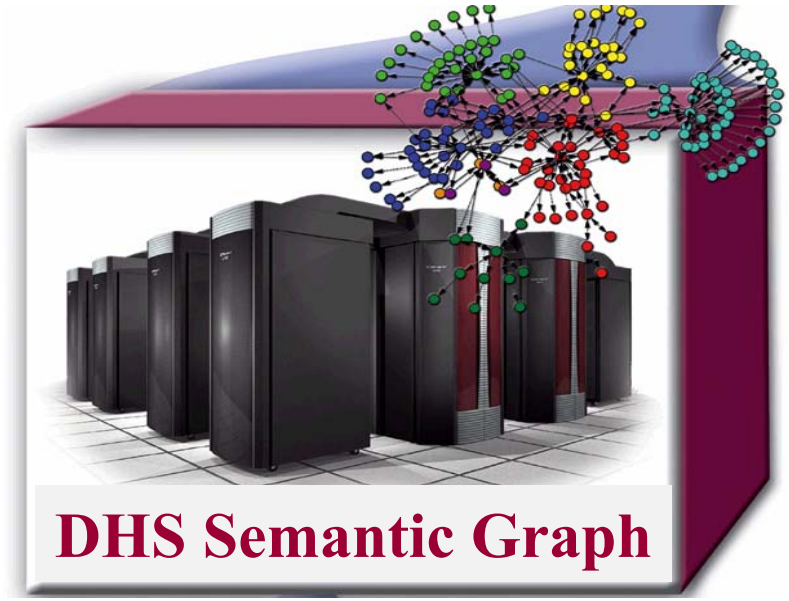


Scalable, Parallel Semantic Graph Algorithms

Provide library of scalable, parallel graph algorithms for polynomial time solutions of NP-hard graph problems.

Library Features:

- **Exact polynomial** solutions via **Fixed Parameter Tractability** (FPT) reduction:
  - Minimum Vertex Cover
  - Sub-graph Isomorphism
  - Maximum or Maximal Clique
  - Maximum Common Subgraph
- Fastest & most scalable solutions (in terms of problem size) versus those reported in literature.
- Supports different types of graphs: directed, undirected, labeled and unlabeled.
- Contact: Nagiza F. Samatova (samatovan@ornl.gov)



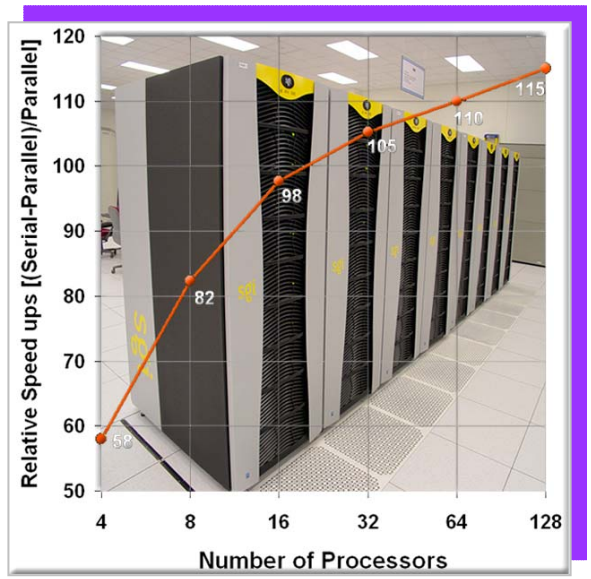
Example Semantic Graph:
12,422 vertices and >100M edges
Maximum Clique: 399 vertices

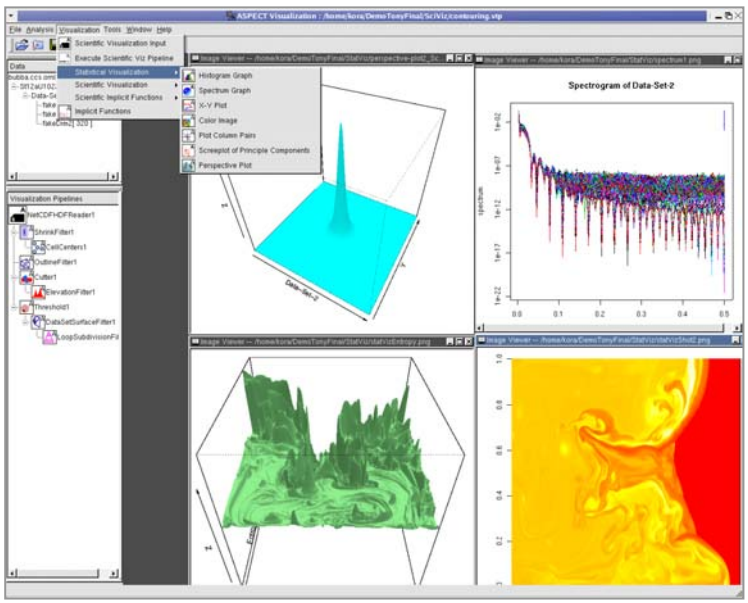
Parallel-R for Large-Scale Statistical Computing

Parallel-R library enables task and data parallelism within the existing statistical data analysis system, called R (http://www.r-project.org).

