SAND2011-8597C

Extending Scalability of Collective IO Through Nessie and Staging

Parallel Data Storage Workshop

November 13, 2011

Jay Lofstead (SNL)
Ron Oldfield (SNL)
Todd Kordenbrock (HP)
Charles Reiss (UCB)

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Motivation

- Collective & Two-Phase IO proven beneficial
  - Relatively modest data volumes, 1-D, 2-D
- But….
- Trade-off of inter-node communication for data reorganization to save IO not always beneficial
  - Large datasets
  - 3-D domain decompositions particularly bad

Motivation

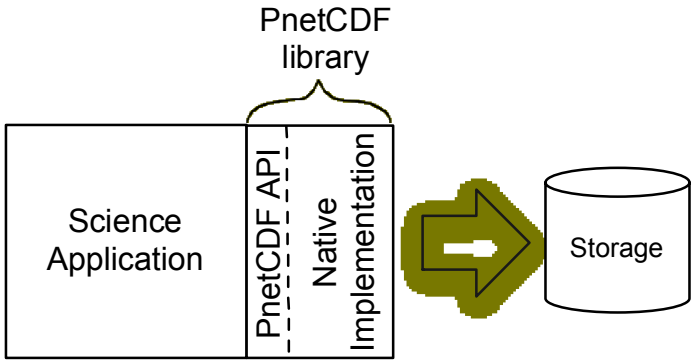
- **Problem: technique is central to some IO APIs**
  - **netCDF4, HDF5**
- **Problem: Changing IO techniques/file format may not be an option for some applications**
  - **CESM climate model is committed to netCDF file format**
- **Problem: Continued scaling of problem sizes and resolution making things worse**

Solution

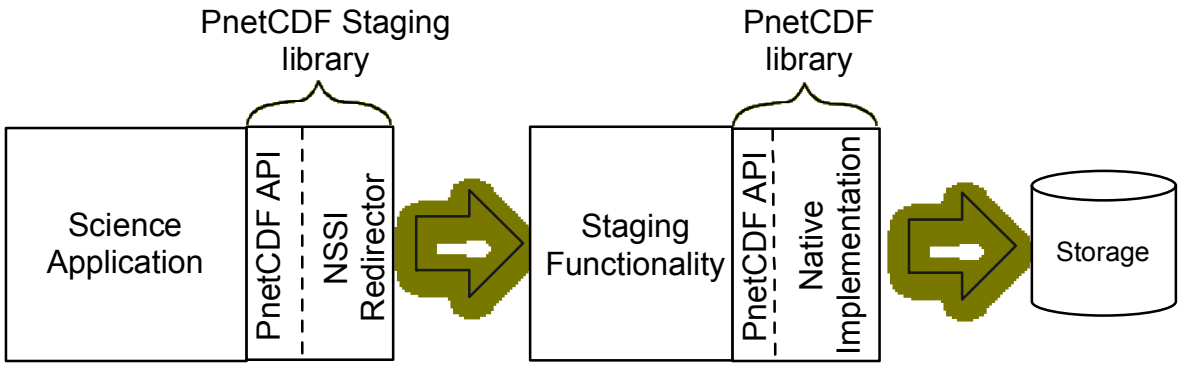
- Use efficient transport layer and data staging transparently in the IO stack
  - Re-implement native IO API (link-time compatible)
  - Ensure format on disk is identical
- Requirements
  - Efficient, portable transport layer
  - Staging area functionality to reduce time to completion

