Application-Level Data Services

SAND2010-4677P

Approved for Public Release SAND2010-XXXXP

Staging Meeting

July 16, 2010

Ron Oldfield
Sandia National Laboratories

Sandia National Laboratories
Projects supported by ASC and LDRD programs

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.

Sandia National Laboratories

Application-Level Data Services

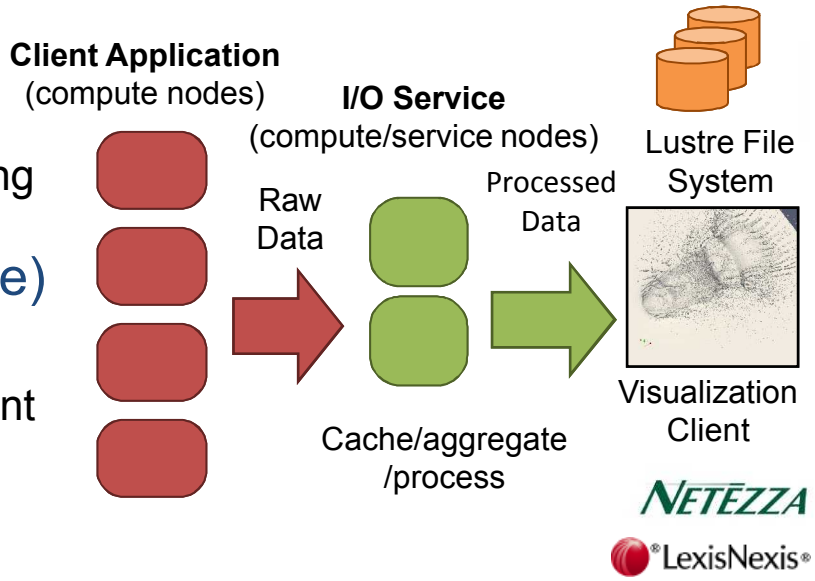
Our I/O Research is About Reducing I/O

Application-Level Data Services

- Leverage available compute/service node resources for I/O caching and data processing

Network Scalable Service Interface (Nessie)

- Developed for the Lightweight FS Project
- Framework for HPC client/server development
- Designed for scalable data movement
- Asynchronous RPC-like API



Examples

- Preprocessing for seismic imaging
- netCDF caching service
- SQL Proxy for HPC/Database Integration
- CTH Particle tracking
- Sparse-matrix viz, real-time network analysis

ASC

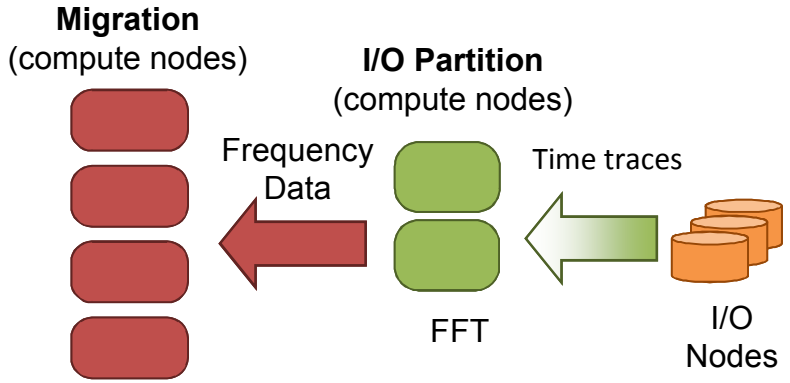
Sandia National Laboratories

Application-Level Data Services

We did this for Salvo Seismic Imaging (circa 1996)

