

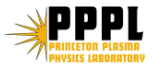
# End-to-End Computing at ORNL: Preparing for the Exascale

Presented by

## Scott A. Klasky


Scientific Computing  
National Center for Computational  
Sciences

In collaboration with:



# It's all about the applications

Application	Developer
GTC	Y. Xiao, Z. Lin
GTS	S. Ethier, W. Wang
M3D	L. Sugiyama
M3D-K	G. Y. Fu
S3D	J. Chen, R. Grout, R. Sankaran
Pixie3D	L. Chacon
PAMR	G. Allen, P. Laguna
XGC-P	S. H. Ku, C. S. Chang
XGC-0	C. S. Chang, J. Cummings, G. Y. Park
XGC-1	C. S. Chang, S. H. Ku
GEM	Y. Chen, S. Parker, W. Wang
Gysela5D	P. Diamond, G. Dif Pradiler
Chimera	B. Messer, T. Mezzacappa
+ many more	Climate, Earthquake, Nuclear physics, ...

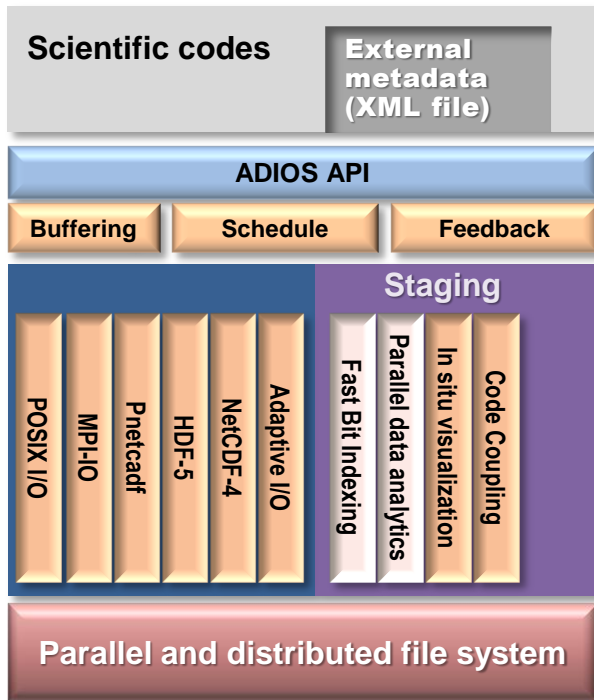
- Simple principles
- KISS 
- Make it portable, scalable, fast, reliable, accurate



# Goals for the exascale

- Reduce **IO overhead** for all platforms
- Reduction of data movement for I/O pipelines
- **Easy to use**
- **Portable, scalable performance with QoS**
- Both synchronous and asynchronous transports without code changes
- Memory-based code coupling in the I/O layer
- Exploration into methods to assure resiliency for simulations
- Real-time provenance capture
- New file formats for petascale and exascale
- Creation of a robust I/O staging framework for in situ analysis and reduction of extreme-scaled application data
- Create a toolkit of I/O modules such as FastBit indexing that scientists can easily utilize
- Create and extend programming models for end scientists to utilize **in situ I/O stream processing**

# ADIOS (I/O componentization)



## Simple API

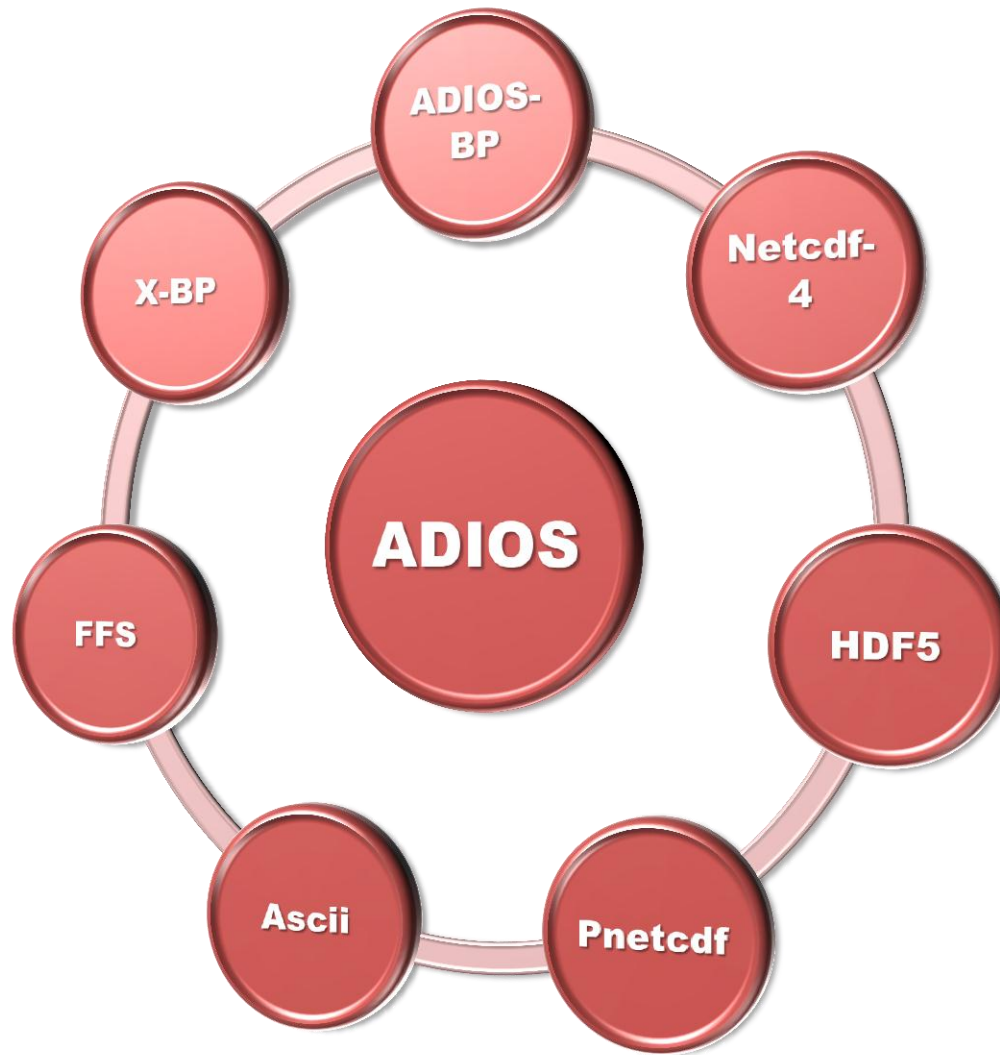
```
adios_open (handle, group,  
outputfile, "w", err)
```

```
#include "gwrite_genarray.fh"
```

```
adios_close (handle, err)
```

