# The I/O Software Stack: Present and Future

Rob Ross

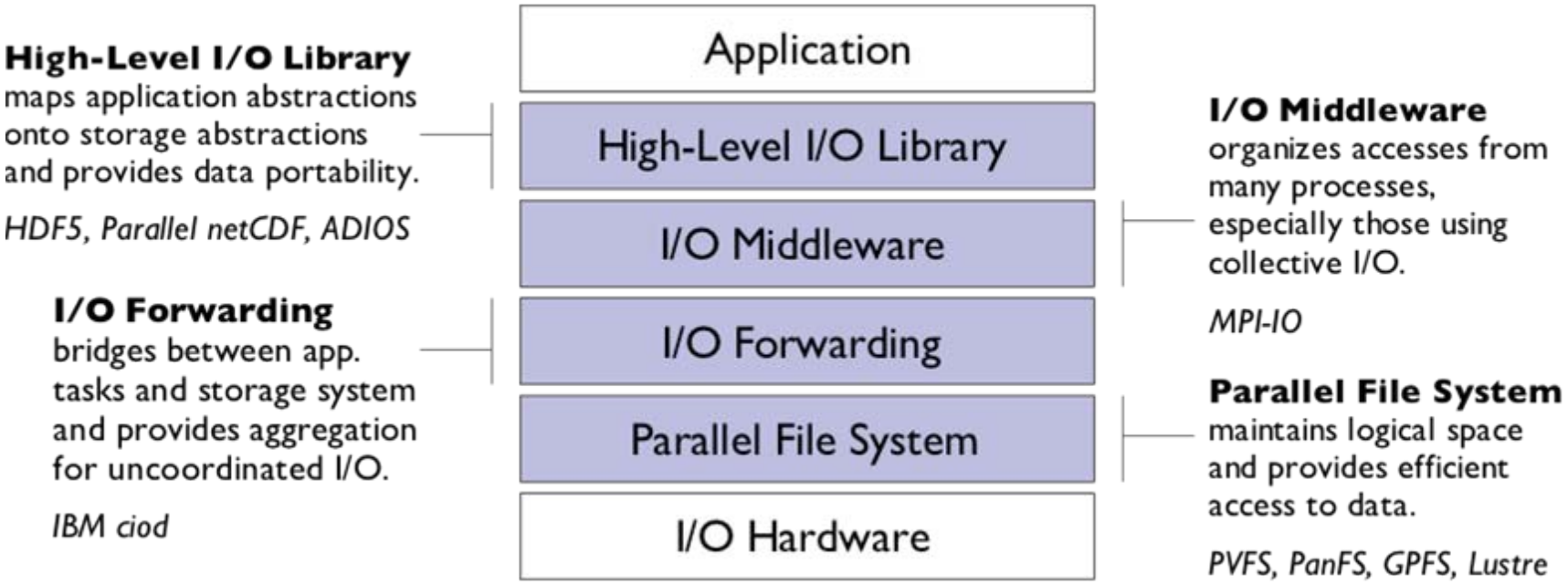
Mathematics and Computer Science Division