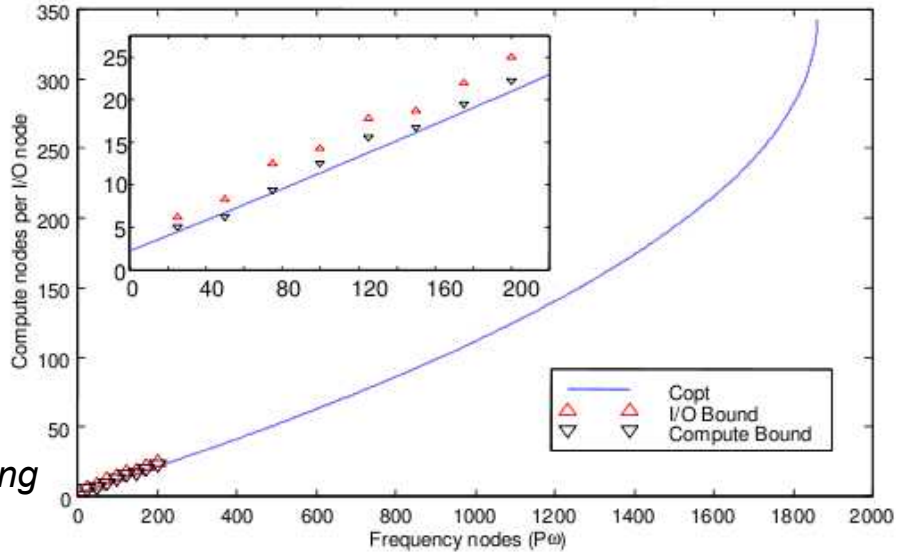
Salvo's I/O Partition

- Partition of application processors (used separate MPI Communicator for I/O)
- Used for FFT, I/O cache, and interpolation
- Async I/O allowed overlap of I/O and computation (pre-process next step)



Results

- +10% nodes led to +30% in performance
- Modeling I/O and compute costs helped find the right balance of compute and I/O nodes



Contacts: Ron Oldfield, Curtis Ober
{raoldfi,ccober}@sandia.gov

Oldfield, et al. Efficient parallel I/O in seismic imaging. *The International Journal of High Performance Computing Applications*, 12(3), Fall 1998

