End-to-End Computing at ORNL

Presented by

Scott A. Klasky

Scientific Computing

National Center for Computational Sciences

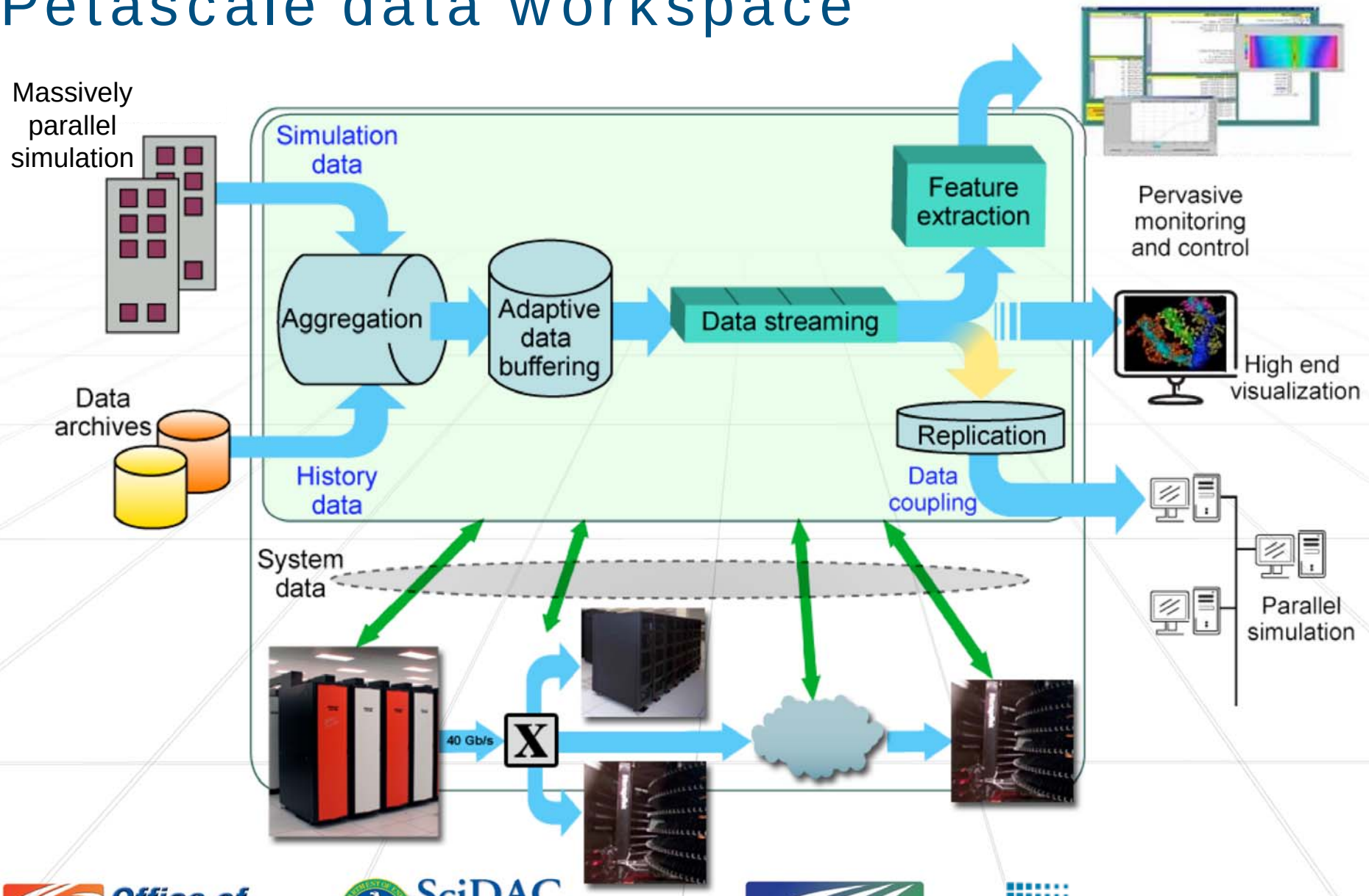
In collaboration with

Caltech: J. Cummings
Georgia Tech: K. Schwan, M. Wolf, H. Abbasi, G. Lofstead
LBNL: A. Shoshani
NCSU: M. Vouk, J. Ligon , P. Mouallem, M. Nagappan
ORNL: R. Barreto, C. Jin, S. Hodson
PPPL: S. Ethier
Rutgers: M. Parashar, V. Bhat, C. Docan
Utah: S. Parker, A. Kahn
UC Davis: N. Podhorszki
UTK: M. Beck, S. Sellers, Y. Ding
Vanderbilt: M. DeVries

SC07

OAK RIDGE National Laboratory

Petascale data workspace



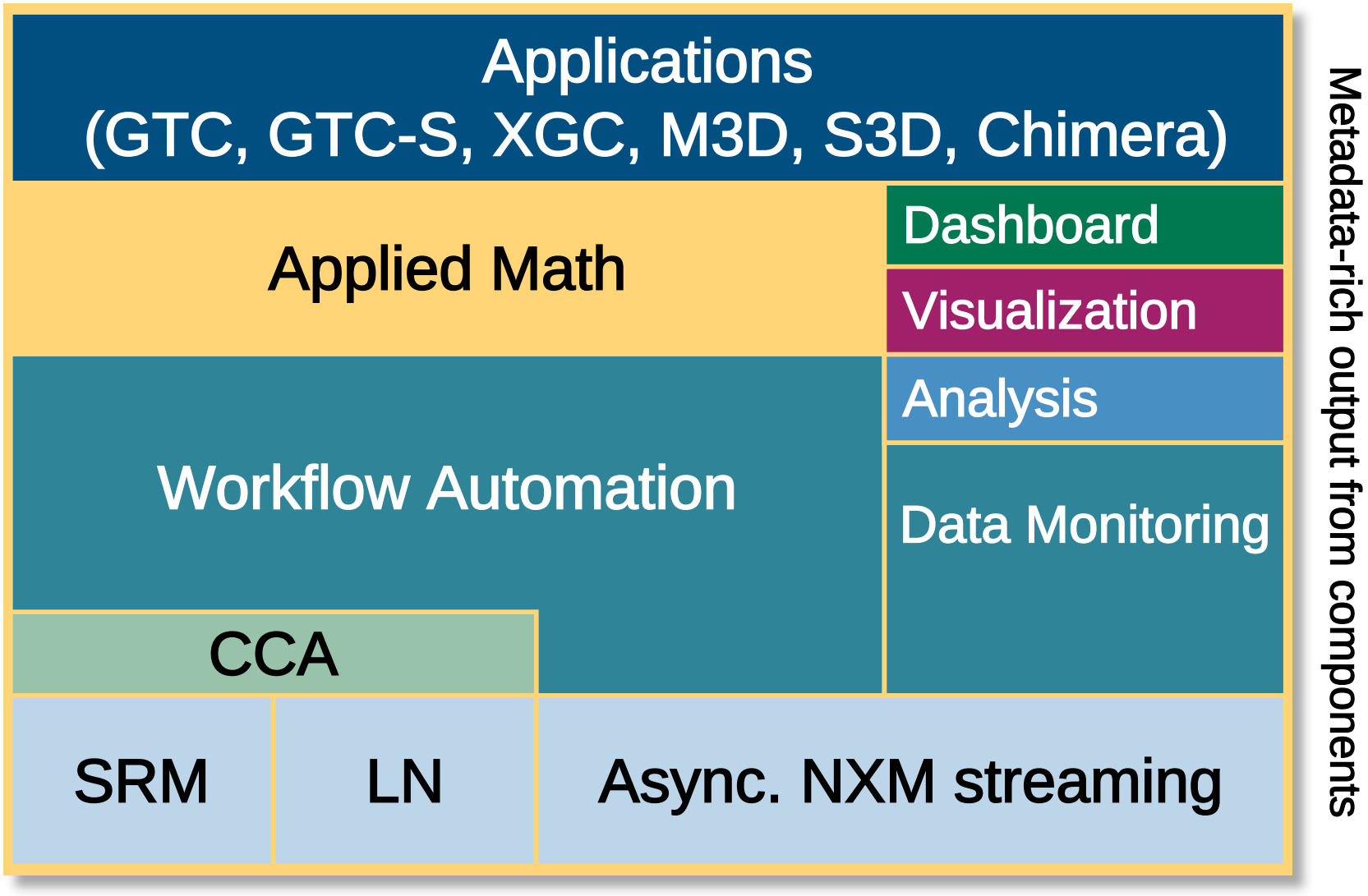
Office of Science
U.S. DEPARTMENT OF ENERGY