Argonne National Laboratory

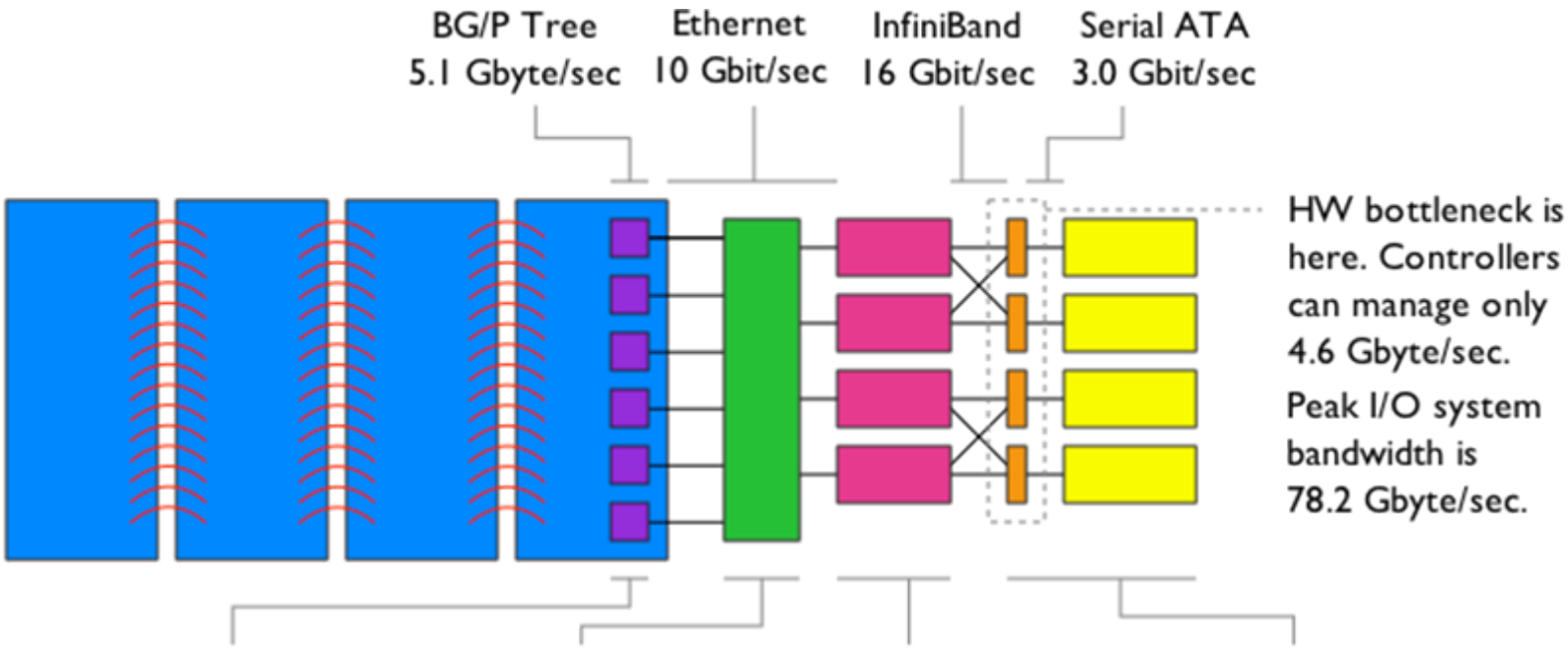
# Acknowledgments


- S. Lang, R. Latham, P. Peterka, R. Thakur (ANL)
- A. Choudhary, K. Gao, W-K. Liao, A. Nisar (NWU)
- S. Klasky, C. Jin, W. Yu (ORNL)
- J. Lofstead (GaTech)

# The I/O Software Stack



# PVFS on IBM Blue Gene/P




**Gateway nodes** run parallel file system client software and forward I/O operations from HPC clients.

*640 Quad core PowerPC 450 nodes with 4 Gbytes of RAM each*

**Commodity network** primarily carries storage traffic.

*900+ port 10 Gigabit Ethernet Myricom switch complex*

**Storage nodes** run parallel file system software and manage incoming FS traffic from gateway nodes.

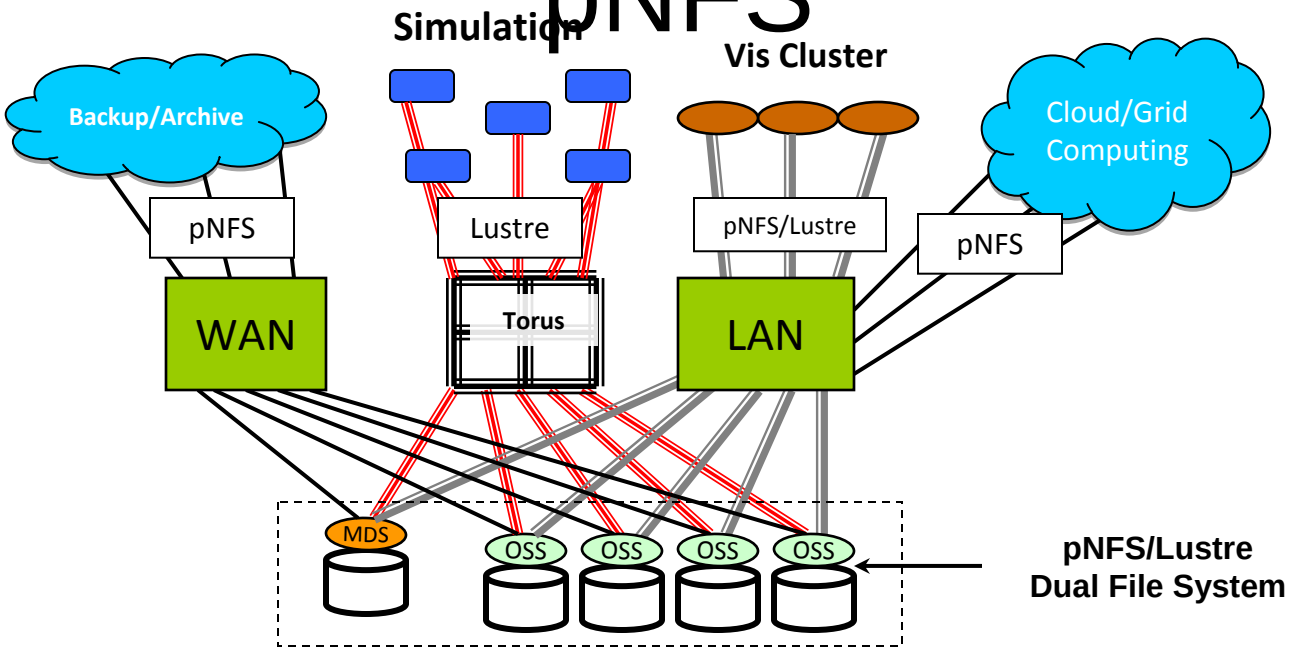
*136 two dual core Opteron servers with 8 Gbytes of RAM each*

**Enterprise storage** controllers and large racks of disks are connected via InfiniBand or Fibre Channel.

*17 DataDirect S2A9900 controller pairs with 480 1 Tbyte drives and 8 InfiniBand ports per pair*

Architectural diagram of the 557 TFlop IBM Blue Gene/P system at the Argonne Leadership Computing Facility.