Solution Architecture

Native



Nessie



Jay Lofstead
www.lofstead.org
gflofst@sandia.gov

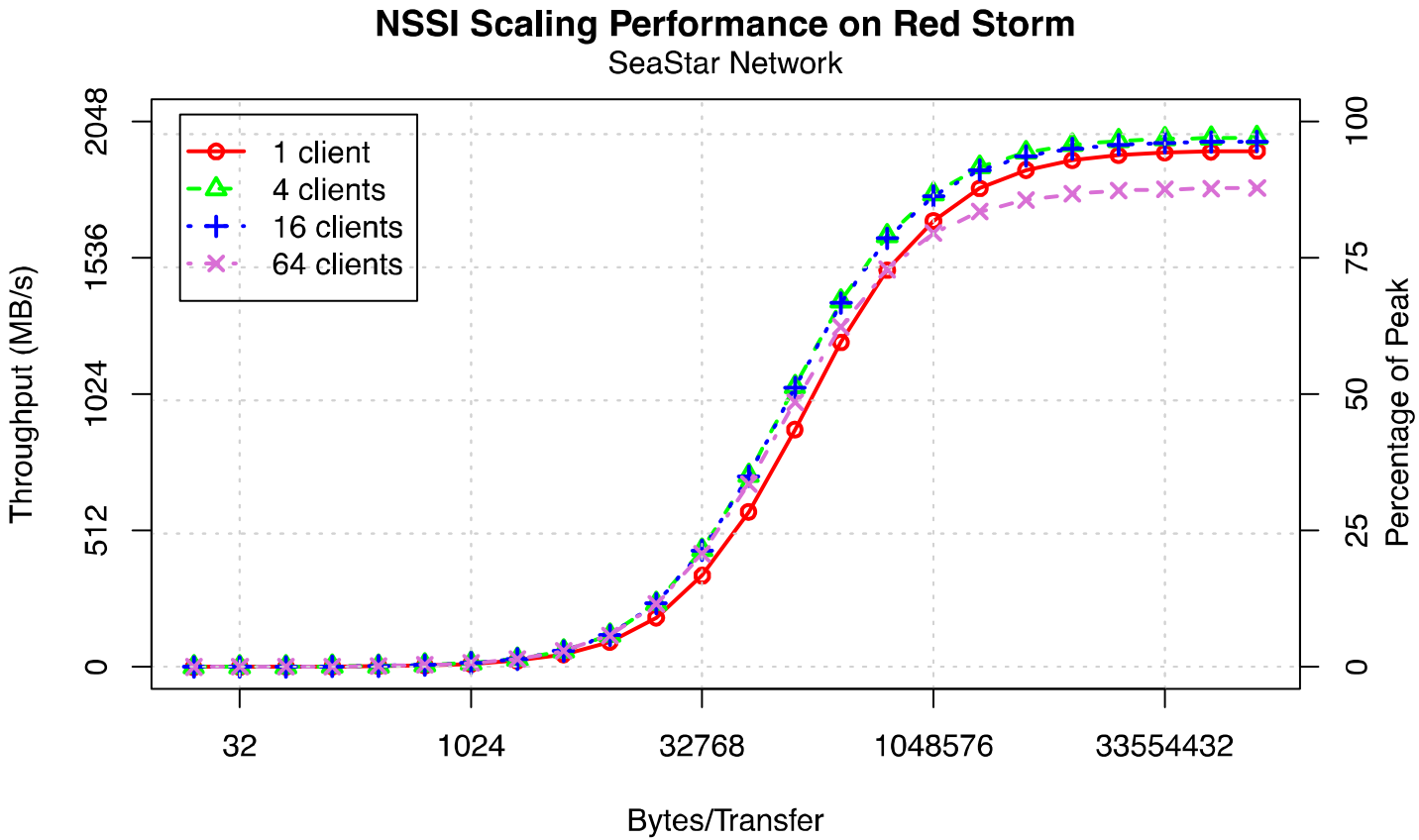
Nessie Transport Layer

- Network Scalable Services Interface (Nessie)
  - Originally developed for the Lightweight File Systems project
  - RPC-like asynchronous API layer supporting RDMA
  - Physical layer support
    - InfiniBand
    - Portals
    - Cray Gemini
  - Server-directed for bulk data
    - Writes: pull from client
    - Reads: push to client

Staging Functionality

- Collect data packets prior to writing to storage
  - Cache data chunks to afford other optimizations
- Perform data rearrangement
  - Perform partial data rearrangement like two-phase IO
- Use different techniques for writing to storage
  - Direct, Caching, Aggregation

