Access to External Resources Using Service-Node Proxies

CUG 2009

Ron Oldfield

Sandia National Laboratories
Supported by Networks Grand Challenge LDRD

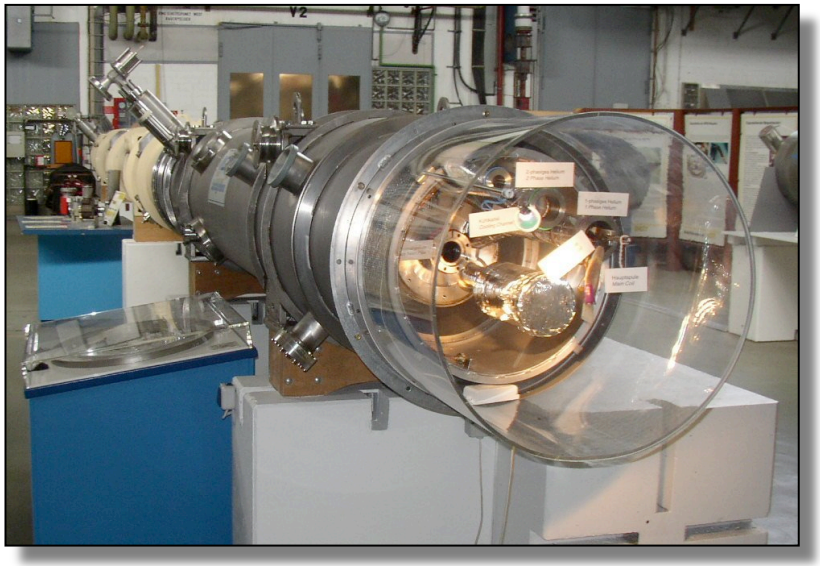
SAND Number: 2009-xxxxP

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

MPP Capability Systems

"MPP systems are the particle accelerators of the computing world", Camp

Red Storm Cabinets at Sandia
(© Sandia Corporation)



Segment of Particle Accelerator from DESY
(Courtesy of Wikipedia)

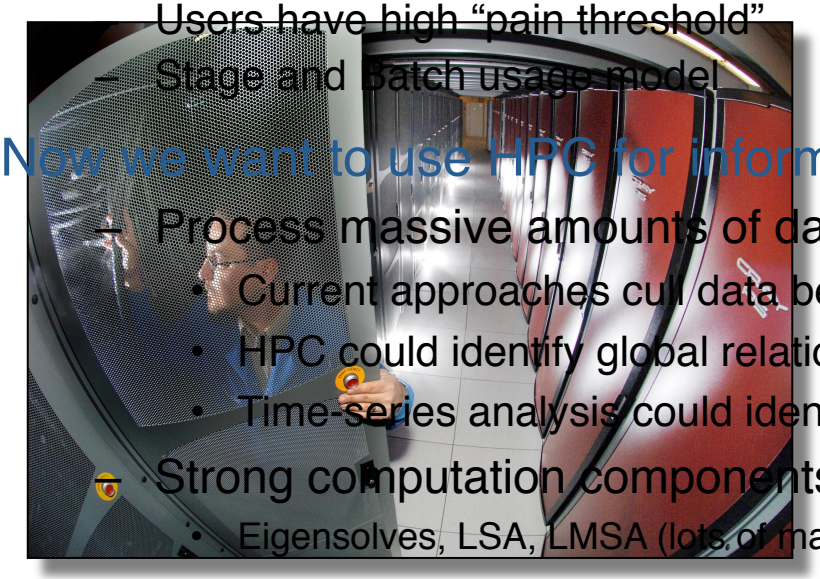
LDRD
LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories

MPP Capability Systems

HPC Systems in a nutshell

- Designed for computational science
- Fast (and specialized) network and FS
- Require specialized training to use
- Users have high "pain threshold"
- Stage and batch usage model