# Ubiquitous Access to Lustre via pNFS

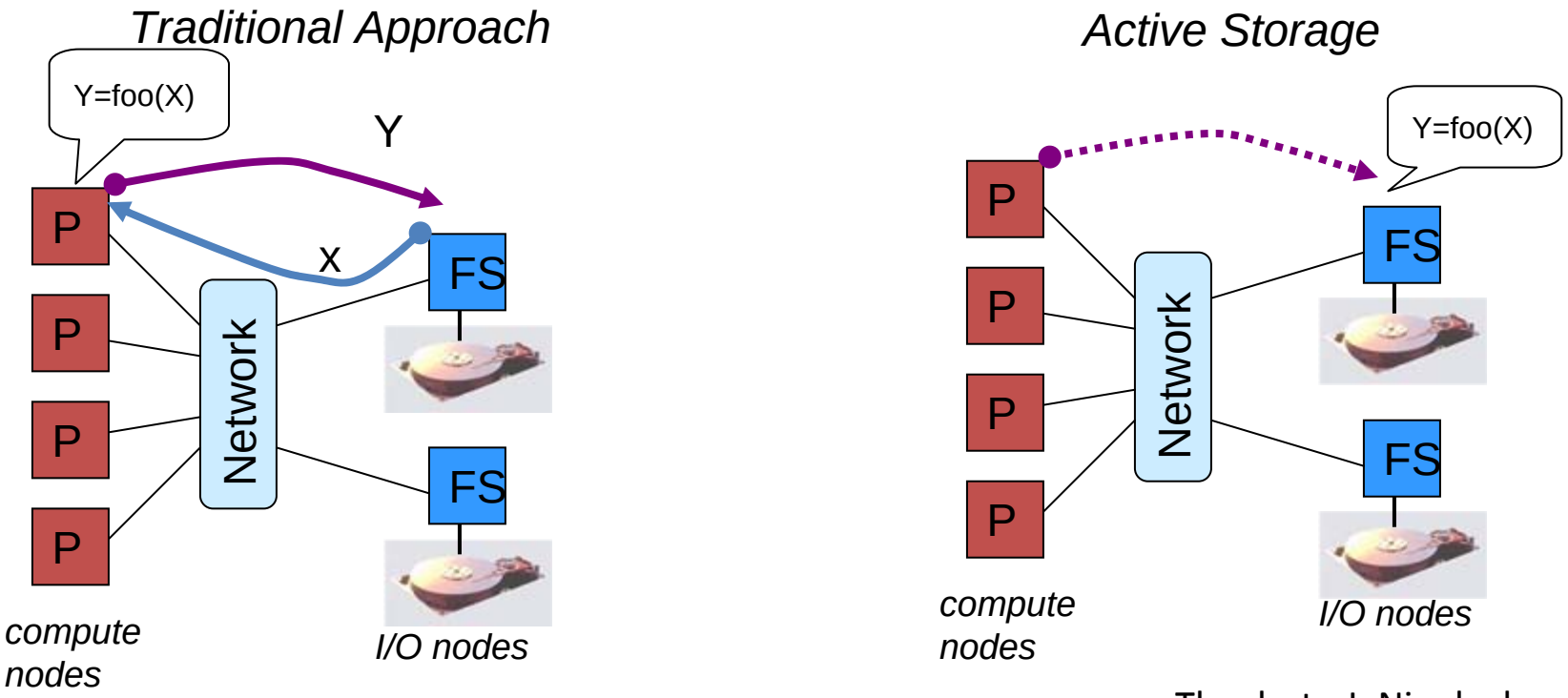



- Enable dual namespaces for versatile usage of Lustre storage
  - Co-located pNFS MDS with Lustre MDS, and pNFS DSes with Lustre OSSes
  - With help from Lustre Center of Excellence at ORNL
- Eliminate data copies across simulation, visualization and data analysis clusters
- Expose Lustre storage as a nearline solution to cloud/grid computing
- Integrate Lustre storage seamlessly to remote data centers for backup/archiving

Thanks to W. Yu.

# Active Storage in Parallel File Systems

- Active Storage exploits the old concept of moving computing to the data source
- Avoids data movement across the network in parallel machine by allowing applications use compute resources on the I/O nodes of the cluster for data processing




Thanks to J. Nieplocha et. al.