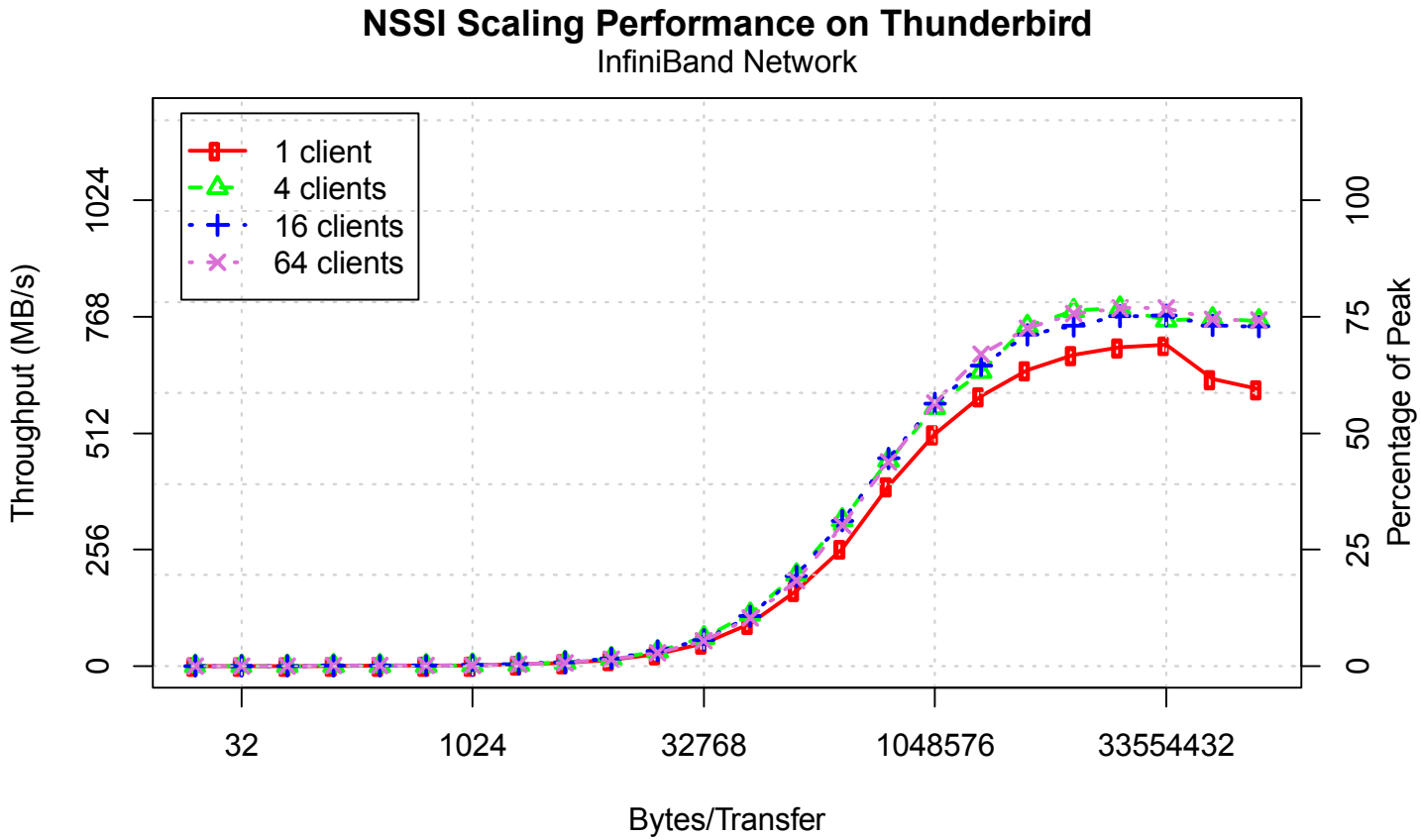
Nessie Performance (Portals)



Performance of xfer_write_rdma on Red Storm

Jay Lofstead
www.lofstead.org
gflofst@sandia.gov

Nessie Performance (InfiniBand)