SciDAC
Scientific Discovery through Advanced Computing

SDM CENTER

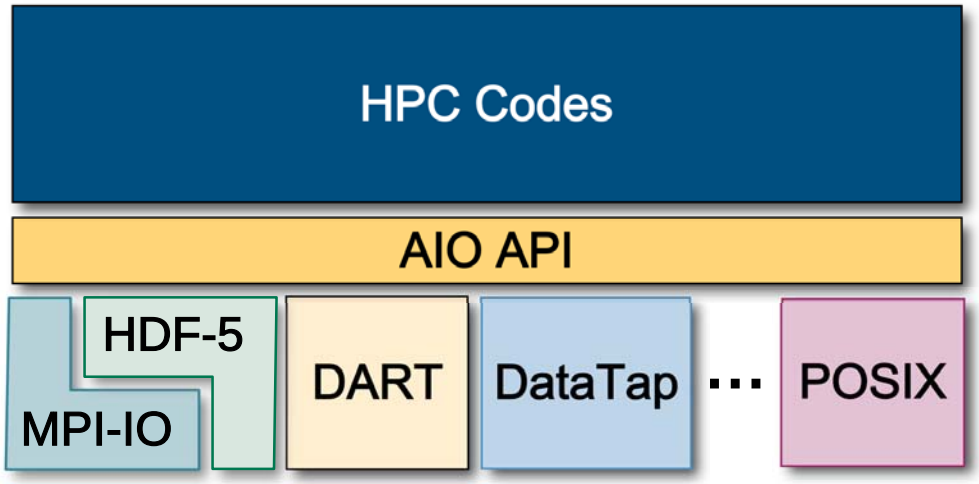
OAK RIDGE
National Laboratory

The End-to-End framework



Unified APIs for MPI/AIO (Lofstead)

- Single, simplified API capable of supporting various low-level implementations (MPI-IO, HDF5, POSIX, asynchronous methods)
- Transmits buffered data only during non-communication phases of HPC codes
- External XML configuration file describing data formats and the storage approach and parameters for each



- Implements best practices for underlying implementations
- Adds data tagging and annotation
- Enables complex inline processing with DataTap and DART (off compute node)
  - e.g., custom compression, filtering, transformation, multiple output organizations from single write, real-time analysis

Asynchronous I/O API usage example

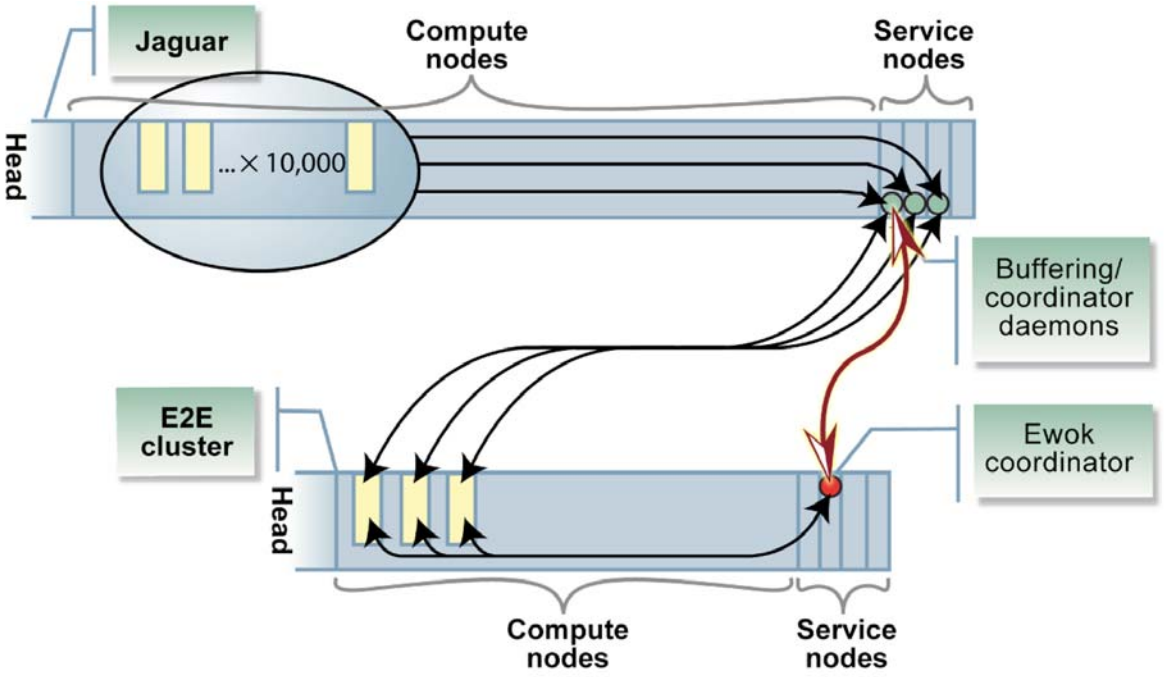
XML configuration file:

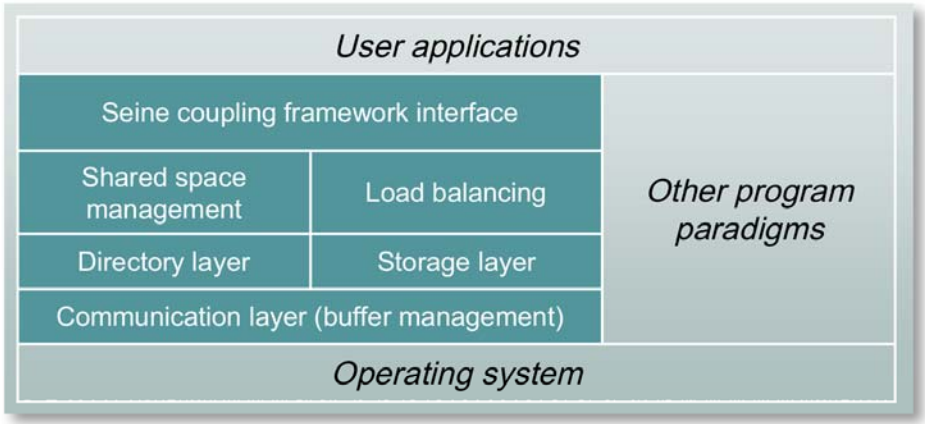
```
<ioconfig>
<datatype name="restart">
<scalar name="mi" path="/param" type="integer"/>
… <!-- declare more data elements -->
<dataset name="zion" type="real"
    dimensions="nparam,4,mi"/>
<data-attribute name="units" path="/param"
    value="m/s"/>
</datatype>
… <!-- declare additional datatypes -->
<method priority="1" method="MPI"
    iterations="100" type="restart"/>
<method priority="2" method="PBIO" iterations="1"
    type="diagnosis">srv=ewok001.ccs.ornl.gov
    </method>
<!-- add more methods for other datatypes -->
</ioconfig>
```

Fortran90 code:

```
! initialize the system loading the configuration file
aio_init (100) ! 100 MB of buffer
! retrieve a declared type for writing
aio_get_type (t1, "restart")
! open a write path for that type
aio_open (h1, t1, "restart.n1")
! write the data items
aio_write (h1, "mi", mi)
aio_write (h1, "zion", zion)
… ! write more variables
! commit the writes for asynchronous transmission
aio_close (h1)
… ! do more work
! shutdown the system at the end of my run
aio_finalize ()
```

Asynchronous petascale I/O for data in transit

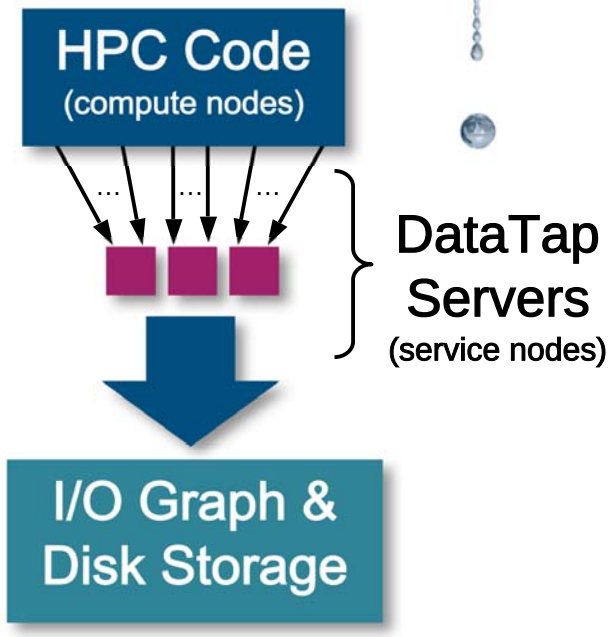




- High-performance I/O
  - Asynchronous
  - Managed buffers
  - Respect firewall constraints
- Enable dynamic control with flexible MxN operations
  - Transform using shared-space framework (Seine)

Lightweight data extraction and processing using a DataTap and I/O Graph

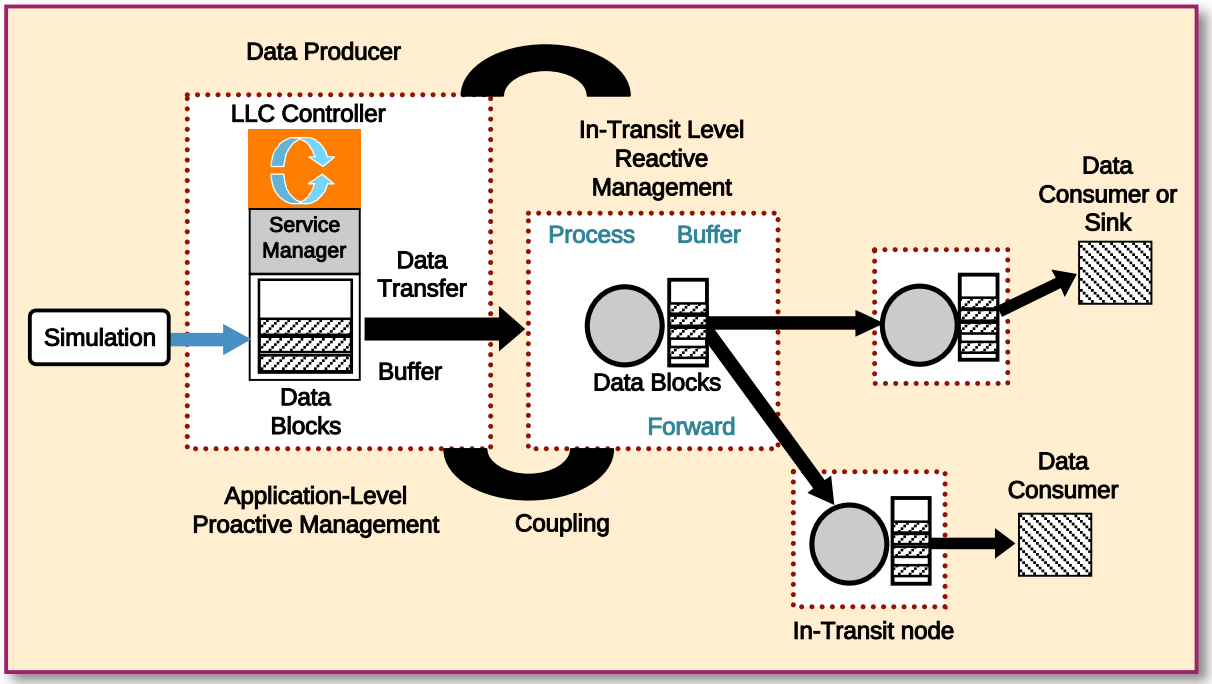
- Adding a DataTap to an HPC code reduces I/O overhead tremendously.
- Rather than writing directly, the client HPC code notifies the DataTap server to read data asynchronously when resources are available.
- The DataTap server scheduler manages data transfer to reduce I/O impact:
  - Guarantees available memory and egress bandwidth consumption does not exceed a user specified limit. Other considerations, such as CPU usage, are also possible.



- The DataTap server is the gateway to I/O graph processing for storage to disk or additional processing--even on another cluster.