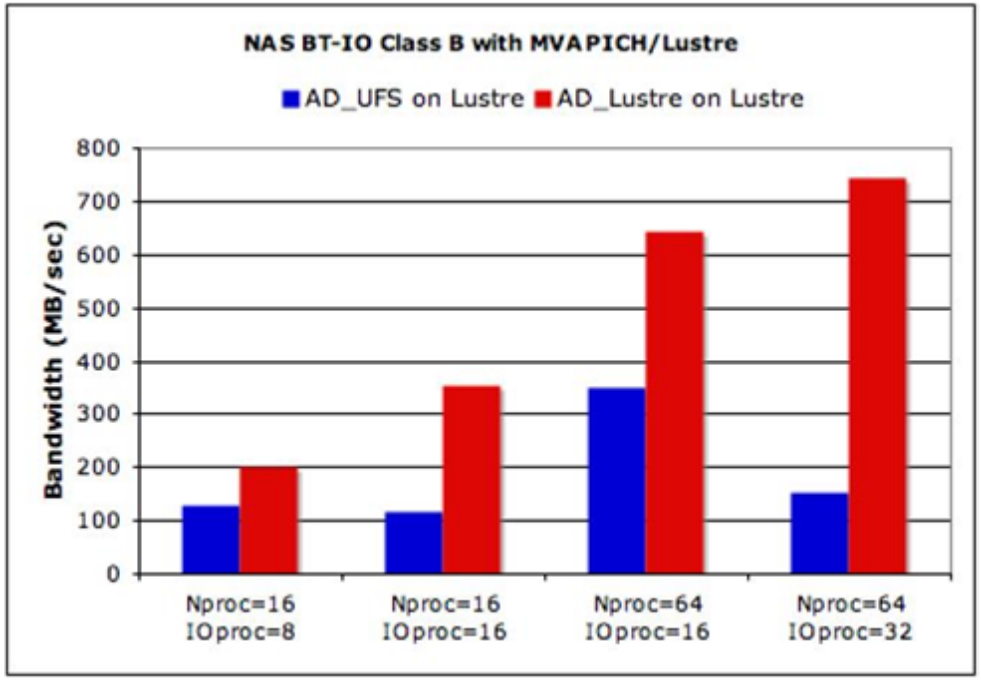
# I/O Forwarding

- IBM ciod on BG/P is current “standard”
- Zoid implementation (FASTOS funding)
- NWU work in collaborative caching in I/O forwarding layers
- Portable I/O forwarding layer underway (IOFSL project, more FASTOS funding)
- SDM team integrating into MPI-IO, hooking into PVFS

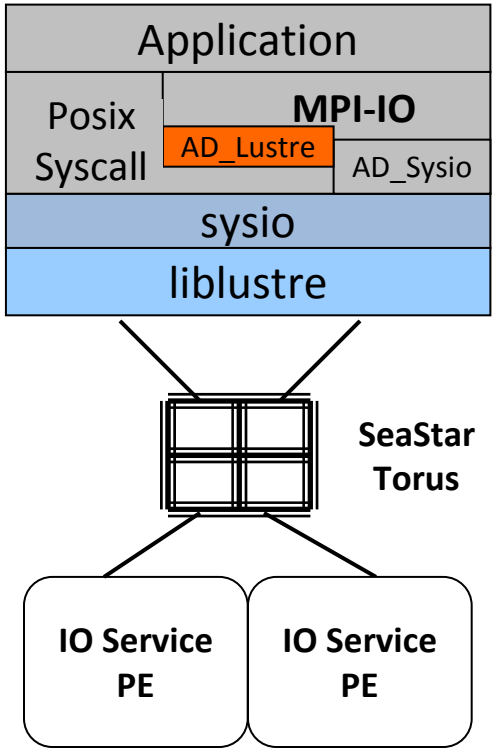
# MPI-IO Driver for Lustre

- Available for Beowulf clusters and Cray XT
- Overcome the restriction of a proprietary MPI-IO stack on Cray XT
- Enabled arbitrary striping specification over Cray XT
- Lustre stripe-aligned file domain partitioning
- Released via MVAPICH-1.0 and MPICH2-1.0.7