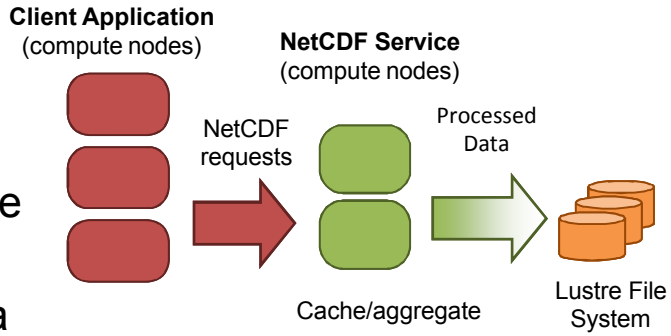
Sandia National Laboratories

Application-Level Data Services

NetCDF I/O Cache

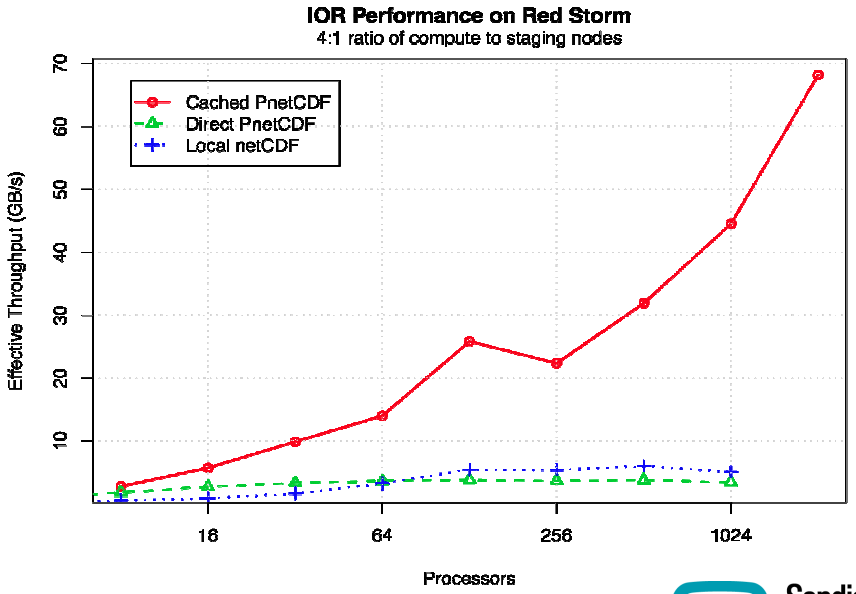
Motivation

- 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 Caching Service

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



Application-Level Data Services

CTH Fragment Detection

Motivation

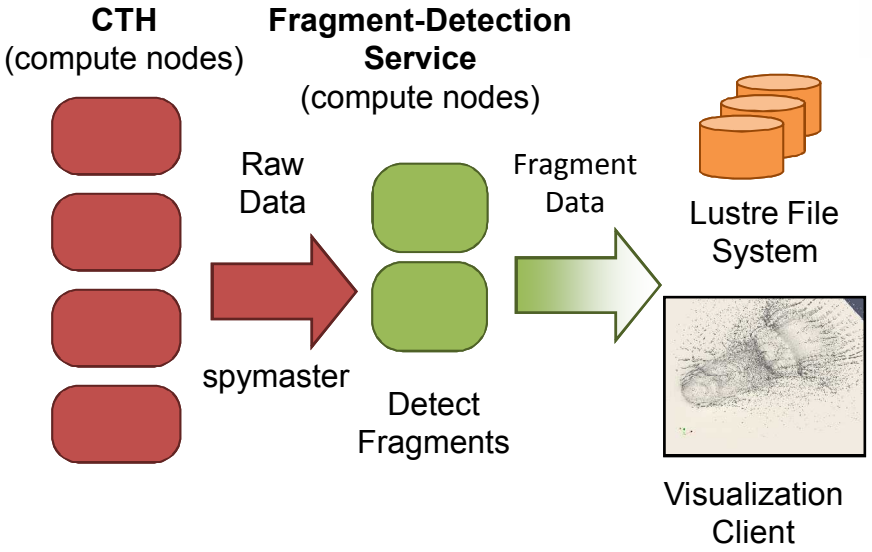
- Fragment detection requires data from every time step (I/O intensive)
- Detection process takes 30% of time-step calculation (scaling issues)
- Integrating detection software with CTH is intrusive on developer

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 (spymaster interface)
  - Can be configured out-of-band

Status

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

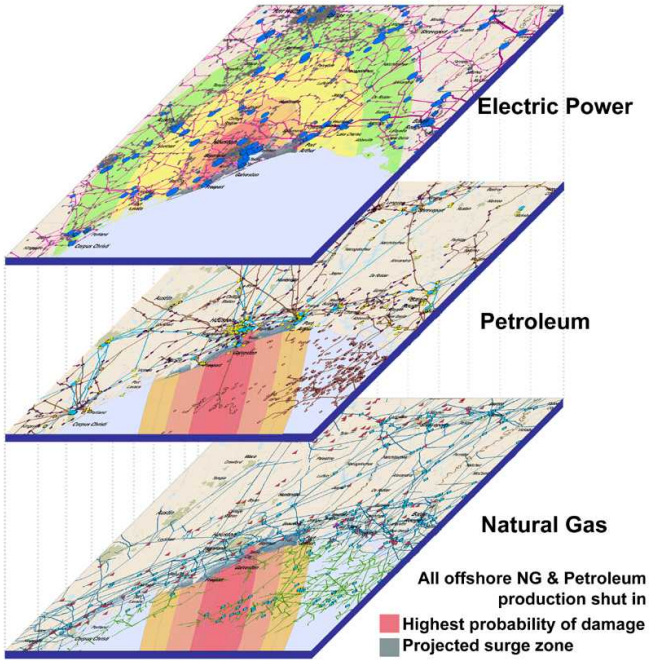


Fragment detection service provides on-the-fly data analysis with no modifications to CTH.