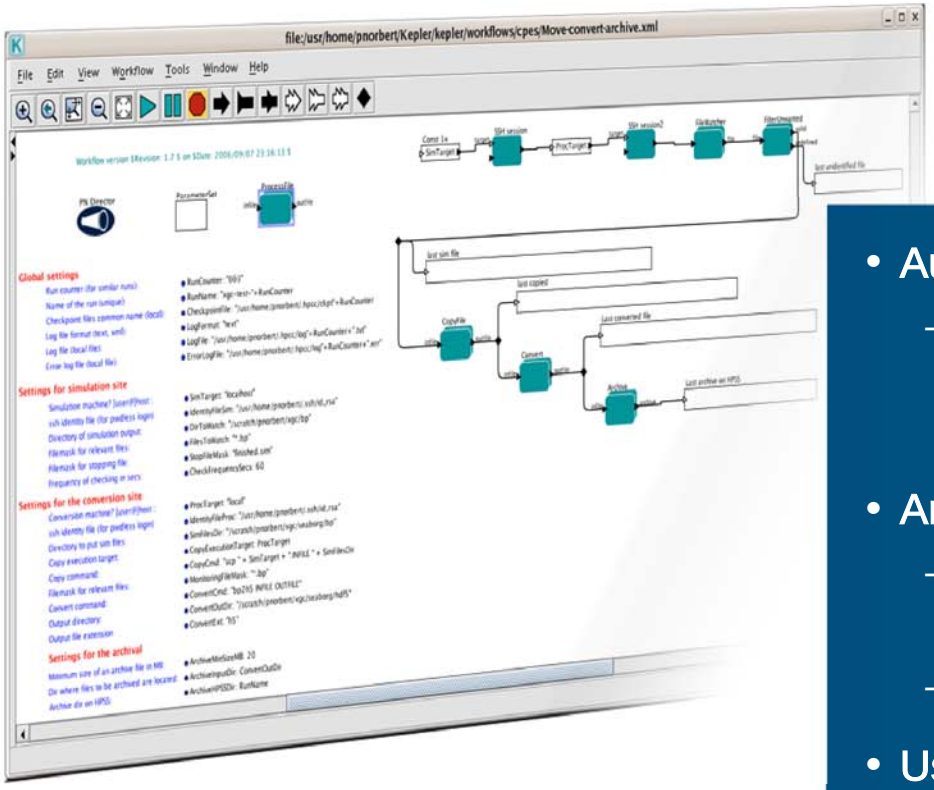
Data streaming and in-transit processing

- Requirements
  - High-throughput, low-latency transport with minimized overheads
  - Adapt to application and network state
  - Schedule and manage in-transit processing
- Approach – Cooperative self-management
  - Application-level data streaming
    - Proactive management using online control and policies
  - In-transit data manipulation
    - Quick, opportunistic, and reactive



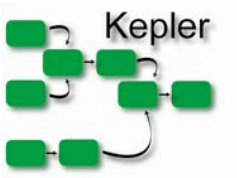
- Experimental evaluation
  - ORNL and NERSC -> Rutgers -> PPPL
  - Adaptive in-transit processing reduced idle time from 40% to 2%.
  - Improved end-to-end data streaming –
    - Reduced data loss.
  - Improved data quality at sink.

Workflow automation

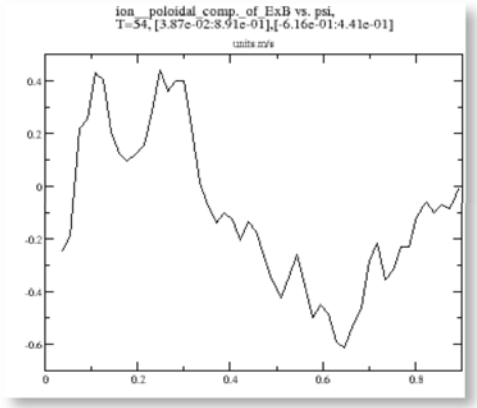


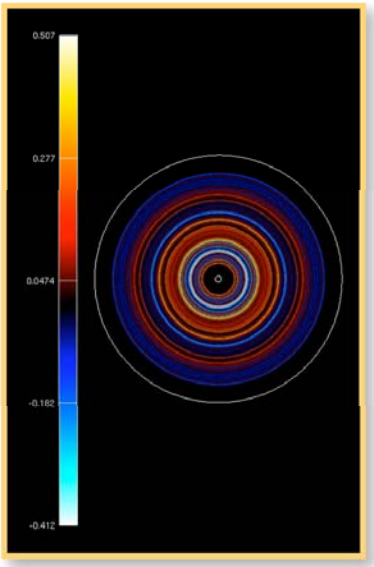
- Automate the data processing pipeline
  - Transfer of simulation output to the e2e system, execution of conversion routines, image creation, data archiving
- And the code coupling pipeline
  - Check linear stability and compute new equilibrium on the e2e system
  - Run crash simulation if needed
- Using the Kepler workflow system
- Requirements for Petascale computing
  - Easy to use
  - Dashboard front-end
  - Autonomic
  - Parallel processing
  - Robustness
  - Configurability

CPES workflow automation



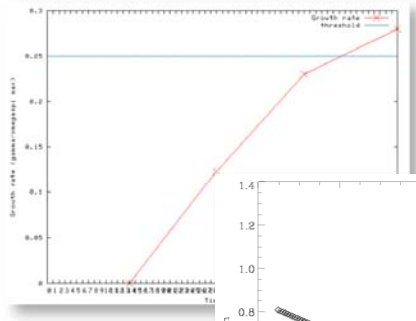
- NetCDF files
  - Transfer files to e2e system on-the-fly
  - Generate images using grace library
  - Archive NetCDF files at the end of simulation

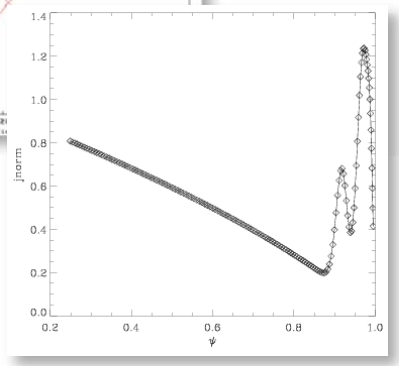




- Proprietary binary files (BP)
  - Transfer to e2e system using *bbcp*
  - Convert to HDF5 format
  - Generate images with AVS/Express (running as service)
  - Archive HDF5 files in large chunks to HPSS