**Performance on an 80-node beowulf cluster**

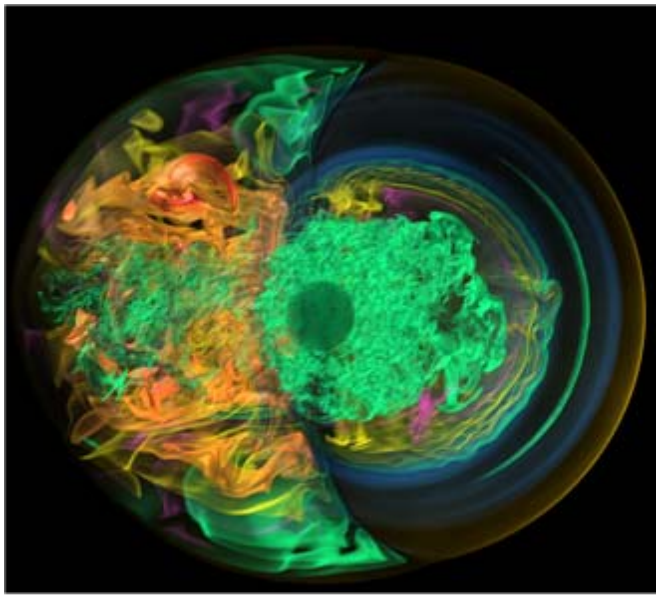


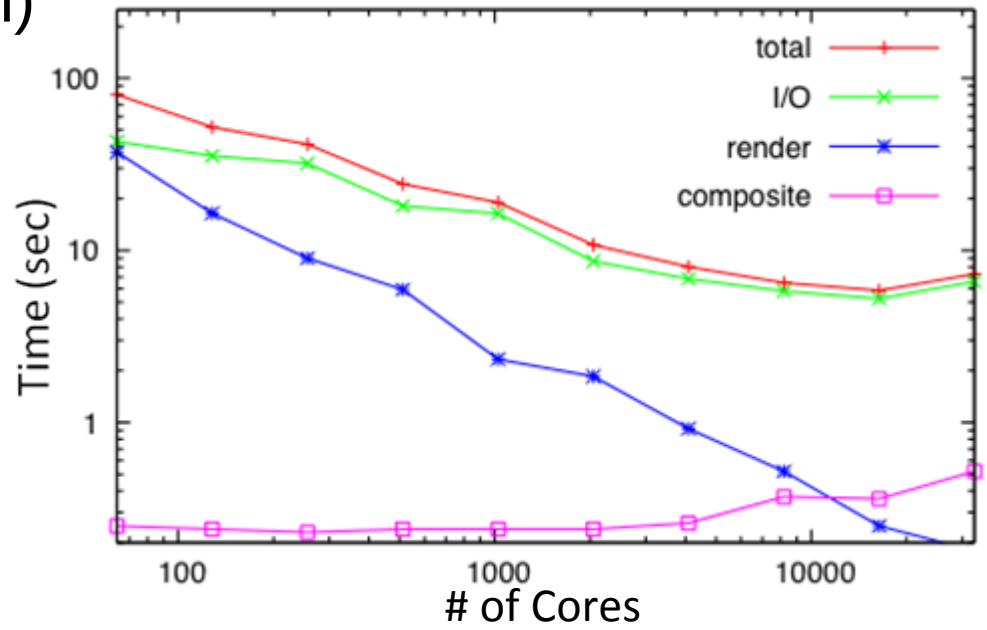

**Software Diagram on Cray XT**



# Parallel Volume Rendering

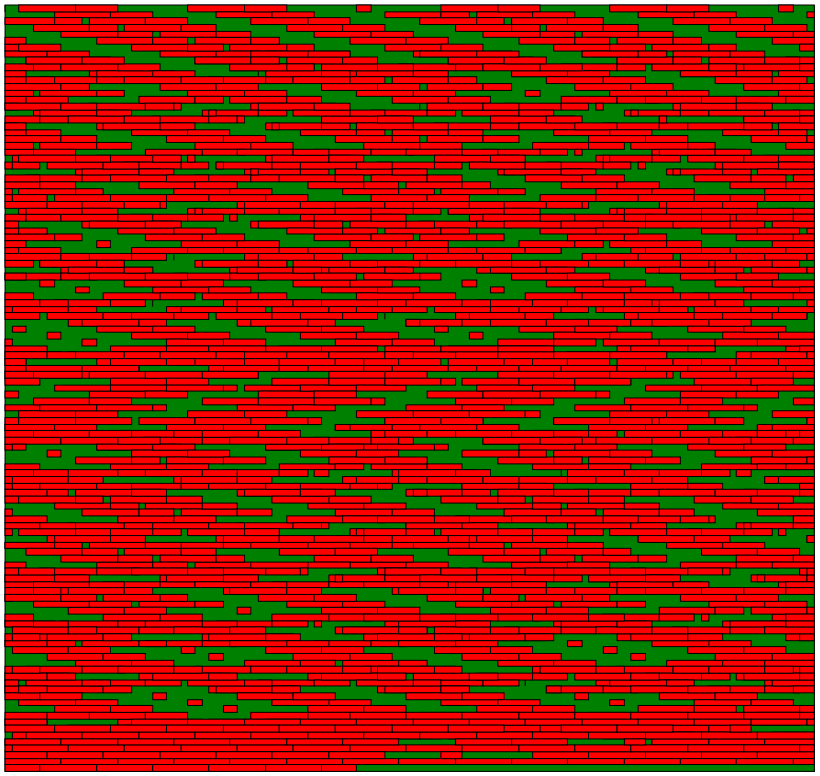
- Supernova model with focus on core collapse
- Parallel rendering techniques scale to 16k cores on Argonne Blue Gene/P
- Produce a series of time steps
- $1120^3$ elements (~1.4 billion)
- Structured grid
- Simulated and rendered on multiple platforms, sites
- I/O time now largest component of runtime

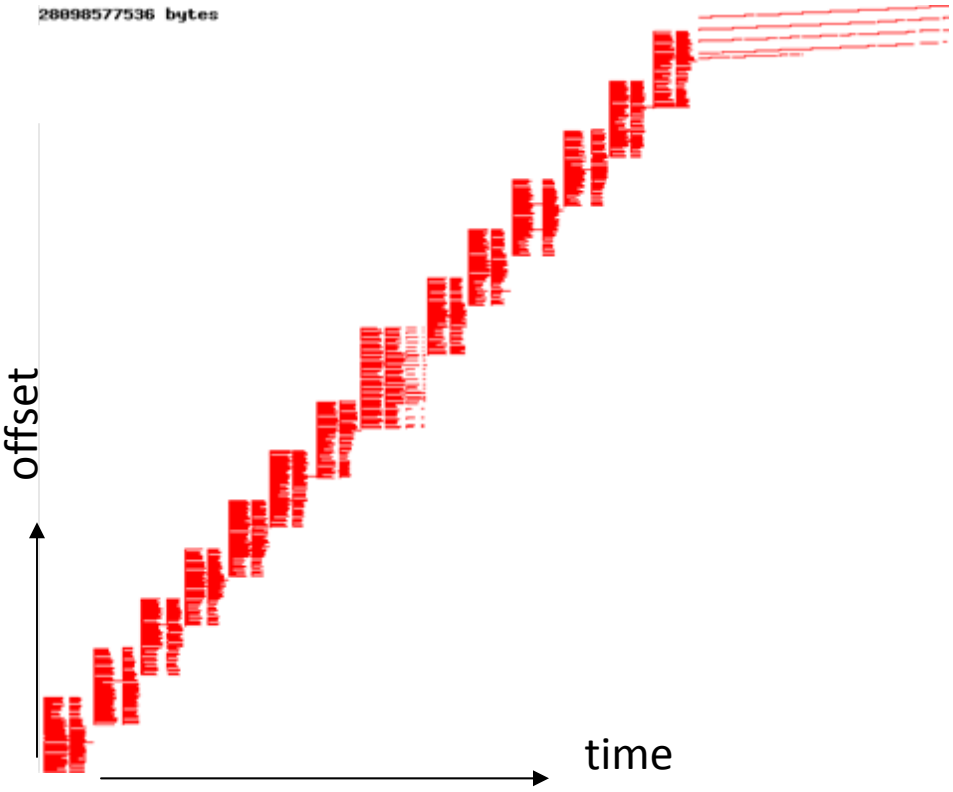





Thanks to T. Peterka et al.