- Allows researchers to create research methods that can be tested in “real-codes” with no changes
- File formats created for increased resiliency and (read and write) performance for current and future performance
- Staging methods allow the creation of I/O pipelines for complex, in situ workflows
- ADIOS is an extendible method that allows new statistics to be automatically captured in the file output

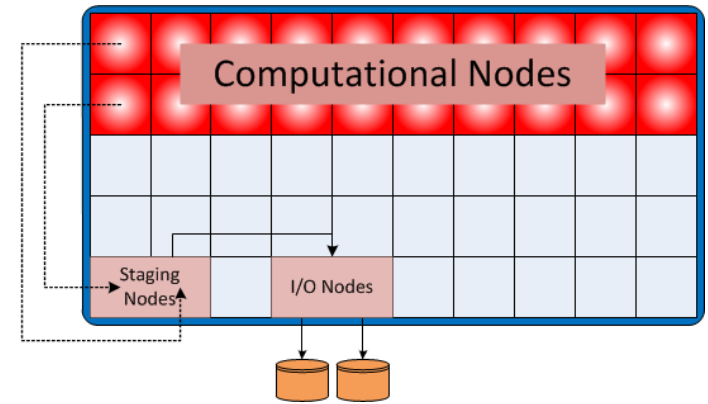
# I/O componentization for ease of use for application performance



# In situ I/O pipeline analysis and visualization for simulation data

- ADIOS includes 3 methods for I/O staging

- DataTap
- DataSpaces
- NSSI



- I/O staging decouples I/O performance from the file system

- Allows data to be analyzed, reduced, indexed, and visualized, before touching the file system

- Allows analytics to be “plug-and-play” in the staging area

# Creation of I/O pipelines to reduce file activity

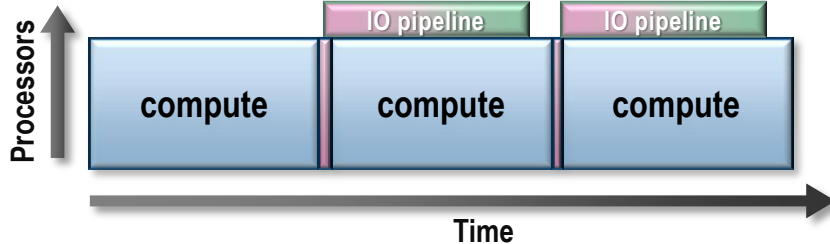
## Traditional approach



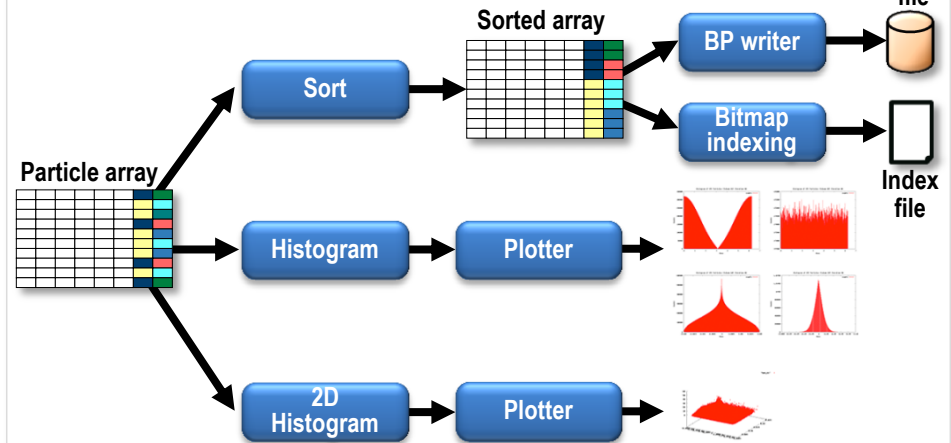
## In-Compute-Node (ICN) approach



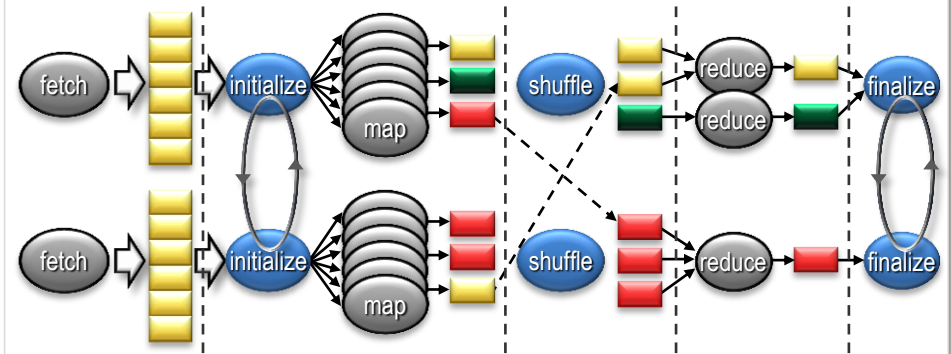
## Asynchronous I/O pipeline approach with DataTap and SmartTap