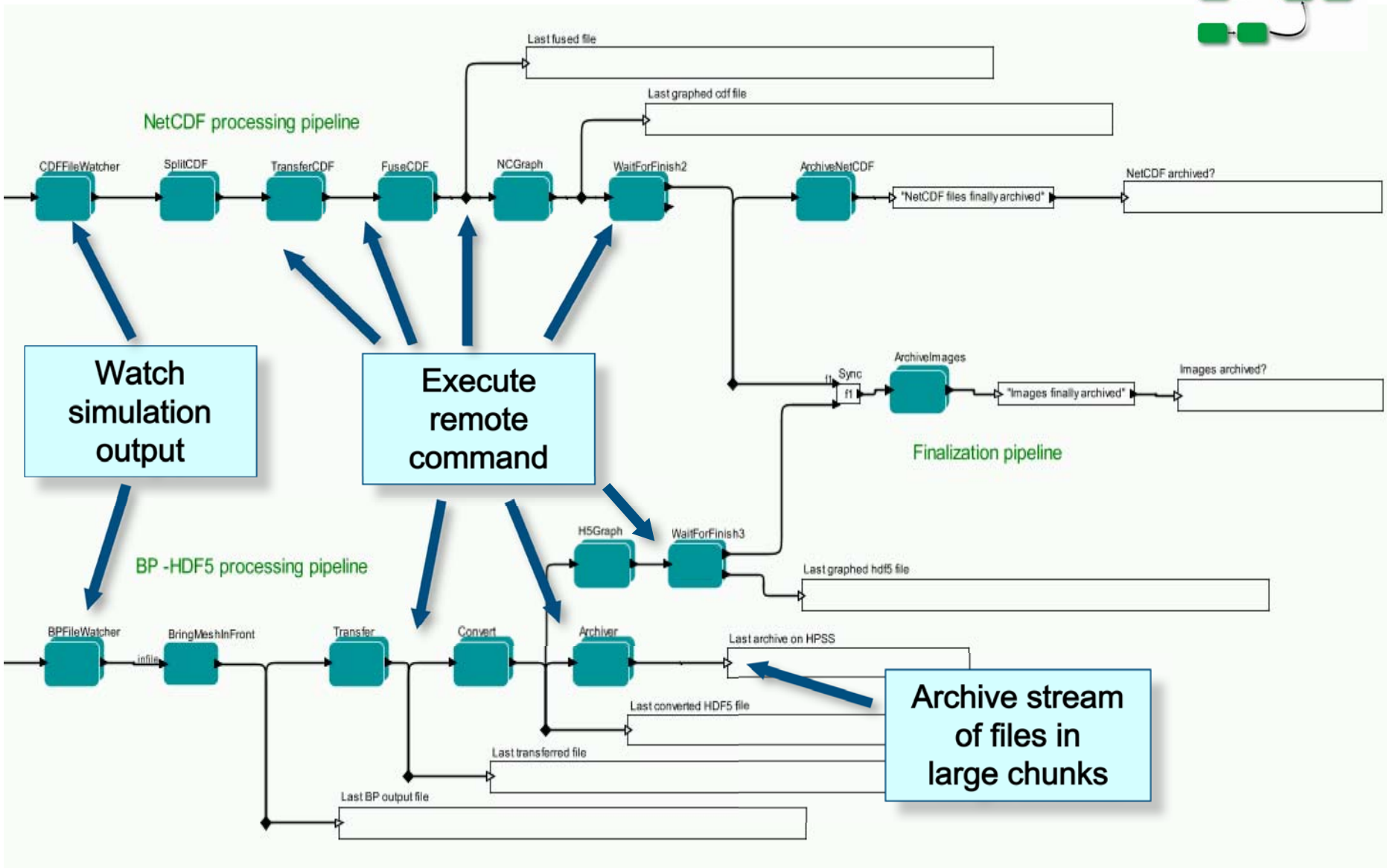
- M3D coupling data
  - Transfer to end-to-end system
  - Execute M3D: compute new equilibrium
  - Transfer back the new equilibrium to XGC
  - Execute ELITE: compute growth rate, test linear stability
  - Execute M3D-MPP: to study unstable states (ELM crash)





OAK RIDGE National Laboratory

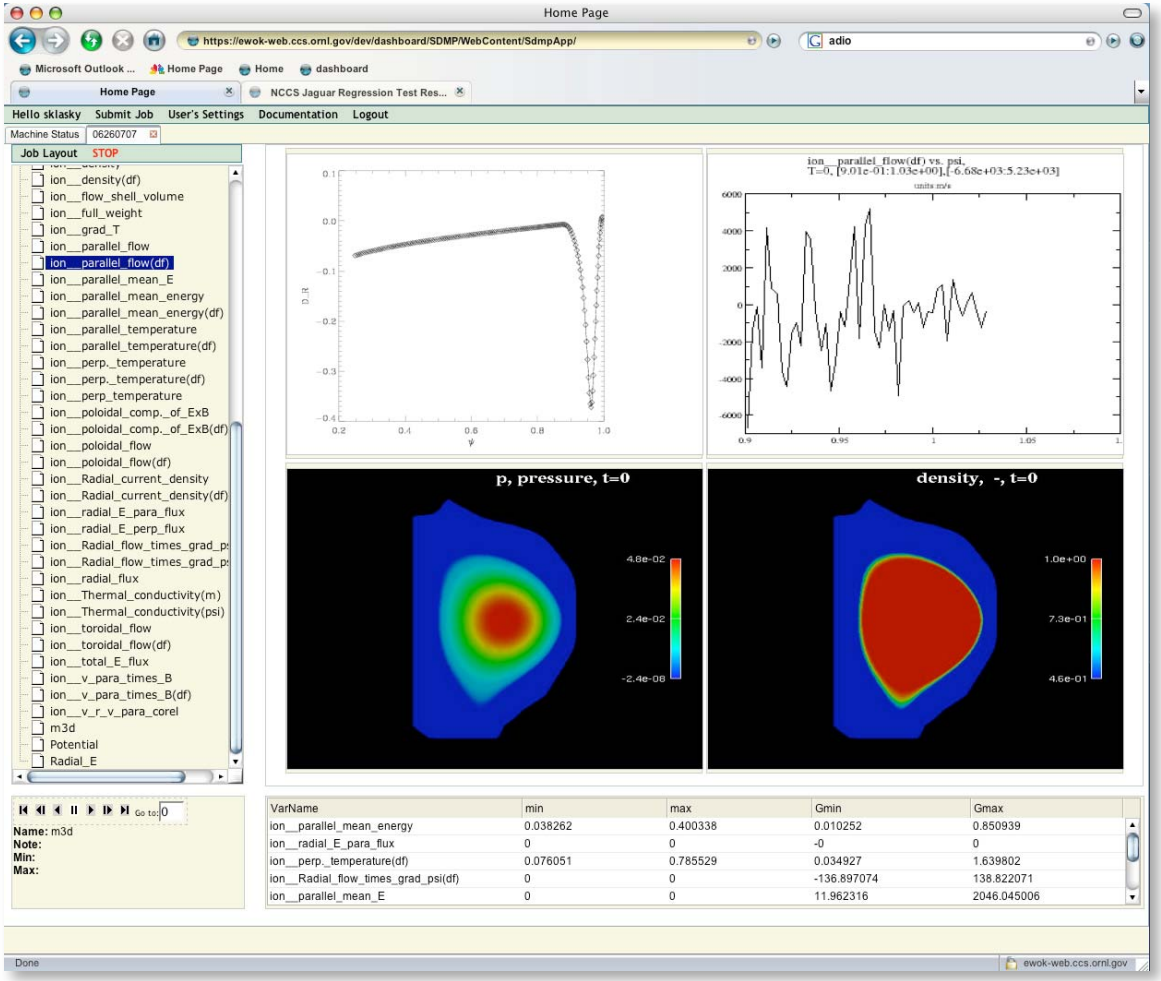
Kepler components for CPES



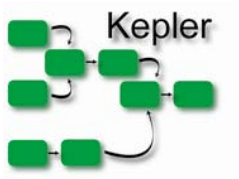
Kepler workflow for CPES code coupling

Combines data from
- AVS/Express,
- Gnuplot,
- IDL,
- Xmgrac.