Application-Level Data Services

A Database Service for NISAC/N-ABLE

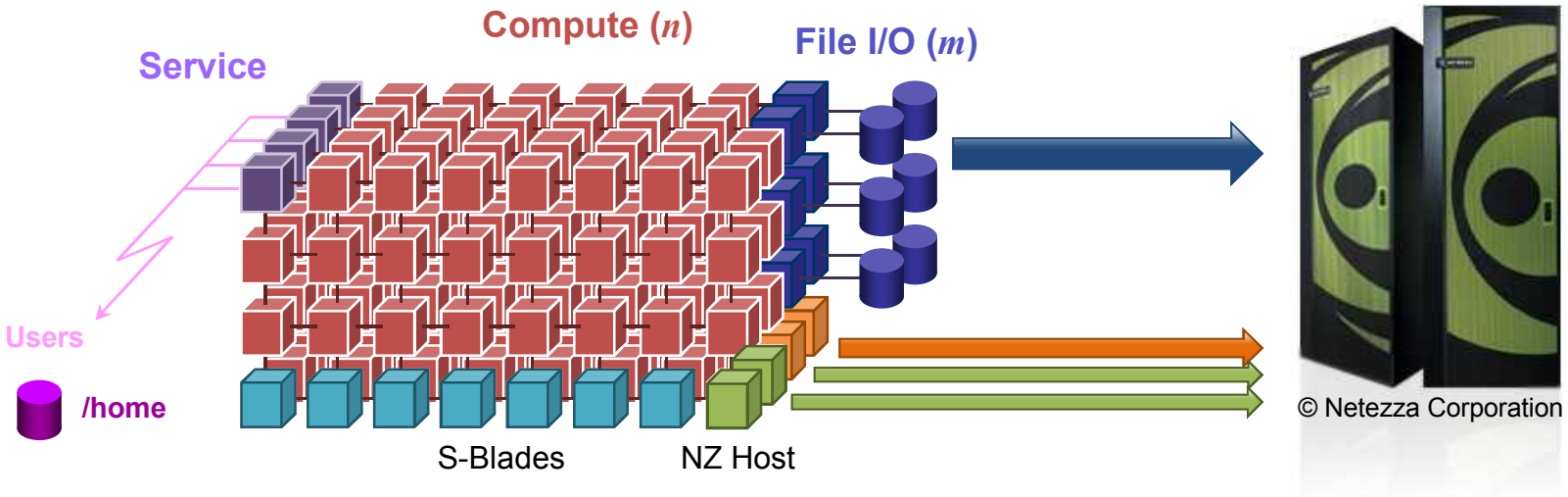
- Model economic impact of disruptions in infrastructure
  - Changes in U.S. Border Security technologies
  - Terrorist acts on commodity futures markets
  - Transportation disruptions on regional agriculture and food supply chains
  - Optimized military supply chains
  - Electric power and rail transportation disruptions on chemical supply chains
- Compute and data challenges
  - Models economy to the level of the individual firm
  - Model transactions from 10s of millions of companies
  - Simulation data ingested into DB for analysis
  - DB ingest is bottleneck (10x time to simulate data)
  - Time to solution is critical… want answers in hours



NISAC identifies potential consequences of disruptions to infrastructures and analyzes cascading impacts due to interdependencies

Sandia National Laboratories

Hybrid Architecture Evolution for Database Services



Research Questions (yet to be answered)