## Example of an I/O pipeline



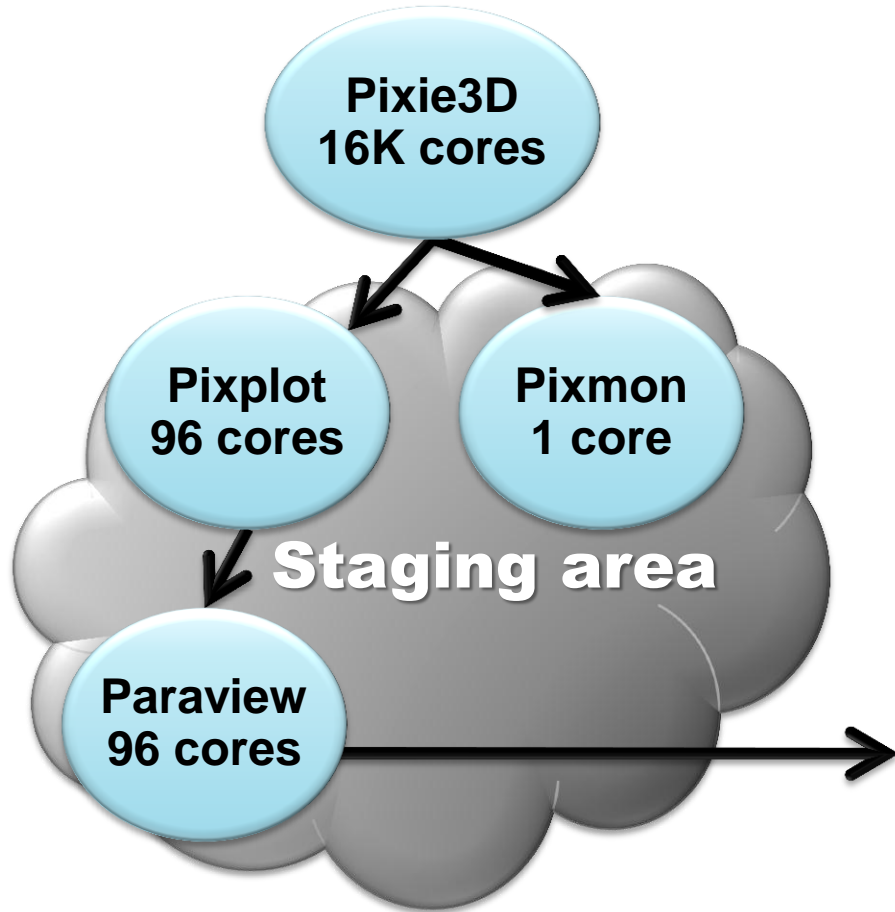
## Streaming processing in staging area



### Differences with MapReduce

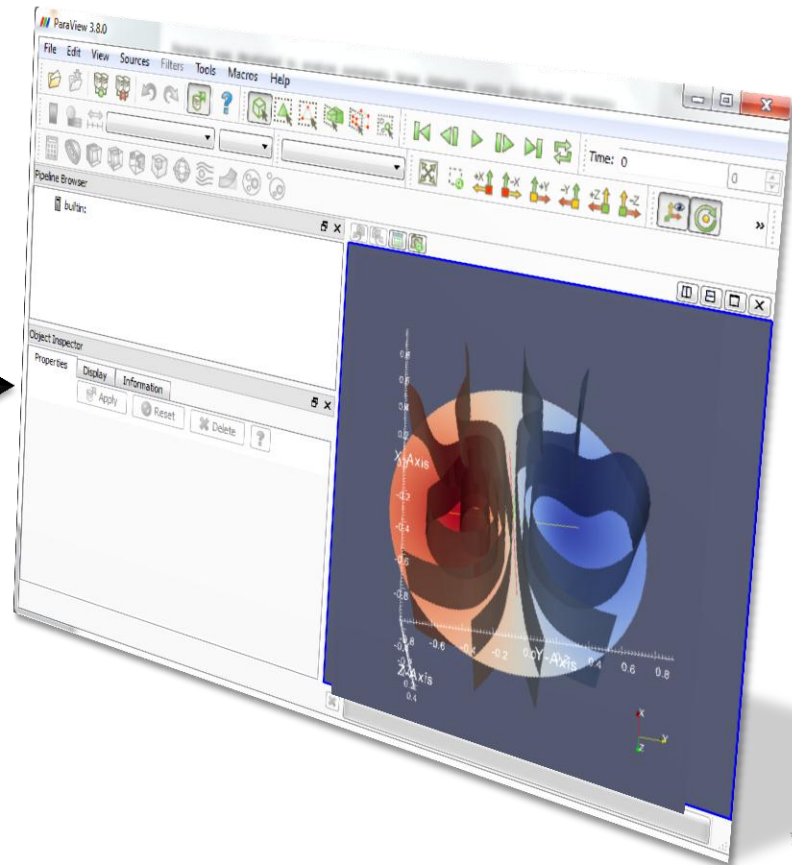
- Two-pass streaming processing (in compute nodes or staging area)
- In-memory storage for speed
- Customizable shuffling phase and additional initialize/finalize

# In situ workflows



Example: Analysis and visualization tasks executed concurrently with a fusion code.

Demonstrated at SC'11 tutorial  
"In-situ visualization with Paraview"



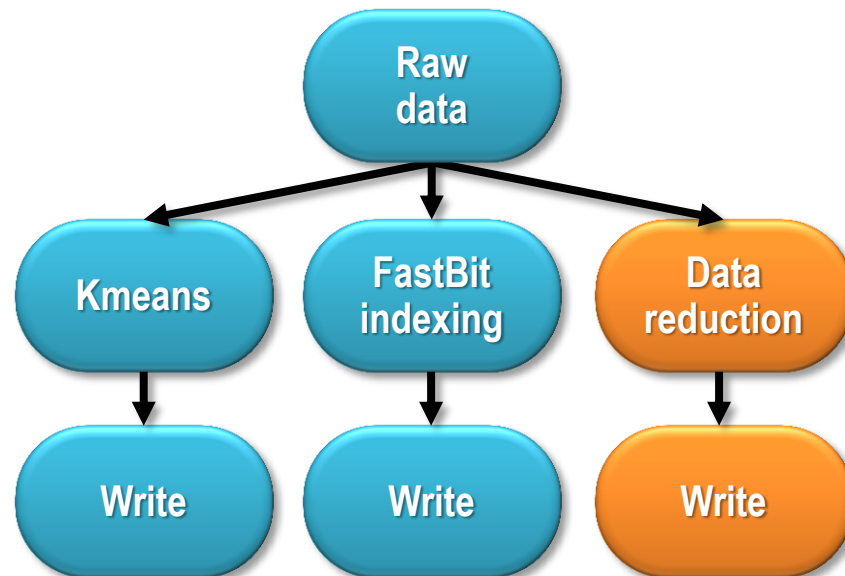


# In situ workflows with DataTap

For the exascale, we are creating in situ workflows

- Allows data to be analyzed, reduced, indexed, and visualized, before touching the file system
- Consist of a runtime environment that can schedule work based on user priorities, expected completion times
- Workflows are performed in the staging area and is fully dynamic

## Creation of an in situ workflow engine

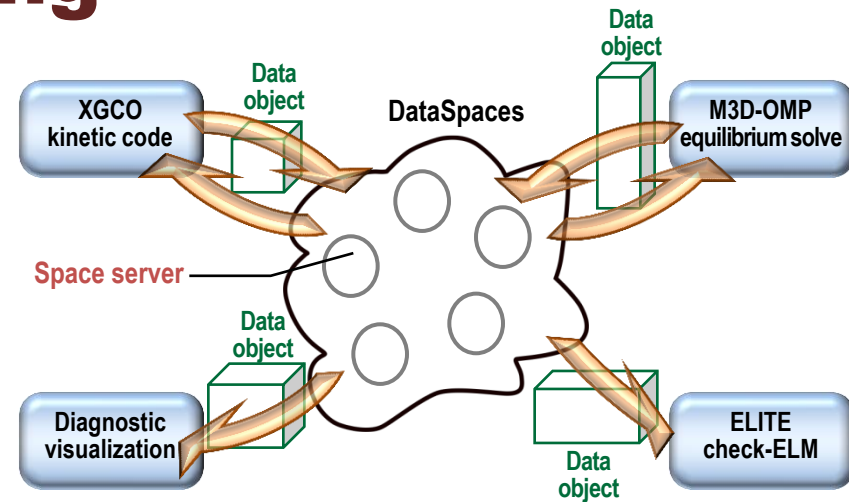


# Programming models for I/O pipelines

- **Programming models allow for a PGAS-like abstraction**
- **Use of Directed Acyclic Graph (DAG) to find optimal scheduling on systems with multicore CPU and GPU accelerators in conjunction with the runtime system**
- **Creation of a programming model in the staging area which will allow users to easily create “plug-ins” to exploit special-purpose devices, such as GPGPUs**
- **The execution environment will be responsible for the overall scheduling of tasks on the available hardware**
  - **Awareness of many potentially beneficial relationships between independent tasks, such as large overlaps in the data accessed**
  - **Exploration into techniques that may transfer information to the runtime scheduler to help the execution environment ensure good load balance and work scheduling**