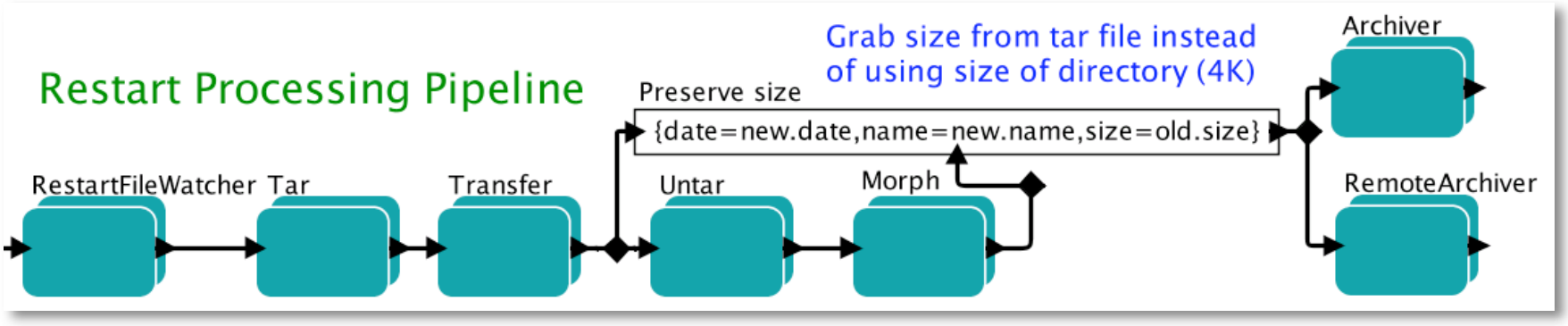
Allows us to monitor the weak code coupling of XGC0 (Jaguar) to M3D-OMP (ewok) to ELITE (ewok) to M3D-MPP (ewok).



S3D workflow automation

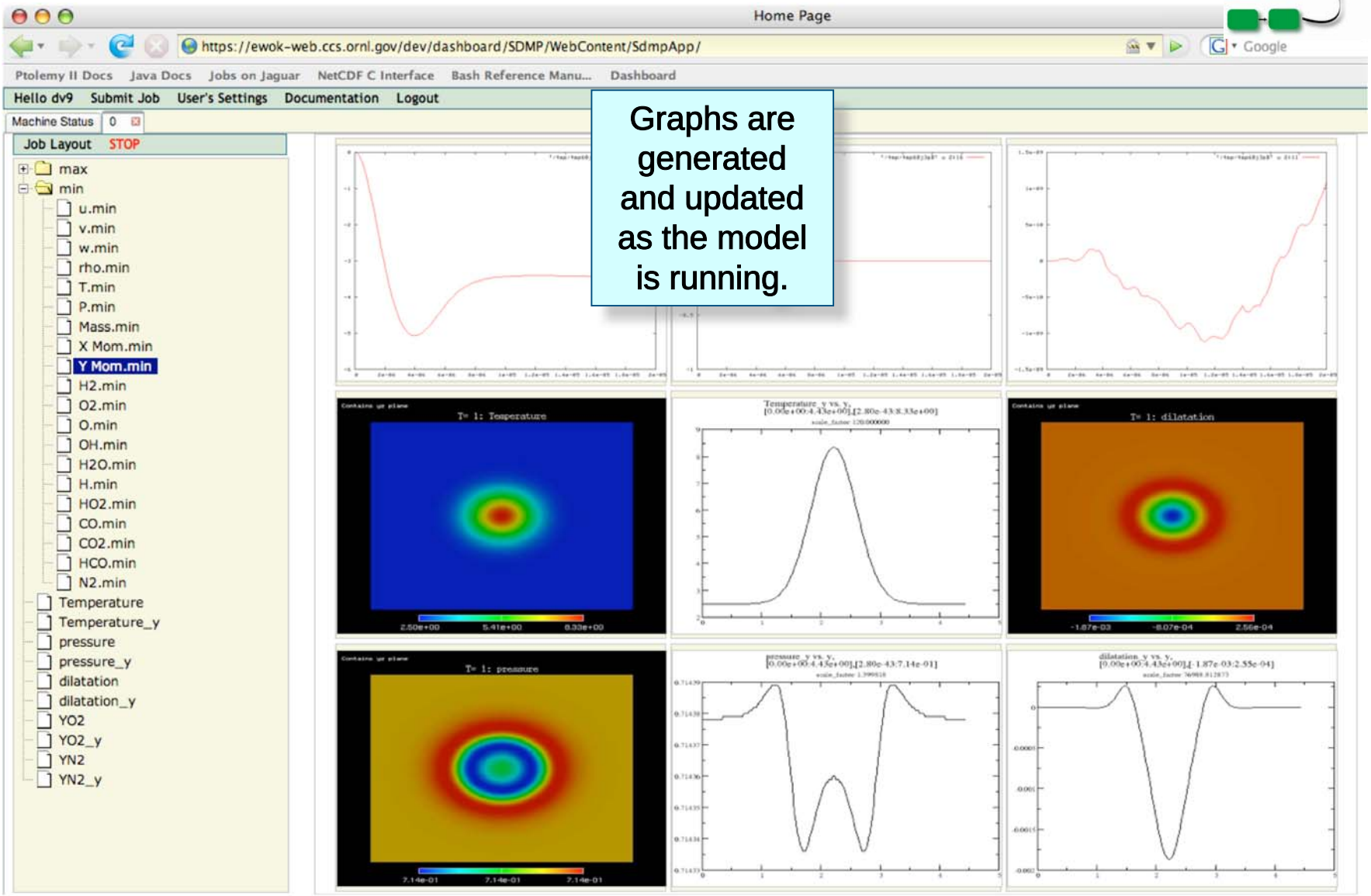


- Restart/analysis files
  - Transfer files to e2e system
  - Morph files using existing utility
  - Archive files to HPSS
  - Transfer files to Sandia
- NetCDF files
  - Transfer files to e2e system on-the-fly
  - Generate images using grace library and AVS/Express
  - Send images to dashboard system
- Min/max log files
  - Transfer to e2e system at short intervals
  - Plot with gnuplot
  - Send to dashboard for real-time monitoring