# Parallel netCDF (no hints)

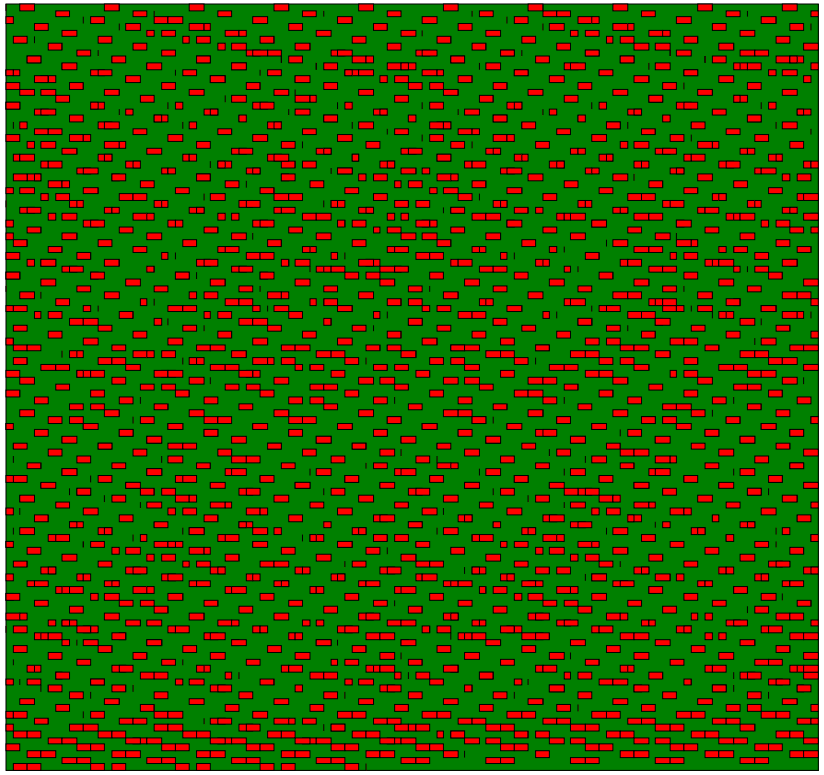


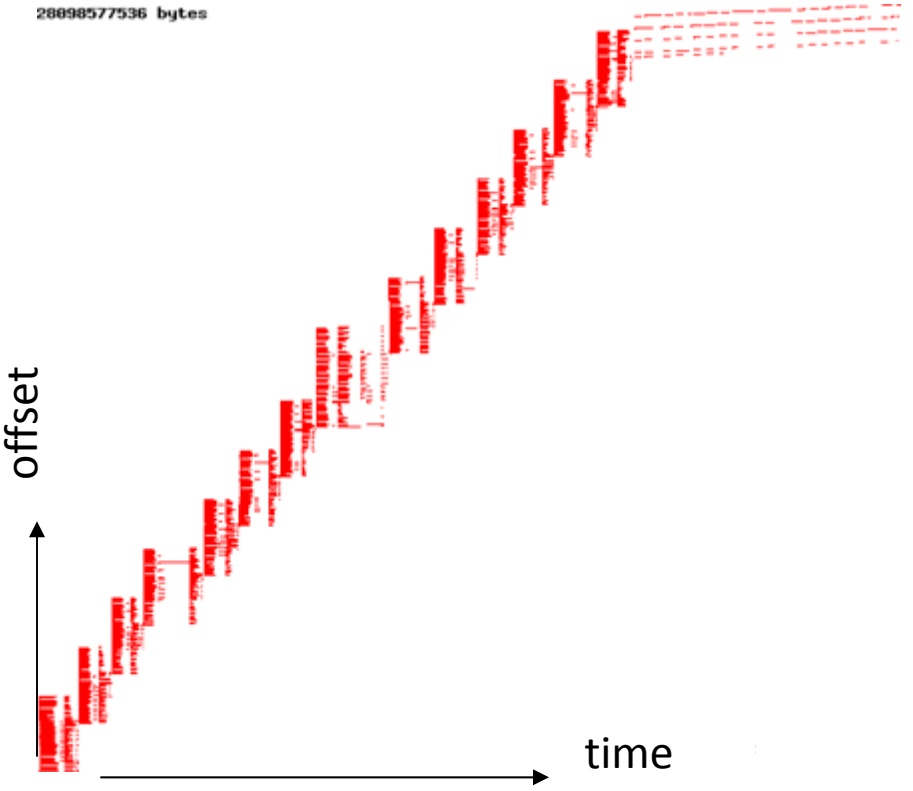



- Block depiction of 28 GB file
- Record variable scattered
- Reading in way too much data!