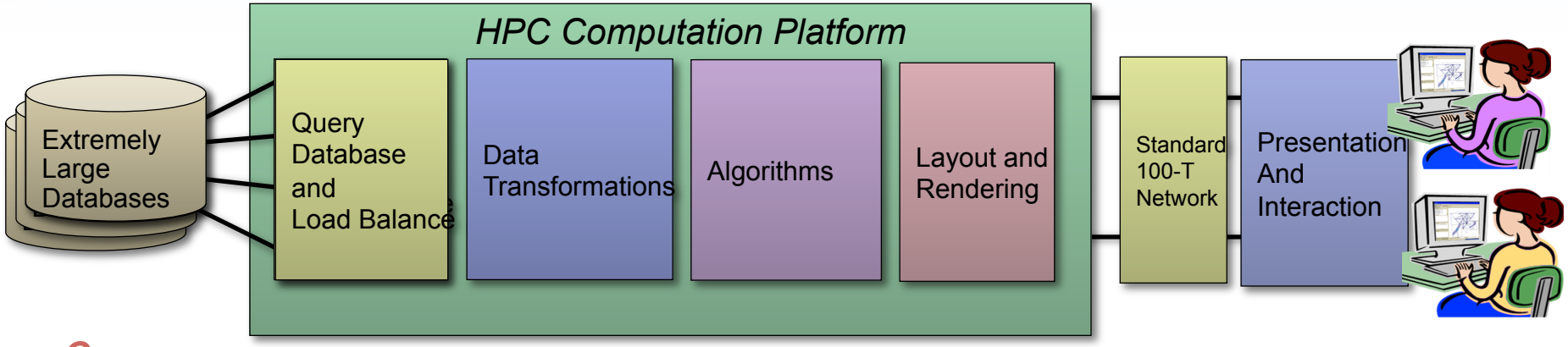


Red Storm Cabinets at Sandia
(© Sandia Corporation)

Now we want to use HPC for informatics

- Process massive amounts of data
  - Current approaches cull data before processing
  - HPC could identify global relationships in data
  - Time-series analysis could identify patterns, but requires lots of data
- Strong computation components
  - Eigensolves, LSA, LMSA (lots of matrix multiplies)
- Strong national security interests

Expanding the Architecture to Support Intelligence Analysis



Query

- Select * from network_packet_table where date > 7/1/2008 and date < 7/8/2008

Data Transformations

- vtkTableToTree
- vtkTableToGraph
- vtkTableToSparseArray
- vtkTableToDenseArray*

Algorithms

- Statistics
- Linear Algebra
- Tensor Methods
- Graph Algorithms

Layout and Rendering

- Tree Layout
- Tree Map
- Graph Layout
- Hierarchical
- Geodesic

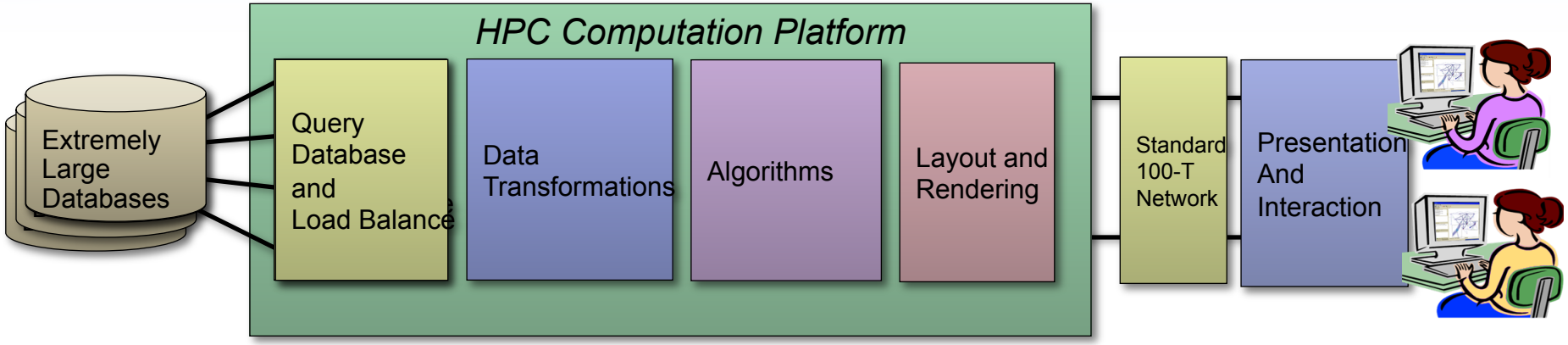
Presentation and Interaction

- Client/Server
- Geometry and Image Delivery
- Cross Platform Qt User Interface
- Linked Selection