OAK RIDGE National Laboratory

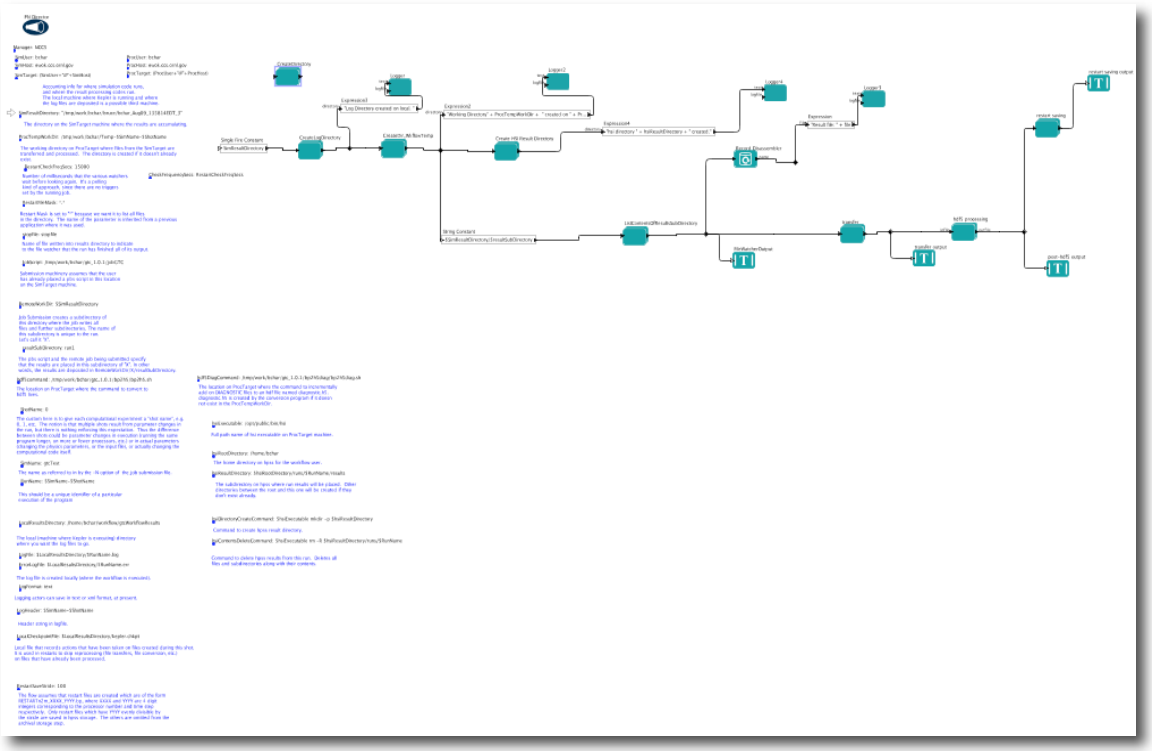
S3D graphs on the dashboard



Graphs are generated and updated as the model is running.

GTC workflow automation

- Proprietary binary files (BP)
  - Convert to HDF5 format
  - Generate images
    - With custom processing programs (bp2h5-png)
    - With connection to VisIt
  - Archive files in HPSS



- Key actor: ProcessFile (N. Podhorszki)
  - Check-perform-record checkpoint pattern
  - Operates on a stream of operands (remote files) in a pipeline of processors

- Logging and error logging of operations provided within the component; just configure for location of log files

Simulation monitoring

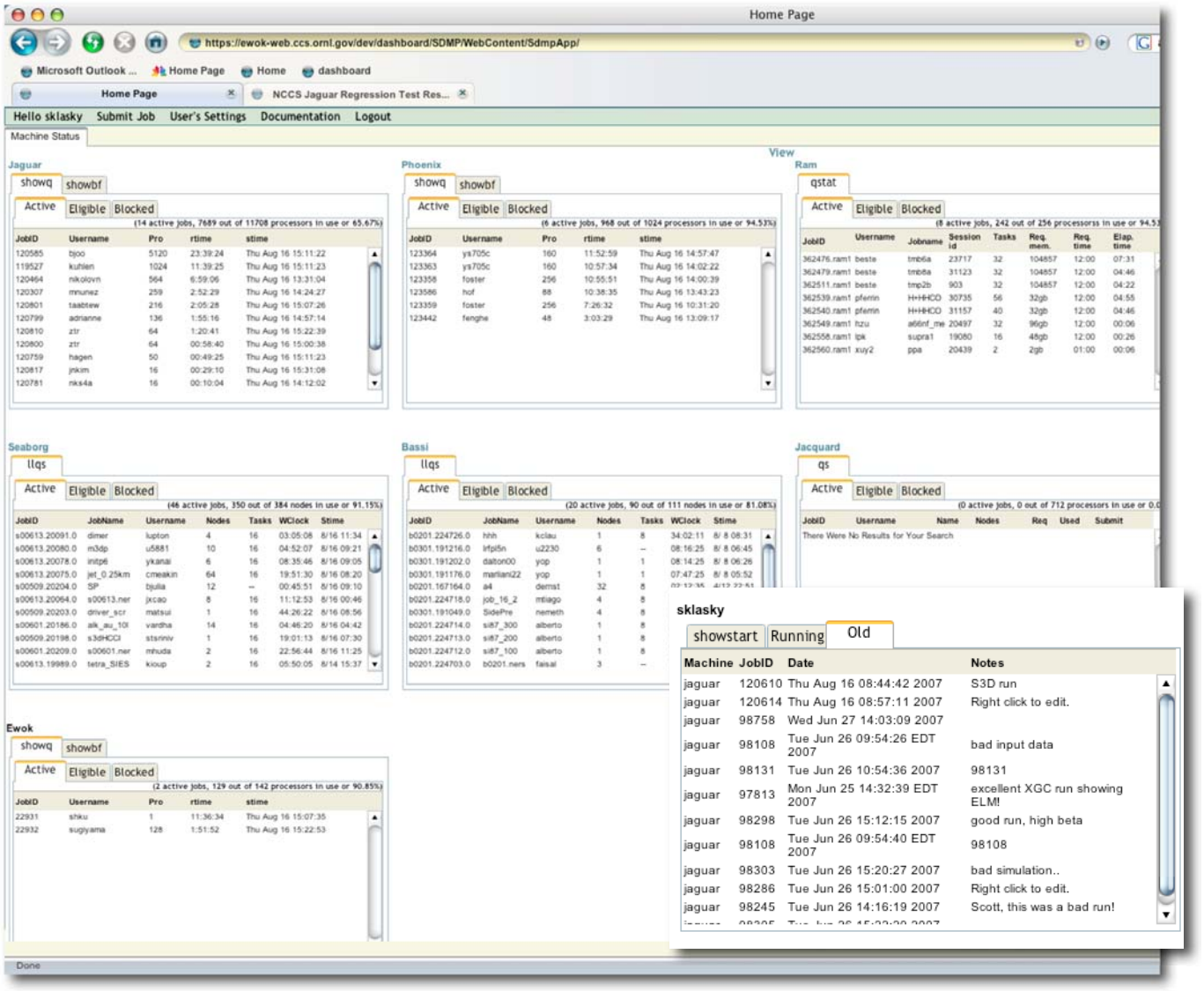
- Simulation monitoring involves the successful integration of several sub-tasks:
  - Monitoring of DOE machines
  - Visualization of simulation data:
    - graphs, movies, provenance data, input files etc.
  - Database integration and High Performance Storage System:
  - Annotating images and runs
    - taking e-notes and maintaining an e-book
  - High-speed data delivery services
  - Workflow system that pieces these tasks together

Scientist with limited knowledge about dashboard technology

- Check machine status and queues
- Submit job through dashboard and workflow
- Visualize simulation data from provenance information to output files and graphs
- Analyze data
- Keep notes on runs
- Download selected information or move to specific storage
- Interact with workflow