# ADIOS with DataSpaces/DART for in-memory code coupling

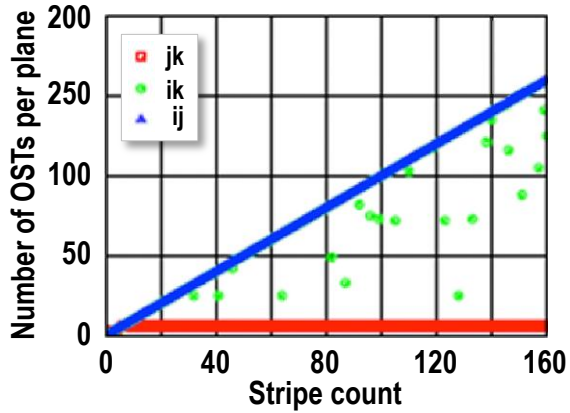
- Semantically specialized virtual shared space
- Constructed on-the-fly on the cloud of staging nodes
  - Indexes data for quick access and retrieval
  - Implements a distributed hash table on a dynamic set of staging nodes
  - Provides asynchronous coordination and interaction and realizes the shared-space abstraction
- Complements existing interaction, coordination mechanisms
- Supports complex geometry-based queries
- In-space (online) data transformation and manipulations
- Robust decentralized data analysis in-the-space



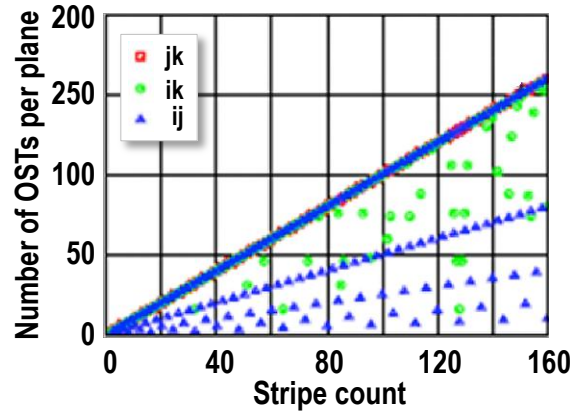
# ADIOS read performance and performance for future file formats

- Goal is to reduce the I/O overhead for current and future systems for both reading and writing
- Using a special layout option within ADIOS, we can speed up reading speeds for common I/O reading patterns, up to 37x, compared to conventional logically contiguous data layouts
- Aggressive data reductions will become part of the I/O pipeline for exascale simulations
- Application Log File (ALF) formats will become critical for performance
- Usage of advanced techniques to make use of low-latency devices (SSD-like devices) alongside rotating media for future file formats
- Fast, aggressive lossy and lossless compression techniques including
  - Multiresolution/temporal data reduction techniques
  - Data deduplication techniques

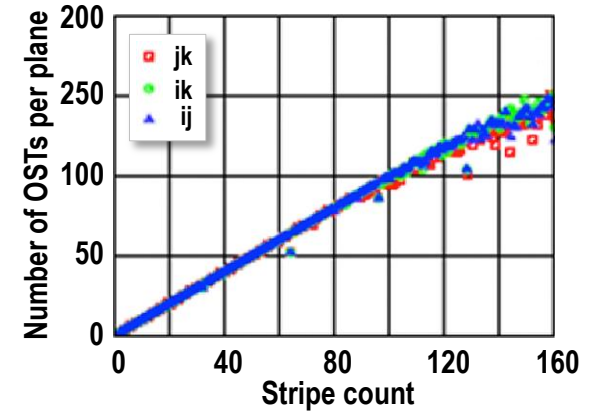
# Read performance of analysis data



(a) Logically contiguous

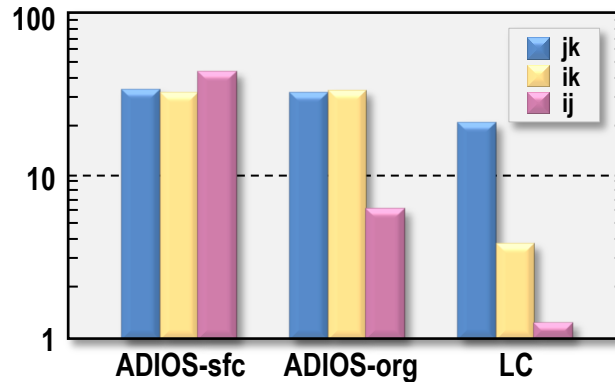
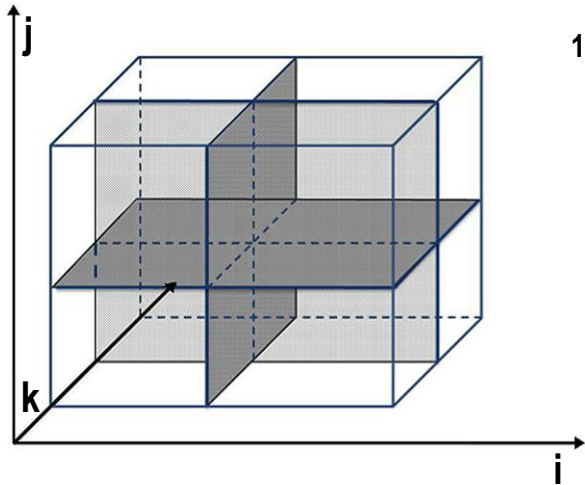


(b) Original ADIOS

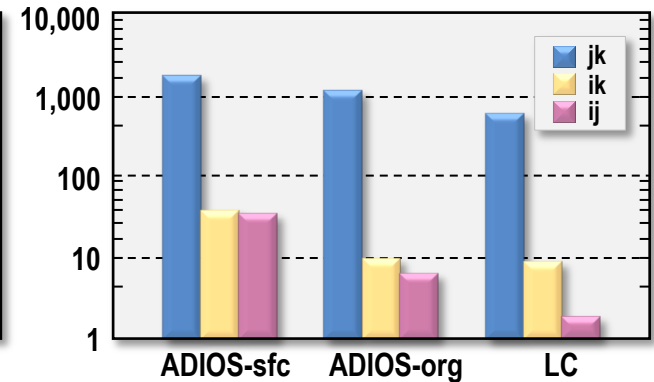


(c) New ADIOS

## OST utilization variation of a 2D plane on 3 dimensions



(a) Small (stripe = 128)