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories

Expanding the Architecture to Support Intelligence Analysis



Issues with informatics on Red Storm

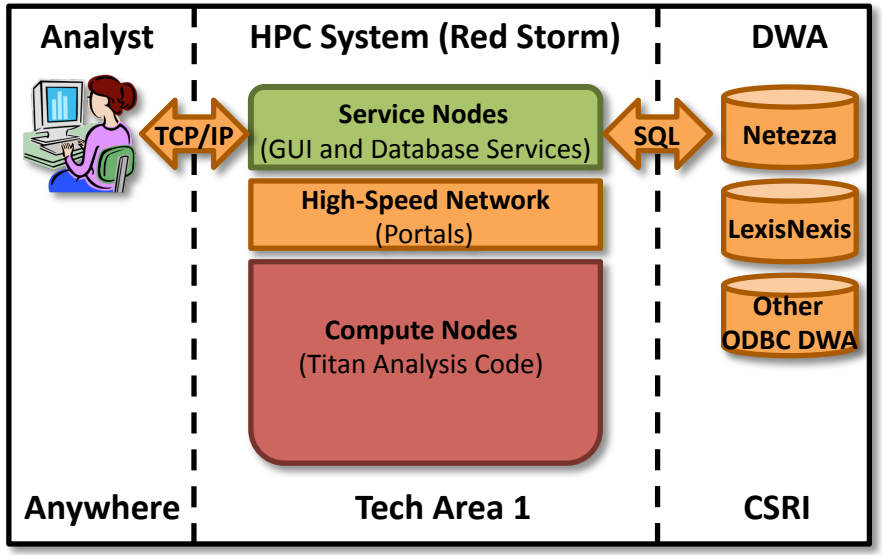
- Specialized network APIs (Portals)
- No database capabilities (i.e., ODBC)
- No interactive visualization

Service-Node Proxy (SQL Service)

Enables Remote Database Access

Features

- Provides "bridge" between parallel apps and external DWA
- Runs on Red Storm network nodes
- Titan apps communicate with services through LWFS RPC (over Portals)
- External resources (Netezza) communicate through standard interfaces (e.g. ODBC over TCP/IP)



Application-level services enable an HPC application to leverage remote resources.