Features:

- Interface maintains look and feel of serial R.
- Enables user to set up parallel environment, distribute data, and perform required task in parallel, to obtain the result through a simple R function call.
- Provides scalability and high-performance.





- Available as open source; mirrored in 25 mirror sites, more than 17 countries.
- Contact: Nagiza F. Samatova (samatovan@ornl.gov)

http://www.aspect-sdm.org/Parallel-R

Parallel-R for Large-Scale Statistical Computing

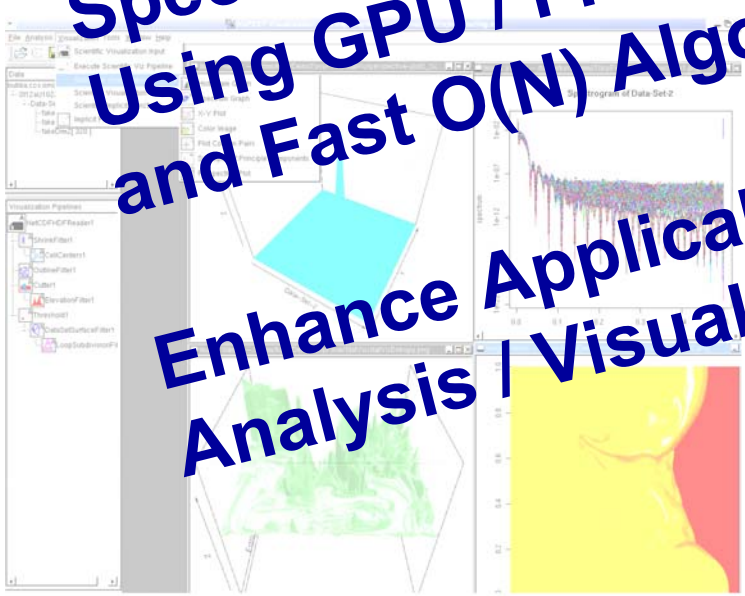
Parallel-R library enables task and data parallelism within the existing statistical data analysis system, called R (http://www.r-project.org).

Features:

Interface maintains look and feel of serial R.
Enables user to set up parallel environment, distribute data, and perform required task in parallel to obtain the result through simple R function call.
Provides scalability and high performance.

Specialize Existing Parallel-R Capabilities Using GPU / FPGA Optimizations, and Fast O(N) Algorithms…

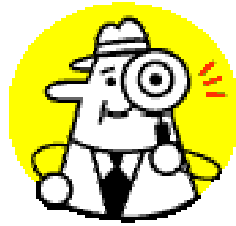
Enhance Applicability for Distributed Analysis | Visualization Scenarios…



Available as open source; mirrored in 25 mirror sites, more than 17 countries.
Contact: Nagiza F. Samatova (samatovan@ornl.gov)

http://www.aspect-sdm.org/Parallel-R

Distributed Agent Visualization



- Large Dataset Visualization is Hard Enough…
  - Distributed Data Sources/Archives ~ Much Worse!
- Traditionally Need Some Data Centralization:
  - Collect Data for HPC Analysis & Rendering…
- Apply Agents ~ Natural Paradigm for Dist Data
  - Distribute the Computation to Match Data Org
  - Agents Wrangle Analysis/Storage/Rendering…
  - In-Place Data Fusion/Imaging ~ Viz More/Faster
    - Capitalize on Inherent Parallelism, Remote Viz Tech
  - Forms the Basis for a Virtual Data Cache Engine…