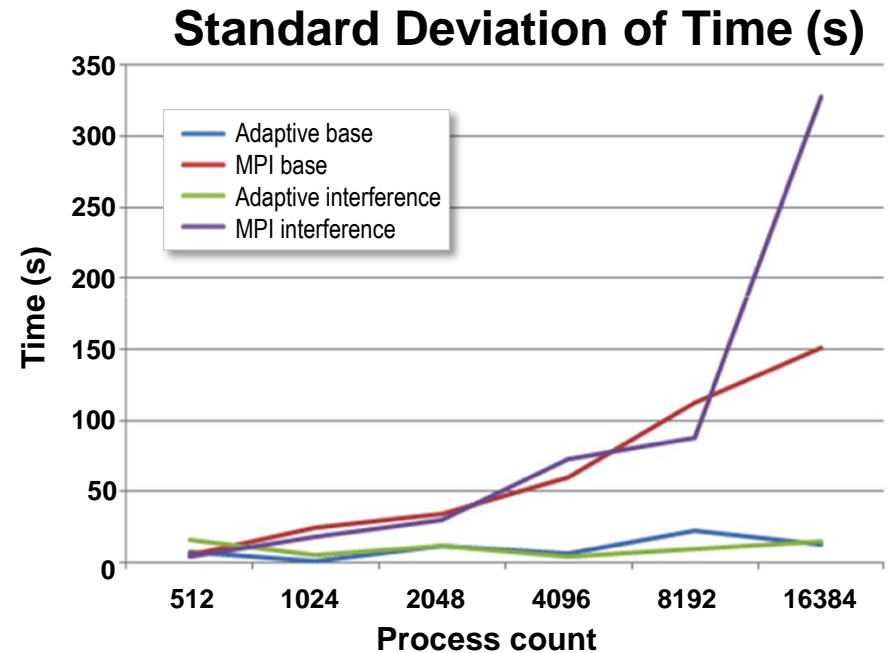
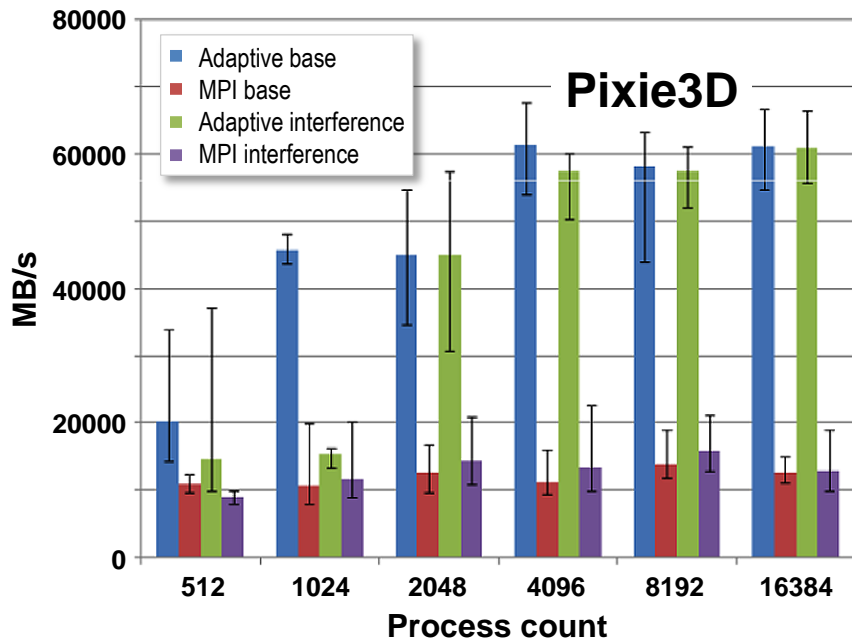


(b) Extra large (stripe = 128)

## I/O performance reading in a 2D plane from 3D grid

# Reducing the I/O variability

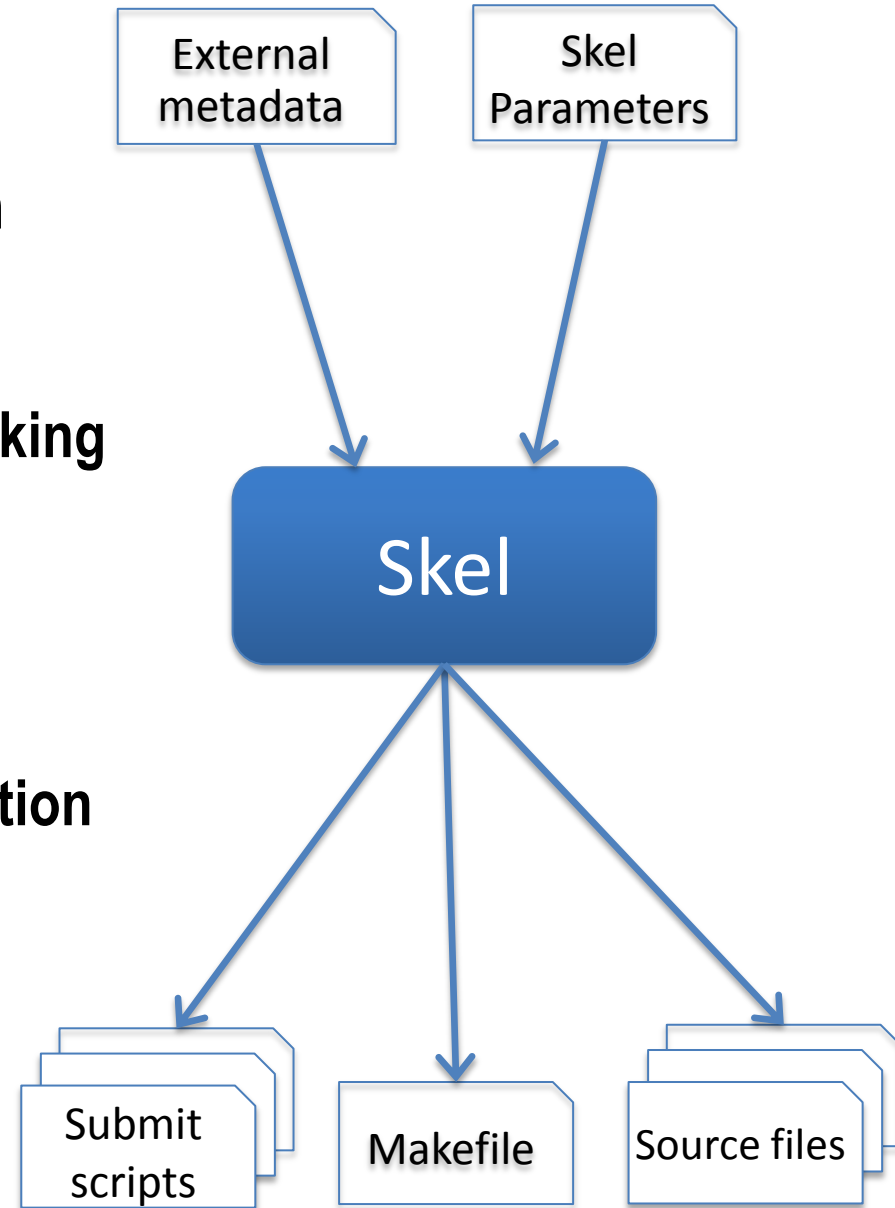
- Utilize the maximum number of storage targets
- Schedule the processes to avoid internal interference
- Shift work from slower to faster storage targets for external interference