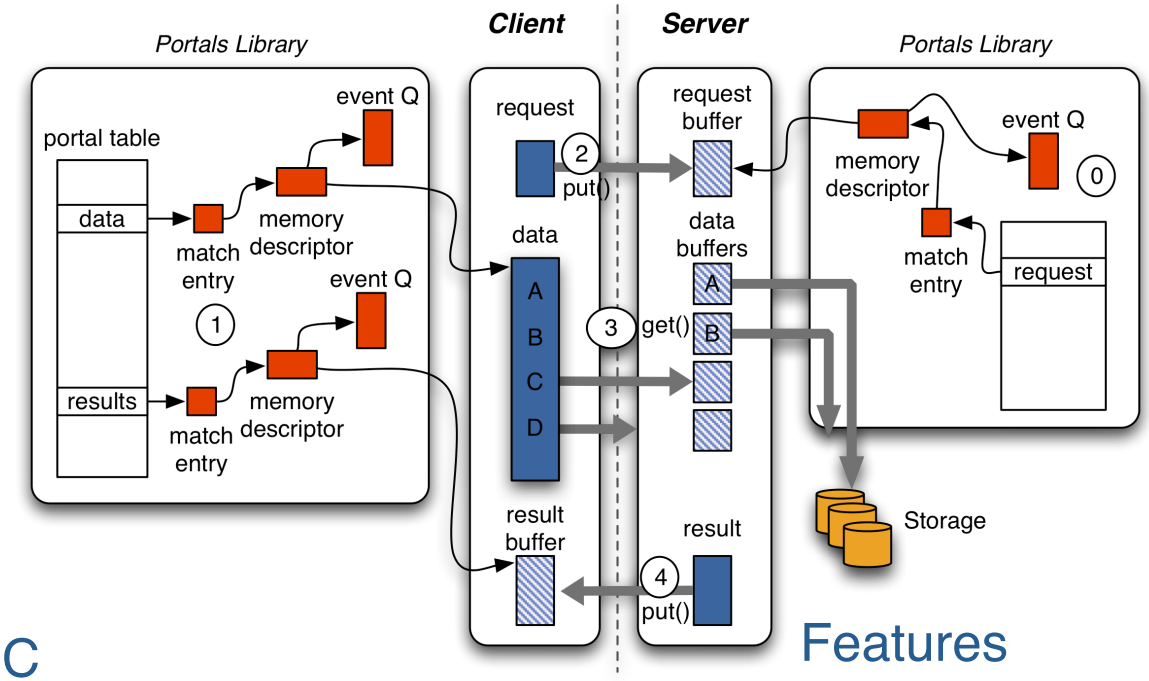
Overview of Talk

- LWFS RPC
- Developing an SQL Service with LWFS RPC
- A parallel statistics demonstration
- Other services in development

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories

LWFS RPC Description



LWFS RPC

- Library for rapid development of services
- Runs on compute or service nodes (catamount and Linux)
- Portals and InfiniBand implementations

Features

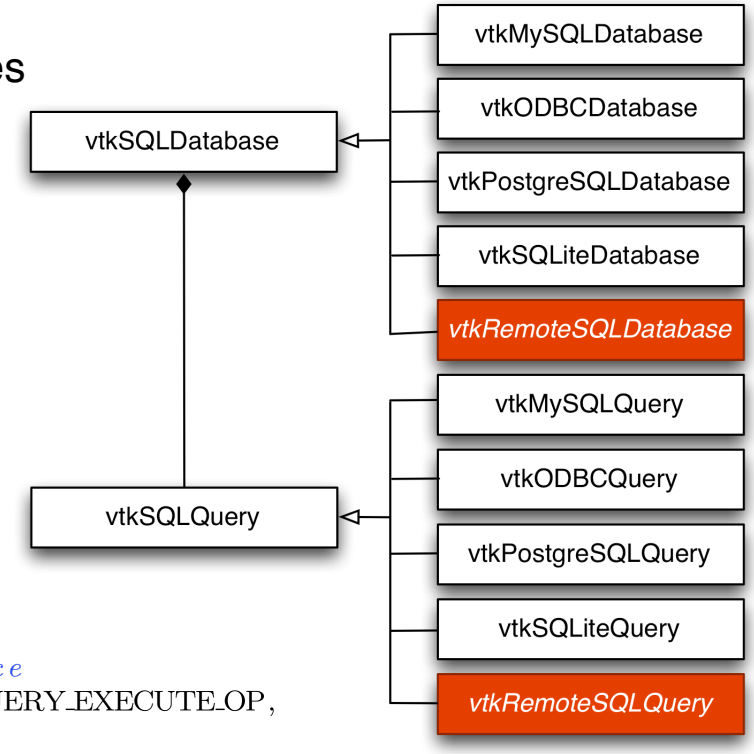
- Asynchronous API
- XDR for encoding reqs
- Server-directed movement
- Separate control/data channels

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories

Developing an SQL Service

- Compute-Node Client
  - Extensions of vtkSQL{Database,Query) classes
  - Package args (id, string) for remote func.




```
bool vtkRemoteSQLQuery::Execute()
{
    // handle to the async RPC request
    lwfs_request req;

    // XDR data structures
    vtk_sql_query_execute_args args;
    vtk_sql_query_execute_res res;

    // Set args for remote Execute function.
    args.qid = this->GetRemoteQueryID();
    args.qstr = this->GetQuery();

    // Marshal and send the request to the SQL Service
    lwfs_call_rpc(this->GetRemoteService(),VTK_SQL_QUERY_EXECUTE_OP,
        &args,NULL,0,&res,&req);

    // Wait for async request to complete
    lwfs_wait(&req,LWFS_INFINITY);

    return res.status;
```

Developing an SQL Service

SQL Server: Executes on Red Storm Network Node

```
int vtk_sql_query_execute_stub(
        const lwfs_remote_pid *caller,
        const vtk_sql_query_execute_args *args,
        const lwfs_rma *data_addr,    // not used
        const lwfs_rma *res_addr)
{
    // A data structure for the result
    vtk_sql_query_execute_res res;

    // Lookup the partner query object (stored in an STL map)
    query = query_map[args->qid];

    // Execute the query
    if (query) {
        query->SetQuery(args->qstr);
        status = query->Execute();
        res.status = status;
    }

    // Send the result back to client
    return lwfs_send_result(VTK_SQL_QUERY_EXECUTE_OP,
        rc, &res, res_addr);
}
```