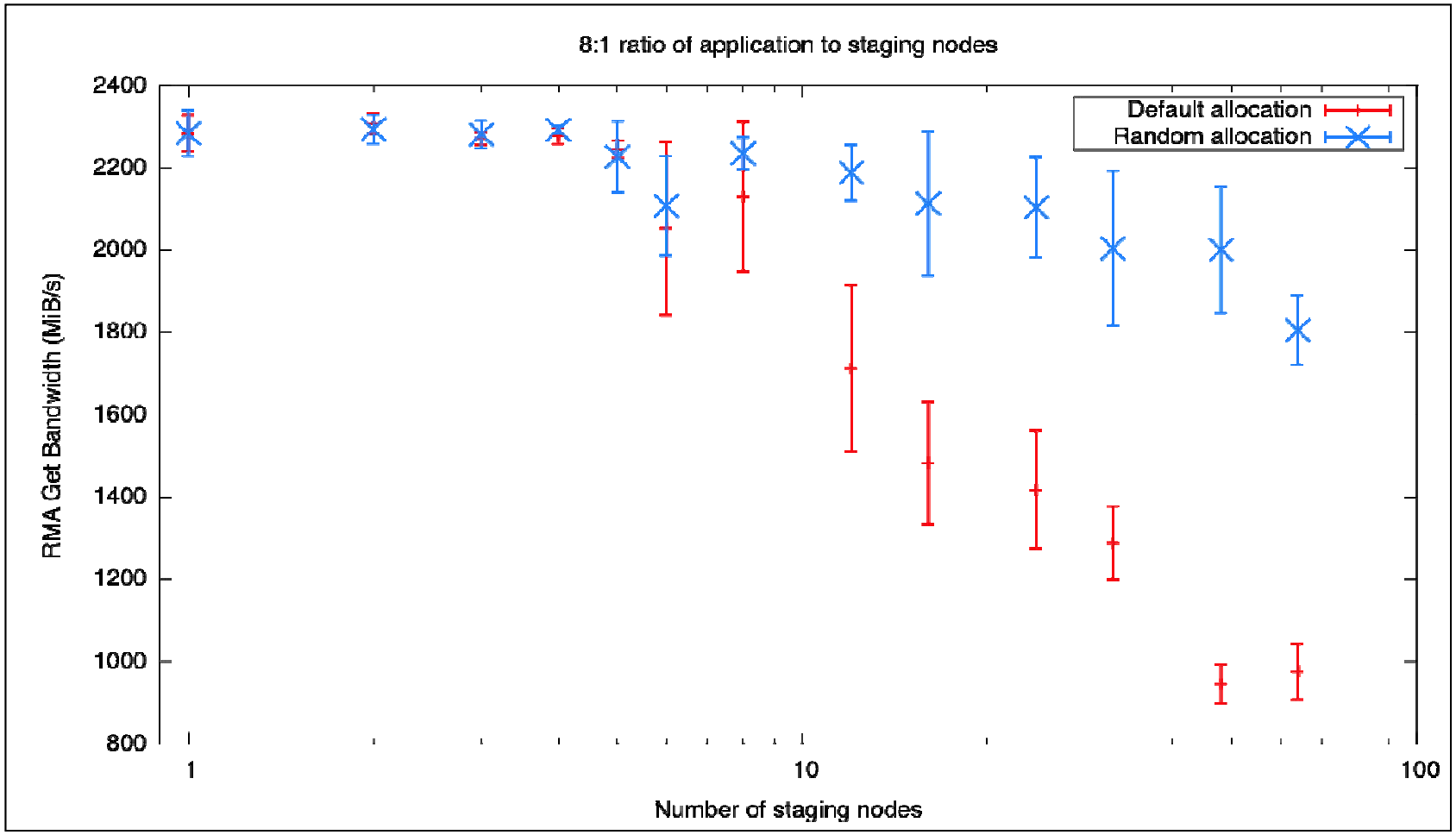
- What ingest rates will keep up with scientific workloads?
- Where are bottlenecks? Between host and S-BLADE?
- What software/networking infrastructure will resolve the bottlenecks?

An evolving architecture to support rapid ingest for HPC workloads

1) Stage data to FS during sim, bulk load to DB after. (post-processing)
2) SQL Server sends ODBC requests to remote Netezza (slow network to host)
3) SQL Server becomes host (fast access to host, slow to S-BLADEs)
4) Multiple service-node hosts (parallel access to back-end S-BLADEs)
5) Really wacky! Hosts and S-BLADEs on fast network (fully integrated)

Sandia National Laboratories

How Can System Software Help?



Sandia National Laboratories