Statistics and Data Science Expertise

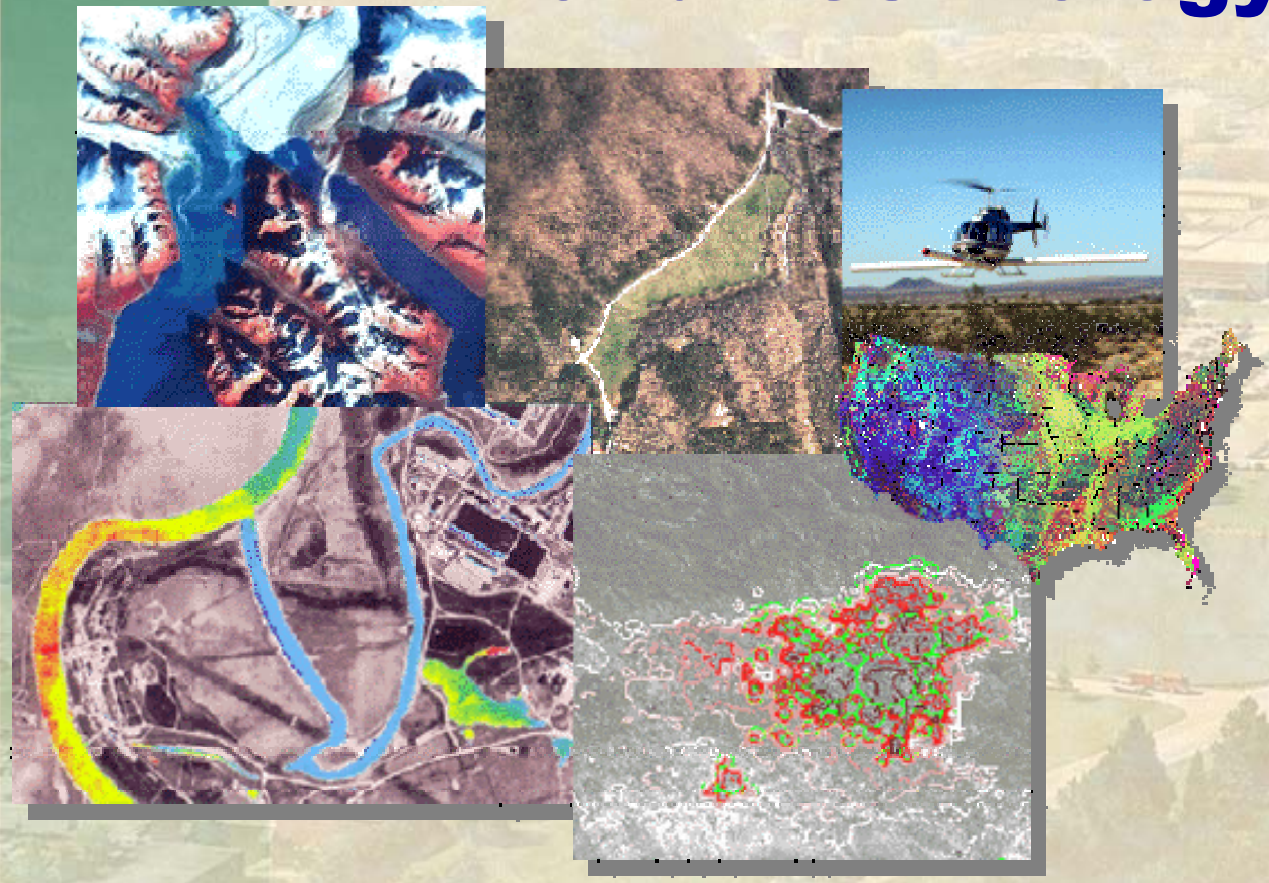
Leading-edge statistical and information technologies with quantitative rigor for scientific investigations.

Methodology strengths include:

- Scalable Analysis and Visualization for High-Dimensional Data
- Information Integration, Infrastructure, Semantic Mediation, Ontologies, Custom Data Structure/Layout Design, Graph Theory
- Biostatistics, Multivariate Analysis, Regression Analysis, Graphical Methods
- Risk, Sensitivity, and Uncertainty Quantification
- Multivariable Testing, Sampling, Process Control and Optimization

Areas of application: Lie Detection, Chemistry, Biology and Genomics, Astrophysics, Climate, Environmental Science, National Security, Forensics, Simulation Science, Epidemiology, Fusion Science, Transportation and Automotive, Health and Safety, Grid Technologies, Manufacturing, Future Combat Systems, Remote Sensing, Computer Network Security, Sensors

ORNL: A pioneer in Geospatial Science and Technology



- Rich History Spanning 30 years
- Defining Geospatial Research Agendas
- Developing New Algorithms, Software, and Data
- Conducting Verification & Validation Studies