Case Study: Parallel Statistics

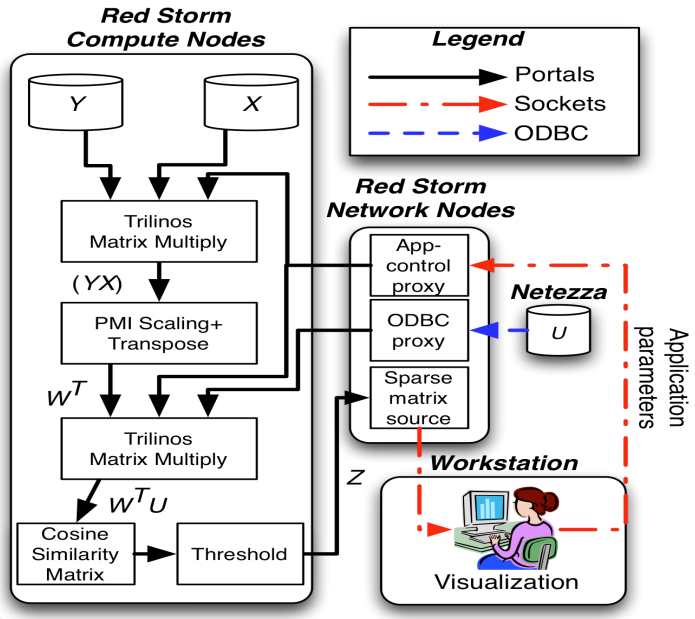
- Implemented Parallel Statistics Code as Demo
  - Pull one or more data sets from Netezza using SQL Service
  - Use MPI to distribute rows of query results evenly to compute nodes
  - Compute mean, variance, skewness, kurtosis, covariance, Cholesky decompositon
  - Insert results in a new table on remote Netezza using SQL Service
- Demonstration of functionality, not performance
  - Implemented minimal set of methods to demonstrate functionality
- Performance issues
  - Limitations of API (small requests)
  - ODBC implementation
  - Netezza limited to one head node (1 GigE/s max)

Other Remote Services on Red Storm

Future Work

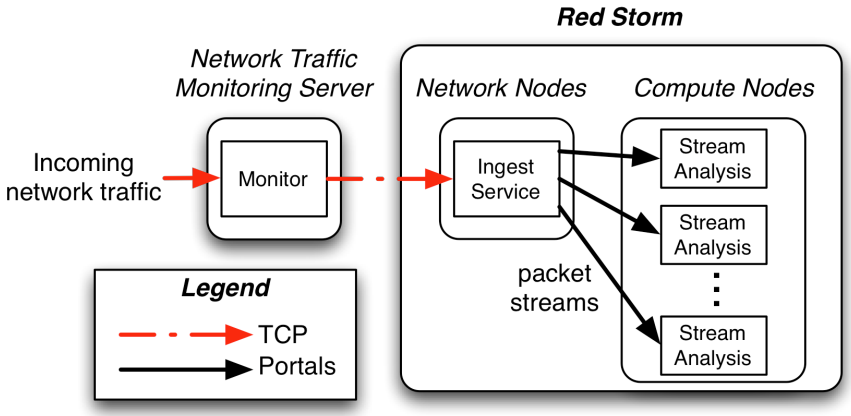
Multi-lingual text analysis

- Data sizes (matrices of 1.6Mx333K)
- Leverage existing technologies for HPC numerics (Trilinos, LMSA) and viz (Titan)
- Services for viz, app-control, database



Real-time analysis of network traffic

- Service to ingest and distribute network traffic (TCP packets) for analysis on compute nodes



Summary

- Exploring viability of Red Storm for informatics
- Informatics applications
  - Exploit numeric capabilities of HPC systems,
  - Require new functionalities to match analyst needs
    - Access to data-warehouse appliances (Netezza, LexisNexis, other remote databases)
    - Interactive requirements for visualization and app control
- LWFS RPC for application services
  - SQL Proxy, Viz proxy, others (see extra slides)
- Statistics demonstration shows functionality
  - Still need to address performance issues