- Y axis larger here
- Default “cb_buffer_size” hint not good for interleaved netCDF record variables

# Parallel netCDF (hints)

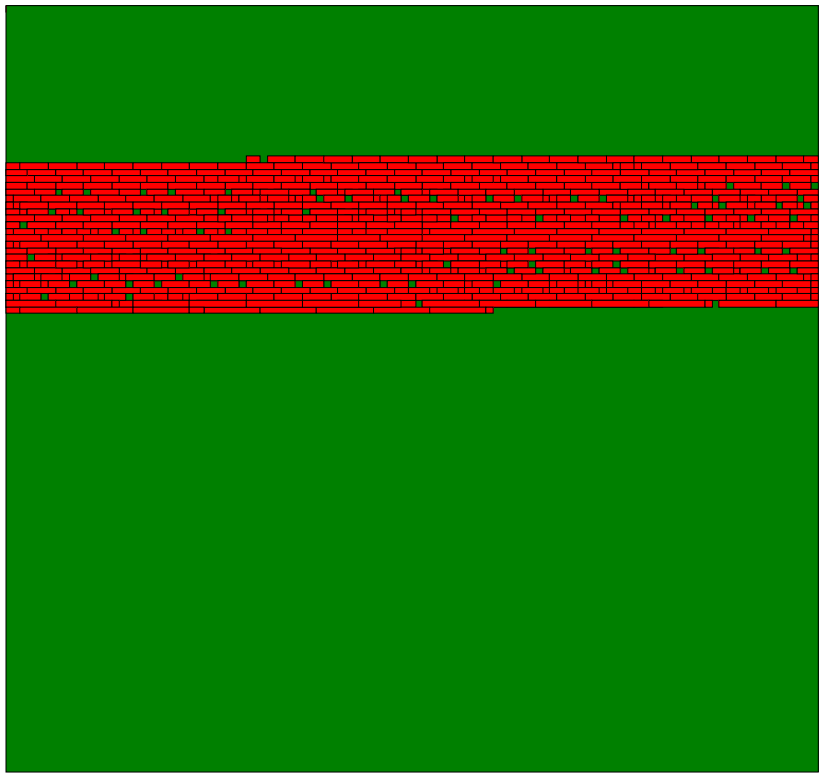




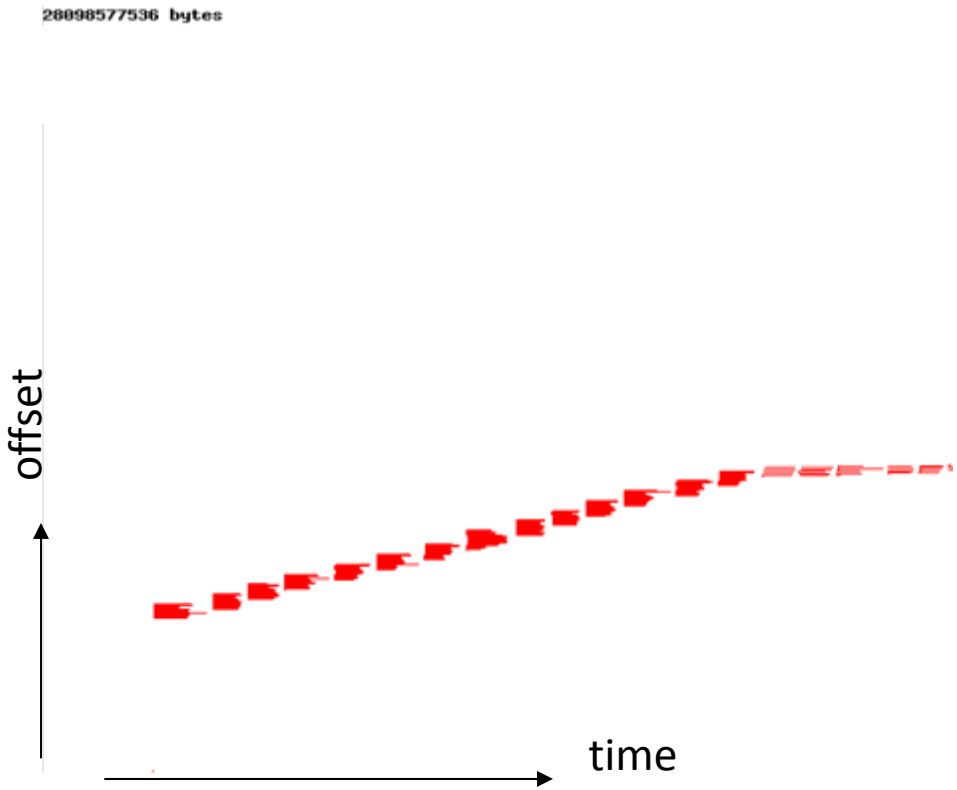

- With tuning, much less reading
- Better efficiency, but still short of MPI-IO

- Still some overlap
- “cb_buffer_size” now size of one netCDF record
- Better efficiency, at slight perf cost