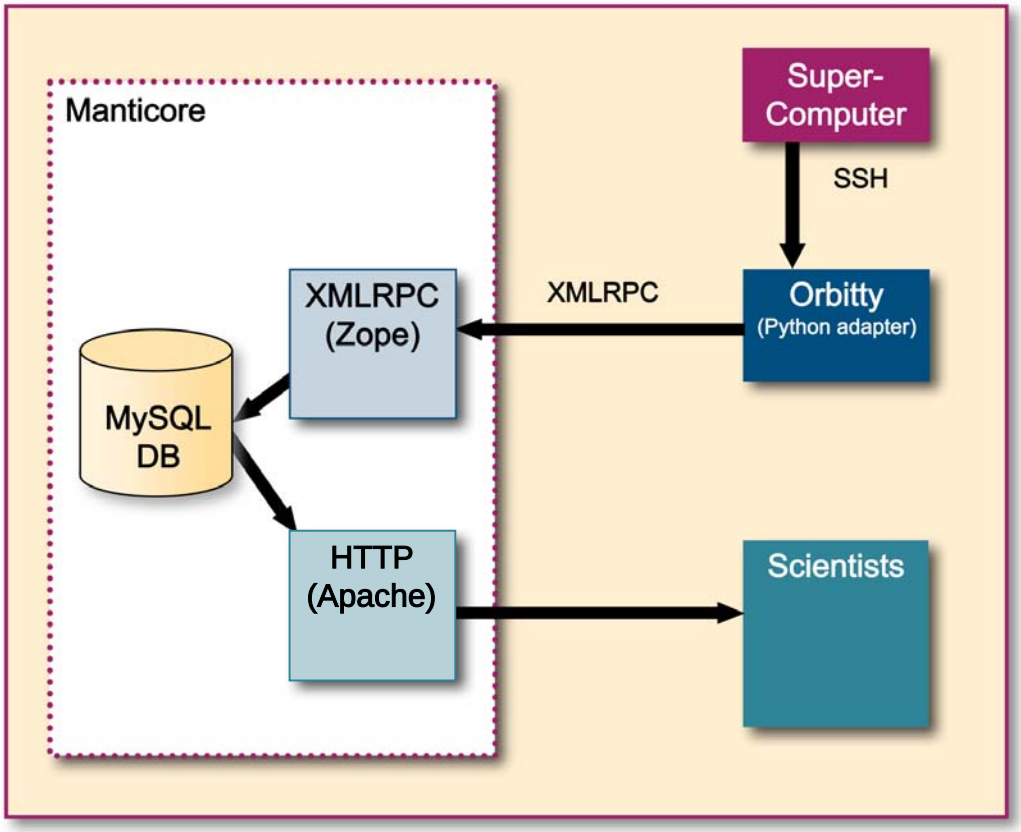
Machine and job monitoring

- Back end: shell scripts, python scripts and PHP
  - Machine queues command
  - Users' personal information
  - Services to display and manipulate data before display
- Dynamic front end:
  - Machine monitoring: standard web technology + Ajax
  - Simulation monitoring: Flash
- Storage: MySQL (queue-info, min-max data, users' notes…)

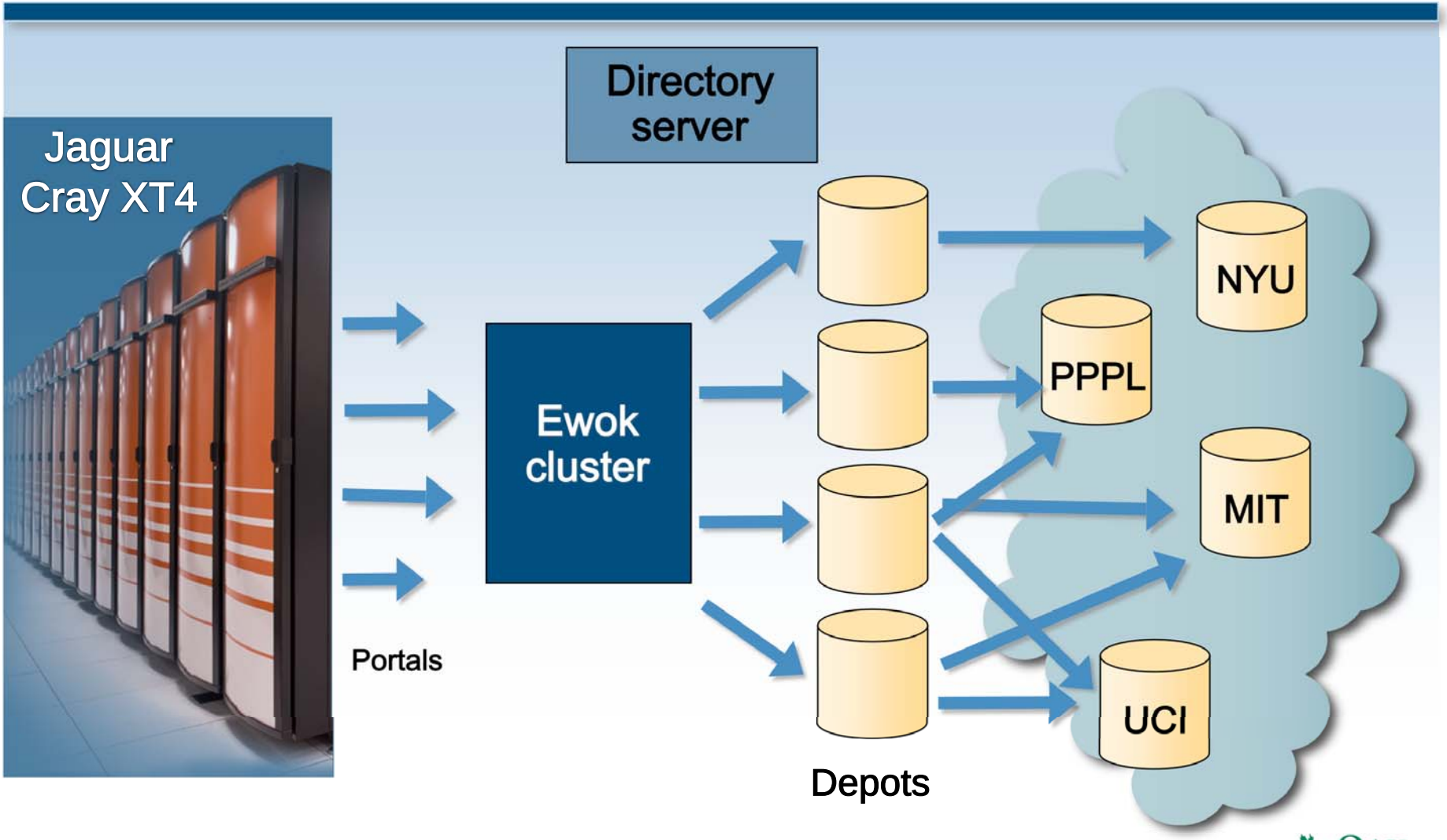


Provenance tracking

- Collects data from the different components of the W/F.
- Provides the scientist easy access to the data collected through a single interface.
- APIs have been created in Kepler to support real-time provenance capture of simulations running on leadership-class machines.



Logistical networking:
High-performance ubiquitous and transparent data access over the WAN



Data distribution via logistical networking and LoDN

- Logistical Distribution Network (LoDN) directory service adapted to run in NCCS environment:
  - User control of automated data mirroring to collaborative sites on per file or (recursive) per folder basis.
  - Firewall constraints require mirroring of metadata to outside server.
- User libraries enable program access to LN storage through standard interfaces (POSIX, HDF5, NetCDF).
- User control over data placement and status monitoring will be integrated with dashboard.
  - Download of data to local system for offline access.

Contact

Scott A. Klasky

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