LDRD
LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories

Extra Slides

Compute-Node Services

- NetCDF I/O cache
- CTH Fragment Detection

Compute-Node Services

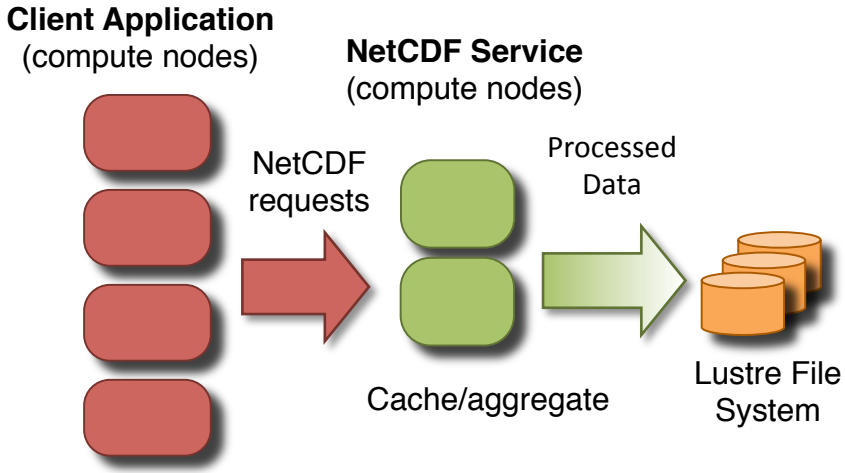
NetCDF I/O Cache

Motivation

- Synchronous I/O libraries require app to wait until data is on storage device
- Not enough cache on compute nodes to handle I/O bursts
- NetCDF is basis of important I/O libs at Sandia (Exodus)

NetCDF Service

- Service aggregates/caches data and pushes data to storage
- Async I/O allows overlap of I/O and computation



NetCDF service provides caching in available compute nodes to improve effective I/O performance.

Results and Status

- 150 GB/s effective writes (10x improvement)
- Plan to integrate with Exodus II (for real application studies)

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories

Compute-Node Services

CTH Fragment Detection

Motivation

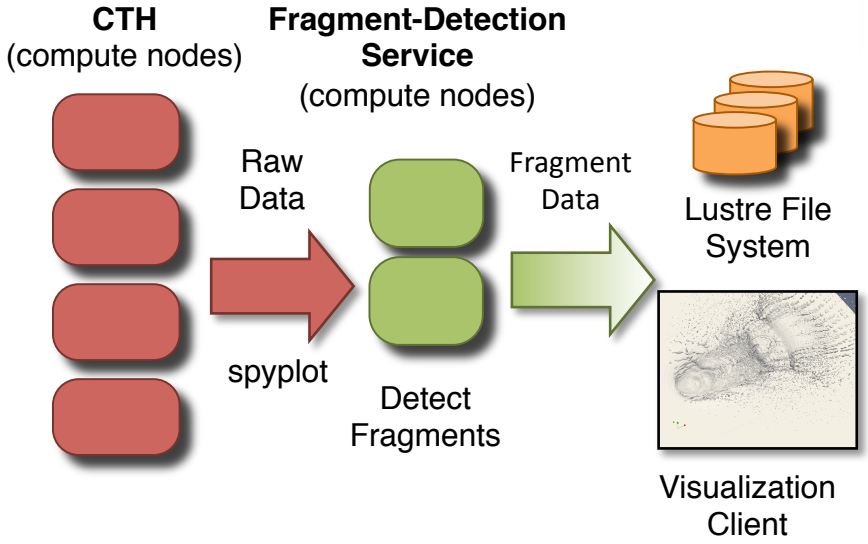
- Fragment detection requires data from every time step (I/O intensive)
- Detection process takes 30% of time-step calculation (scaling issues)
- Integrating detection software with CTH is intrusive on developer

CTH fragment detection service

- Extra compute nodes provide in-line processing (overlap fragment detection with time step calculation)
- Only output fragments to storage (reduce I/O)
- Non-intrusive
  - Looks like normal I/O (spyplot interface)
  - Can be configured out-of-band

Status

- Developing client/server stubs for spyplot
- Developing Paraview proxy service



Fragment detection service provides on-the-fly data analysis with no modifications to CTH.

LDRD
LABORATORY DIRECTED RESEARCH & DEVELOPMENT

Sandia National Laboratories