**New methods got almost 100% of peak performance of loaded I/O system, with minimal degradation of I/O performance at scale (green, blue lines)**

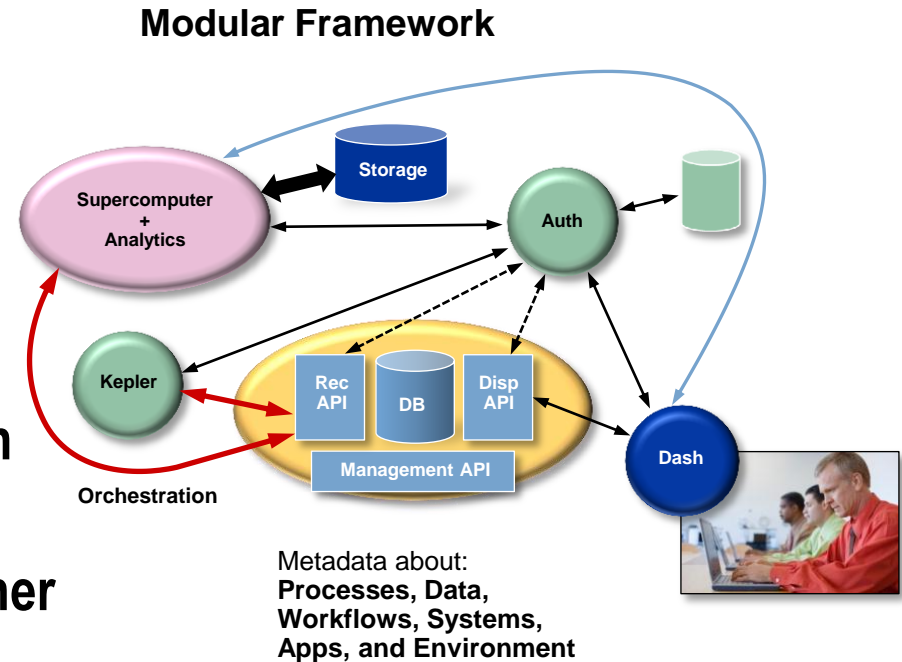
# Skel

- **I/O Skeletal Applications**
  - Simplified version of a production application, omits application's computation & communication
  - Useful for performance benchmarking and infrastructure testing
  - Generated automatically by Skel
- **Skel**
  - Allows simple creation and execution of I/O skeletons, easy updates
  - Creation, building and execution are standard across many apps
  - Allows for a large, extensible collection of I/O benchmarks



# Provenance is key

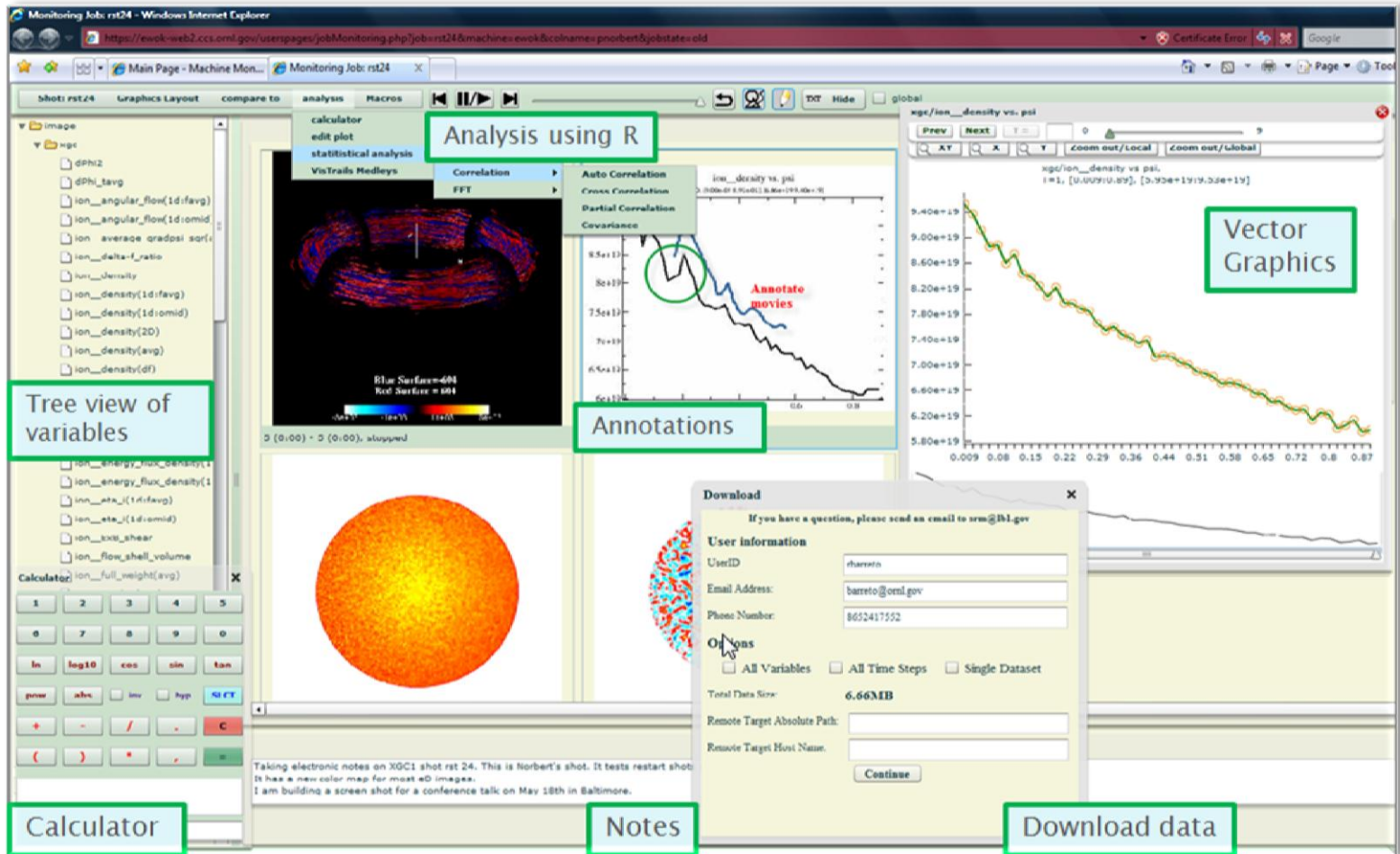
- One of the challenges in this context is the tracking of the data files that can be produced in very large numbers during stages of the workflow, such as visualizations
- The Kepler provenance framework collects all or part of the raw information flowing through the workflow graph
- This information then needs to be further parsed to extract metadata of interest; this can be done through add-on tools and algorithms
- We show how to automate tracking of specific information such as data file locations