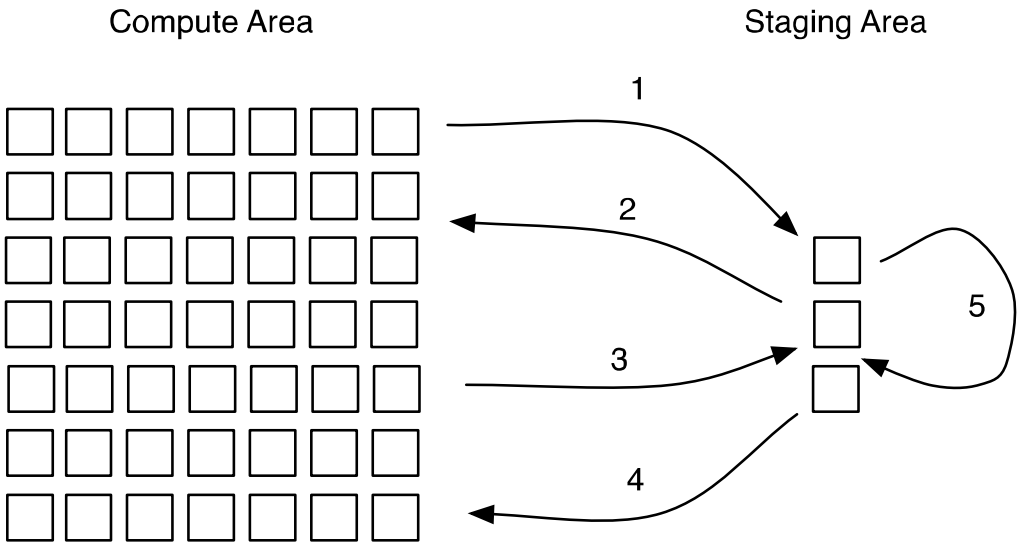


Performance of xfer_write_rdma on Thunderbird

Jay Lofstead
www.lofstead.org
gflofst@sandia.gov

NetCDF Staging Operation

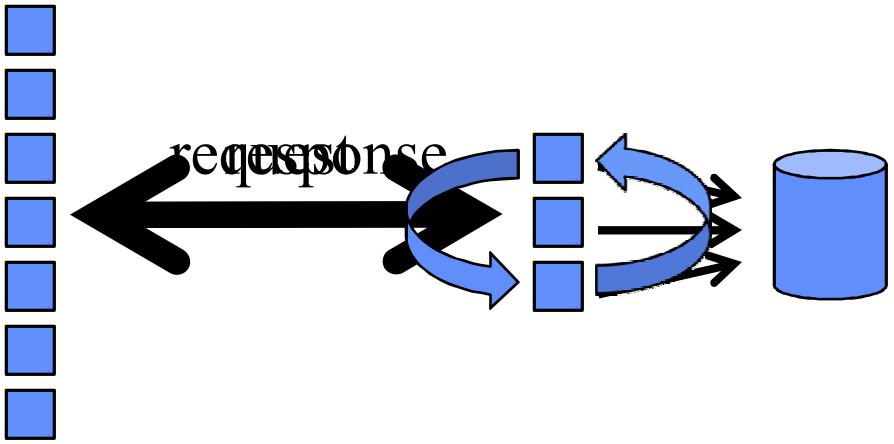
1. Initiate Request
2. Start data retrieval
3. Move data
4. Put completion/result
5. Process in staging area



Jay Lofstead
www.lofstead.org
gflofst@sandia.gov

NetCDF Staging Functionality

- NetCDF4 and PnetCDF API supported
- Direct – synchronous with client calls
- Cache Independent – asynch with client calls
- Aggregate Independent – asynch with client calls, but aggregate data prior to writing (on node only)