Geospatial Technology Highlights

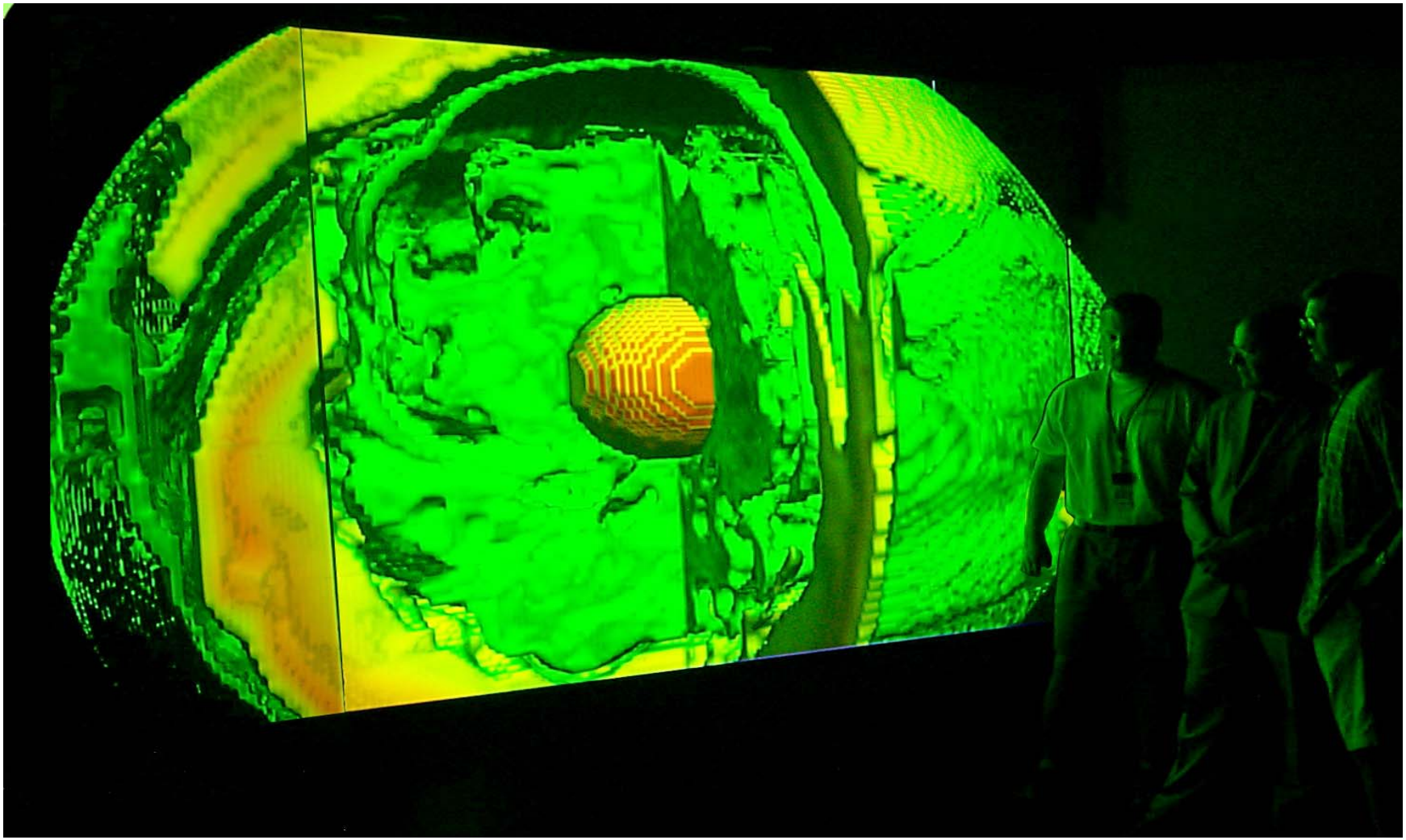
- Geospatial Science
  - Multi-resolution data analysis
  - Advanced spatial modeling
  - Data integration and visualization
  - Geographic Information Systems (GIS)
  - Decision Support Systems (DSS)
- Scientific Data Access
  - Collection, Distribution, and Management
  - Large scale data manipulation

EVEREST

Exploratory Visual Environment

35 MPixel Display, 8’x30’

Astrophysics



Simulation results of core collapse supernovae mechanisms from John Blondin and Tony Mezzacappa as a part of the SciDAC Terascale Supernova Initiative (TSI).

Viz for Leadership Computing

- Data Sources: SciDAC, GTL, Sensors (uCAT)…
- Remote and Streaming Visual Content Delivery
- Server and Parallel Algorithms + GPUs/ReadBack
  - Feature Extraction/Solvers, Multires Analysis
- Scalable Viz / Analysis Hardware
  - 256-CPU (SSI) SGI Altix, 2TB Shared RAM
  - Analysis & Rendering Engine Cluster
    - 64 Dual Opterons, GigE & Quadrics, 2GB/node, Tens of TBs
    - nVidia FX5900, Quadro FX3000G (244 BTexels/sec theor)
  - Additional Display Capabilities
    - Reconfigurable CAVE, Immersadesk, 12MPixel LCD Tiled Display
    - 2 - 9MPixel IBM T221, Stereo Geo Wall (UIC ~ Jason Lee)
    - HP Sepia2…

Summary: Potential ORNL Collaborations
with PNNL NVAC / Visual Analytics Architecture

- Component Frameworks, Dist Computing, Viz
  - CCA, Harness, CUMULVS
- Multiscale Image Filters, Feature Extraction
  - On CPUs, GPUs and FPGAs
- Scalable NP-Hard Graph Algorithms
- High-Perf Statistical Analysis (Parallel-R)
- Distributed Agents (Large Data Viz)
- High-Resolution Display Infrastructure