# Parallel netCDF (current SVN)



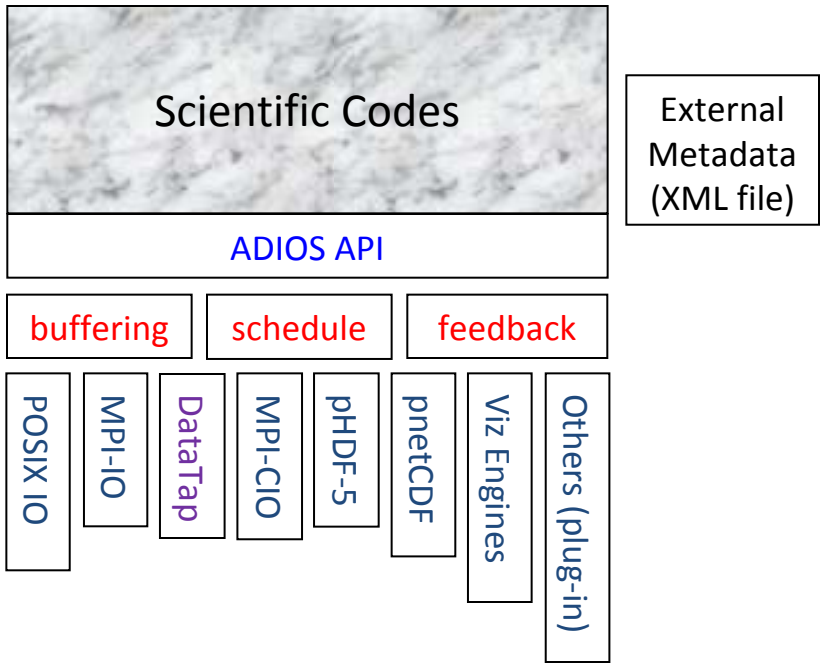
- Development effort to relax netCDF file format limits
- No need for record variables
- Data nice and compact like MPI-IO and HDF5




- Rank 0 reads header, broadcasts to others
  - Much more scalable approach
- Approaching MPI-IO efficiency
- Maintains netCDF benefits
  - Portable, self-describing, etc.

# ADIOS

- Allows plug-ins for different I/O implementations
- Abstracts the API from the method used for I/O
- Simple API, almost as easy as F90 write statement
- Best practice/optimized IO routines for all supported transports "for free"
- Thin API
- XML description of structure
  - data groupings with annotation
  - IO method selection
  - buffer sizes
- Supports common optimizations
  - Buffering
  - Scheduling
- Pluggable IO routines




Thanks to S. Klasky et al.

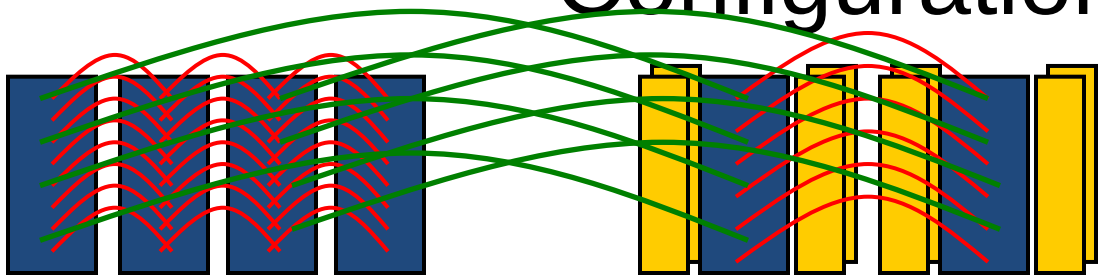
# The Revolutionary Approach to Storage

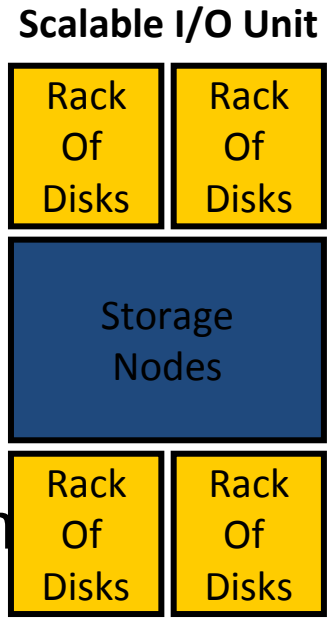
(or, We're Being Too Heavily Taxed!)

- Using enterprise SAN is costly
  - The hardware is expensive
  - Only provide a small fraction of data disk bandwidth (~20-30%)
- Building an external network (mainly) to attach storage is costly as well

- If we are willing to **manage resiliency ourselves**, we can afford to toss enterprise storage controllers
- **Directly attaching storage system to the compute system** can eliminate extra hops and bottlenecks that reduce the percentage of available disk BW
- We wouldn't be the first revolutionaries

# A Revolutionary Hardware Configuration






- Storage attached via low latency, high bandwidth links directly into petascale system
- Scalable I/O units
  - Interconnect within/between units to facilitate resiliency
  - Higher ratio of storage nodes to disks
  - With laptop drives:
    - 4 racks, 8 drawers per rack, 192 drives per drawer = 6144 drives
    - Raw space with 1TB drives is 6.1PB
  - PCIe, eSATA, SAS for connectivity
    - PCI-Express 8x provides 4 GB/sec BW per storage node
    - 256+ GB/sec raw bandwidth in a scalable unit

# Closing Remarks

- SDM Center has activities across the I/O software stack
  - Complementary work is being performed under FASTOS, NSF, and DOE base funding
- Many SDM products are in use today
- Current SDM projects also looking to next generation of systems and applications
  - Active storage, pNFS, architectures, I/O forwarding and aggregation, …
  - The software will make or break the storage system