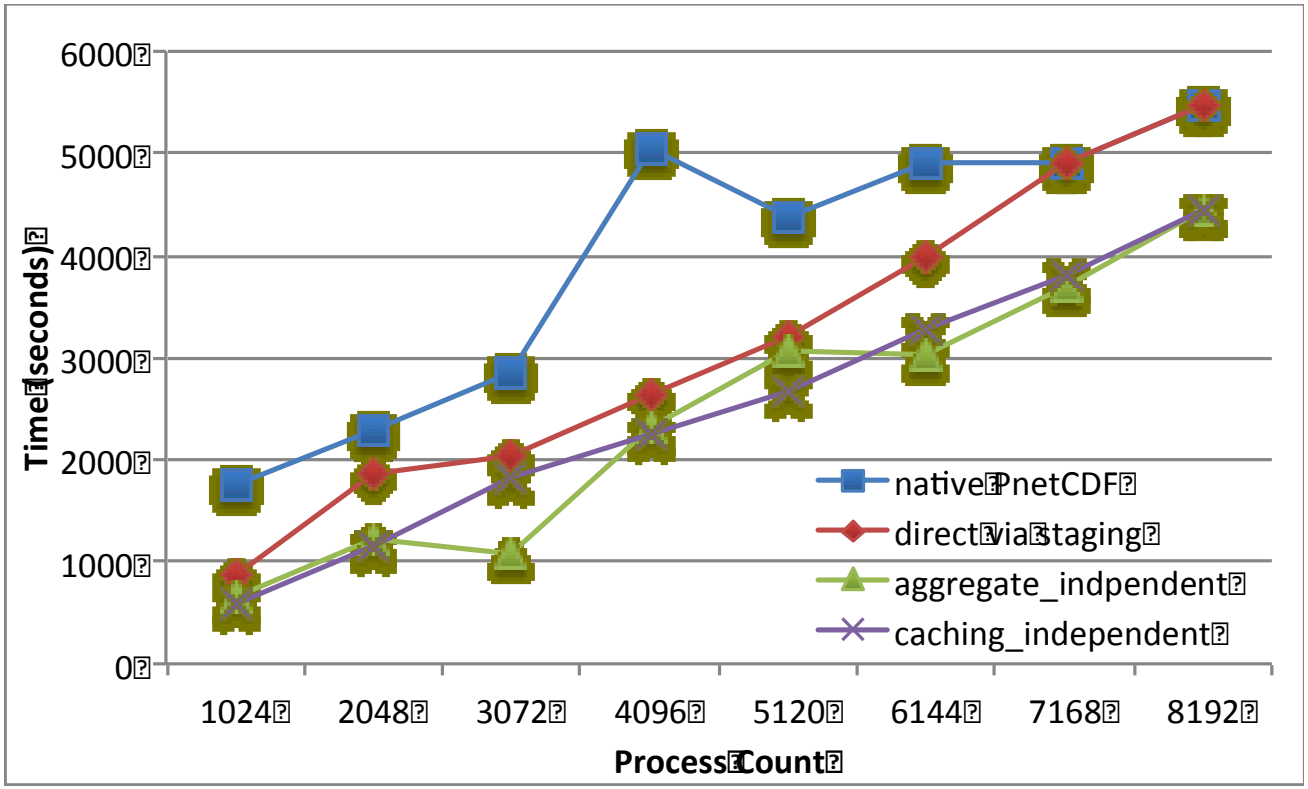


Jay Lofstead
www.lofstead.org
gflofst@sandia.gov

NetCDF Staging Functionality

- Untested functionality for this paper
  - Collective IO versions of cache and aggregate – like independent versions, but a maximal number of collective IO calls made for writing
  - Other data manipulation
    - Different implementation using Nessie for data analysis hosting

NetCDF Staging Performance



Testing on JaguarPF using S3D IO kernel

Current Status

- Nessie and NetCDF staging now part of Trilinos
  - Trios capability area
- Port to BlueGene underway
- Integration of accelerators for staging processing

Future Work

- Finish tests on RedSky to isolate Lustre issues
- Test collective IO routines
- Examine impact on reading performance

- For Nessie, other applications
  - 'In flight' data analysis routines
  - Transactions for resilience in data staging (see our poster!)
  - Hybrid, high level IO routine complications
    - Exodus + NetCDF