P. Mouallem, M. Vouk, S. Klasky, N. Podhorszki, R. Barreto, "Tracking Files Using the Kepler Provenance Framework," *Proceedings of 21st International Conference on Scientific and Statistical Database Management, SSDBM'09 (2009)*



# Where do we see the data? What about collaboration?



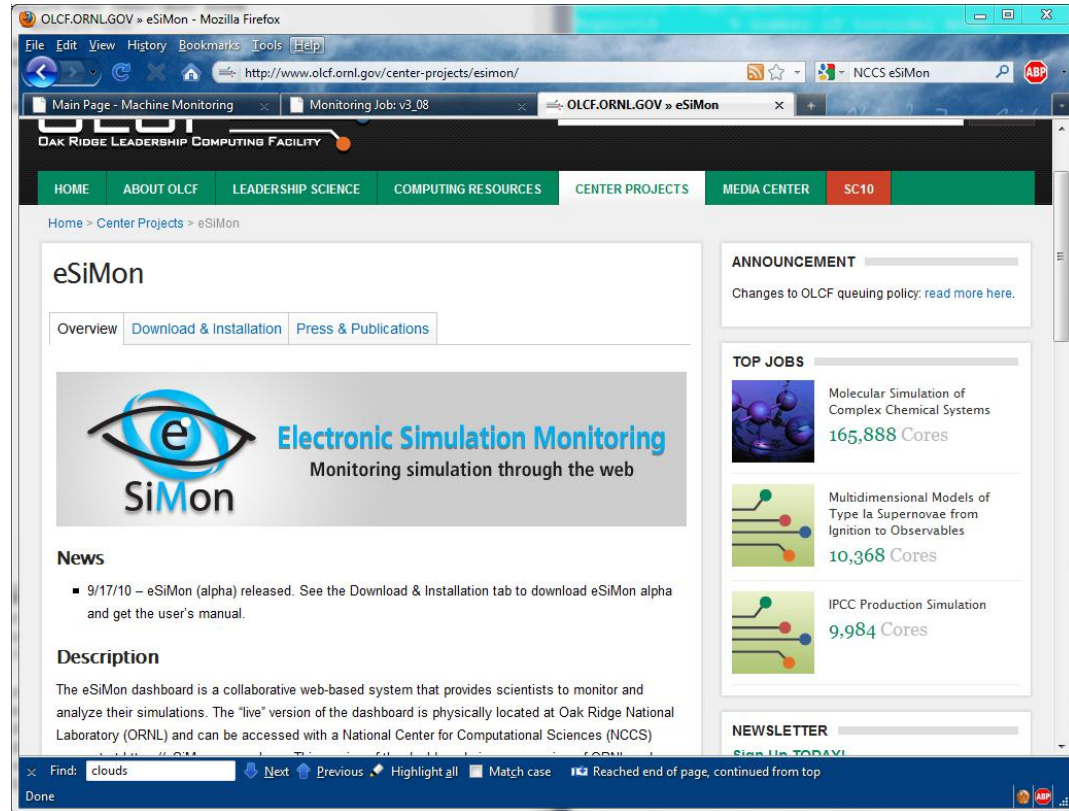
The screenshot displays a web-based monitoring portal for a simulation job. The interface includes several key components:

- Tree view of variables:** A sidebar on the left lists various simulation variables such as `dPhi2`, `ion_angular_flow(1d:favg)`, and `ion_density(1d:omid)`.
- Analysis using R:** A central panel shows a plot of `ion_density vs. psi` with a red line and a blue line. A green circle highlights a specific data point, with the text "Annotate movies" next to it.
- Vector Graphics:** A plot on the right shows a vector field with a yellow and green color scale.
- Calculator:** A numeric keypad is located at the bottom left of the interface.
- Notes:** A text area at the bottom center contains the text: "Taking electronic notes on XGC1 shot rst 24. This is Norbert's shot. It tests restart shot. It has a new color map for most 2D images. I am building a screen shot for a conference talk on May 18th in Baltimore."
- Download data:** A dialog box is open in the foreground, titled "Download". It contains a form for "User information" (UserID: rbarrett, Email Address: barreto@ornl.gov, Phone Number: 8652417552) and options for "Options" (All Variables, All Time Steps, Single Dataset). The total data size is listed as 6.66MiB.

R. Tchoua, S. Klasky, N. Podhorski, P. Mouallem, M. Vouk, "Collaboration Portal for Petascale Simulations," 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009)

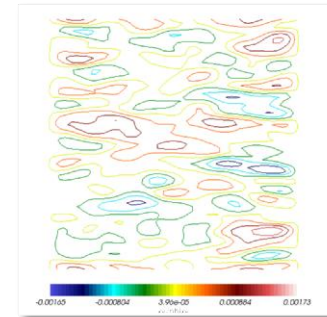
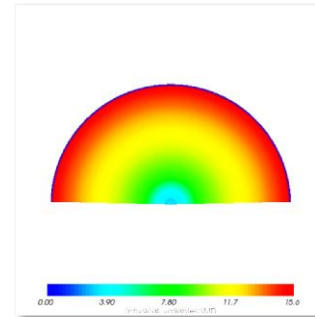
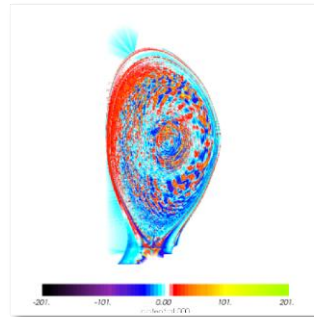
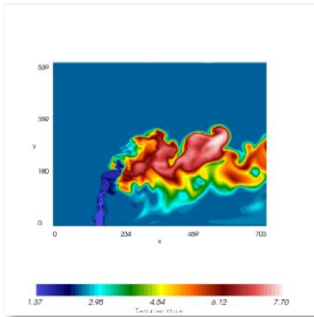
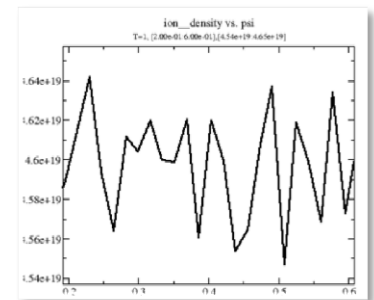
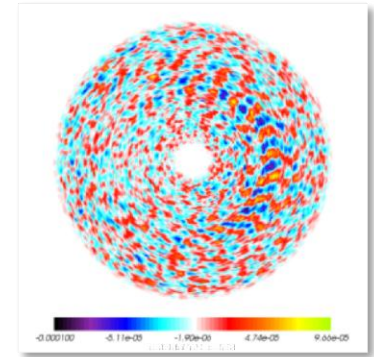
# eSiMon dashboard

- Allow for secure logins with OTP
- Allow for DOE machine monitoring
- Search old jobs
- See collaborators' jobs
- Allow for built-in and plugin analysis
- Share simulation output and analysis results
- Share analysis routines
- Allow for job submission
- Allow for killing jobs

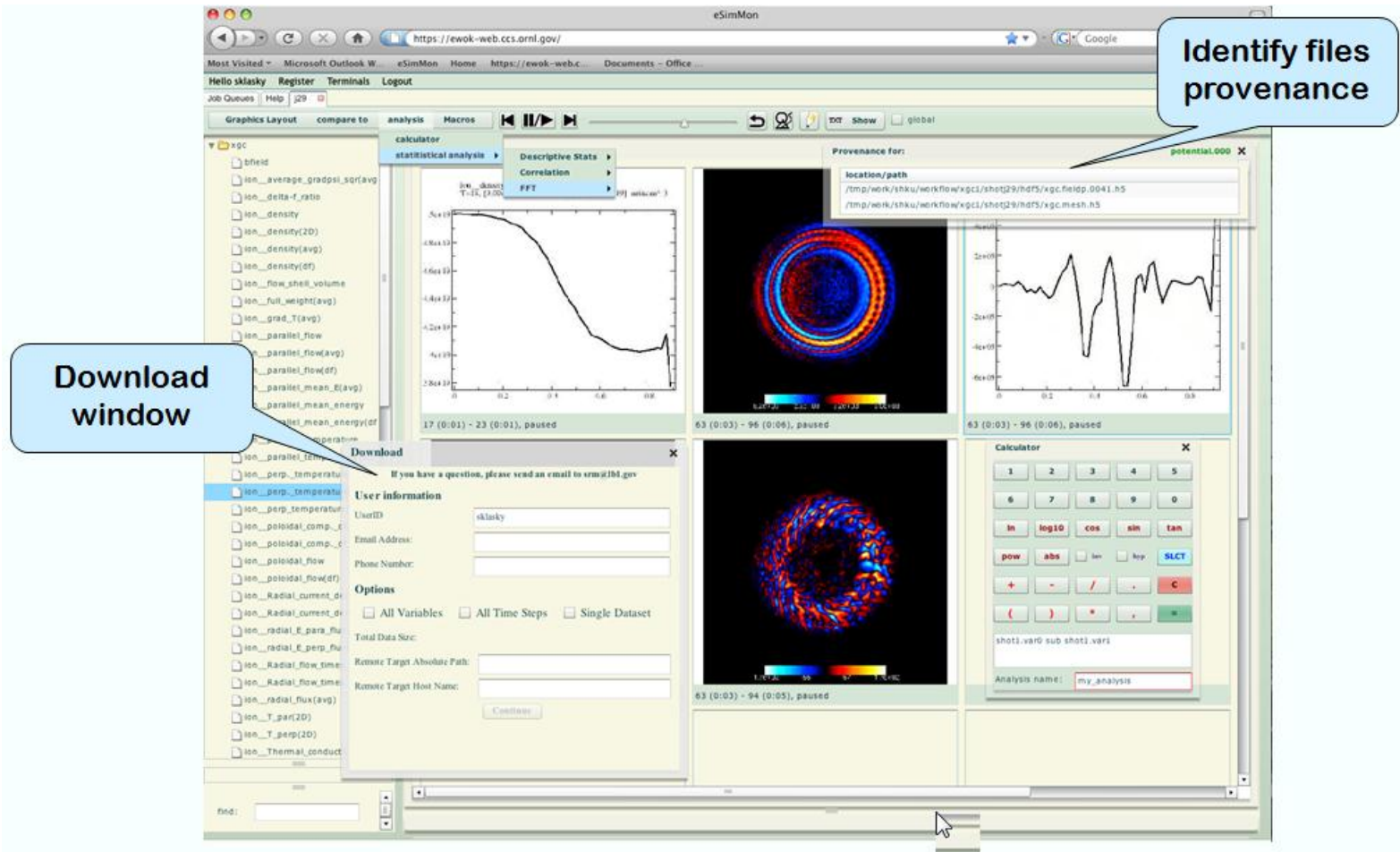


# Need better viz tasks for the workflow for eSiMon dashboard: the plotter

- Goal: Solve all viz tasks of our workflows with one tool
  - This is where we spent most of the time when developing new workflows
- A recipe collection
- A uniform visualization schema
- Reads ADIOS BP/NetCDF/HDF5 arrays
  - Any slice from any multidimensional array
- Use xmgrace to make x-y plots
- x and y array can come from different files
- Loop over a dimension to make many plots at once
- regexp for plotting many variables at once
- Additional “t” array to index the time loop for XGC restart support



# If scientists really need to move the data, use DataMover lite



**Preliminary test showed high transfer rates (93 MB/s), provided source, network, and target have transfer capacity**

# **This brings about the service-oriented architecture**

- **Workflow engine**
  - Orchestrates the execution of the components within a simulation as well as end-to-end application workflows
- **Monitoring services**
  - Dynamic real-time monitoring of large-scale simulations (during execution) is critical
- **I/O services**
  - Efficient, adaptable I/O support is critical
    - Such an I/O service should be componentized so that it allows codes to switch between and tune different methods easily
    - With such an approach, a user may easily switch between file-based I/O to memory-based I/O or switch formats without changing any code, all the while maintaining the same data model when switching I/O components
- **Code-coupling services**
  - Include codes that (1) can run on the same or different machines, (2) are tightly coupled (memory-to-memory), and (3) are loosely coupled (exchange of data through files)

# SOA-2

- **Automatic online (in-transit) data processing services**
- **Automated collection of provenance**
  - **Data provenance**
  - **System provenance**
  - **Workflow provenance**
  - **Performance provenance**
- **Interfaces and portals**
  - **Powerful, but simple to use, user interfaces (e.g., dashboards) provide critical access to the simulation process and the data products for understanding and exploration, as well as management and control**

# Contact

## Scott A. Klasky

Lead, End-to-End Solutions  
Scientific Computing  
National Center for Computational Sciences  
(865) 241-9980  
klasky@ornl.gov