BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

Parallel I/O Performance: From Events to Ensembles

Lenny Oliker
Computational Research Division
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

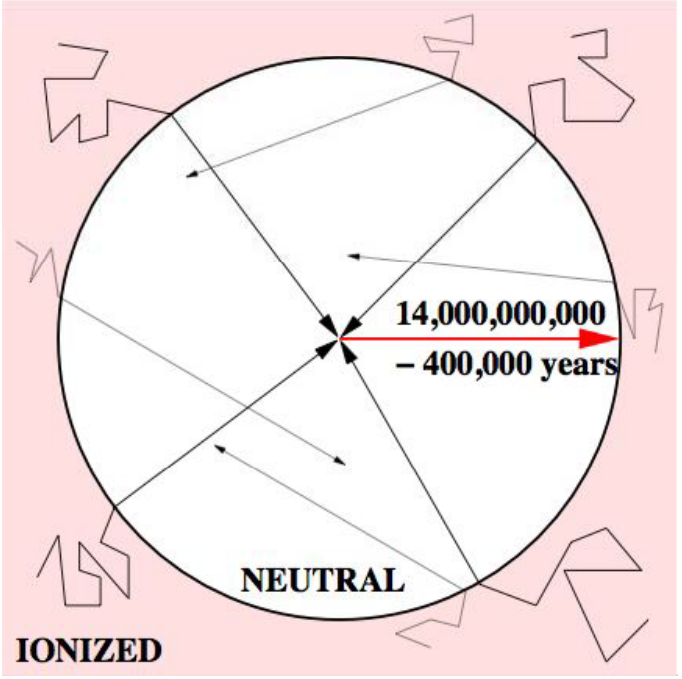
Parallel I/O Evaluation and Analysis

- Explosion of sensor & simulation data make I/O critical component
  - HPC centers are becoming larger consumers than producers of data
- Necessary to study I/O under realistic conditions and access patterns
  - Difficult to correlate existing I/O benchmarks to HPC requirements
- Present MADbench2: lightweight, portable, parameterized I/O benchmark
  - Derived from large-scale cosmology application
- Present IPM I/O – allows for lightweight, portable, scalable tracing
- Petascale I/O requires new techniques: analysis, visualization, diagnosis
- Statistical methods can be revealing – examine modes/spectroscopy
  - Present results using IOR, CHOMBO, MADbench and GCRM

Cosmic Microwave Background

- After Big Bang, expansion of space cools the Universe until it falls below the ionization temperature of hydrogen when free electrons combine with protons
- CMB is therefore a snapshot of the Universe at the moment it first becomes electrically neutral about 400,000 years after the Big Bang
- Tiny fluctuations in its temperature (1 in 100K) and polarization (1 in 100M) encode the fundamental parameters of the Universe
  - –Geometry, Composition, Ionization history

• Nobel prizes: 1978 (Penzias & Wilson) detection CMB, 2006 (Mather & Smoot) detection CMB fluctuations

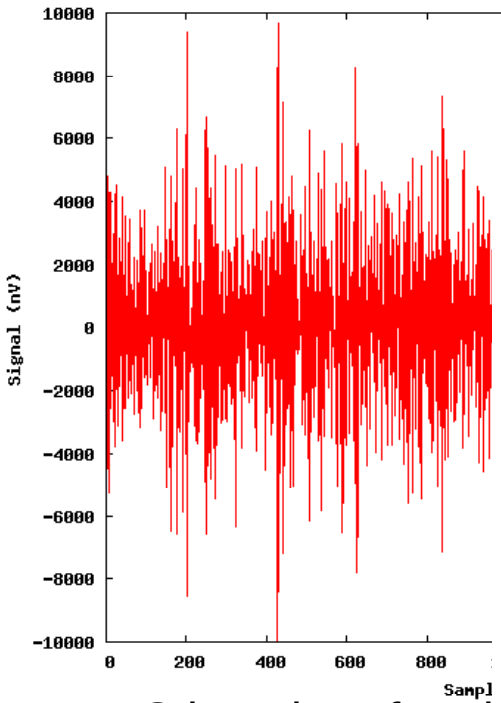


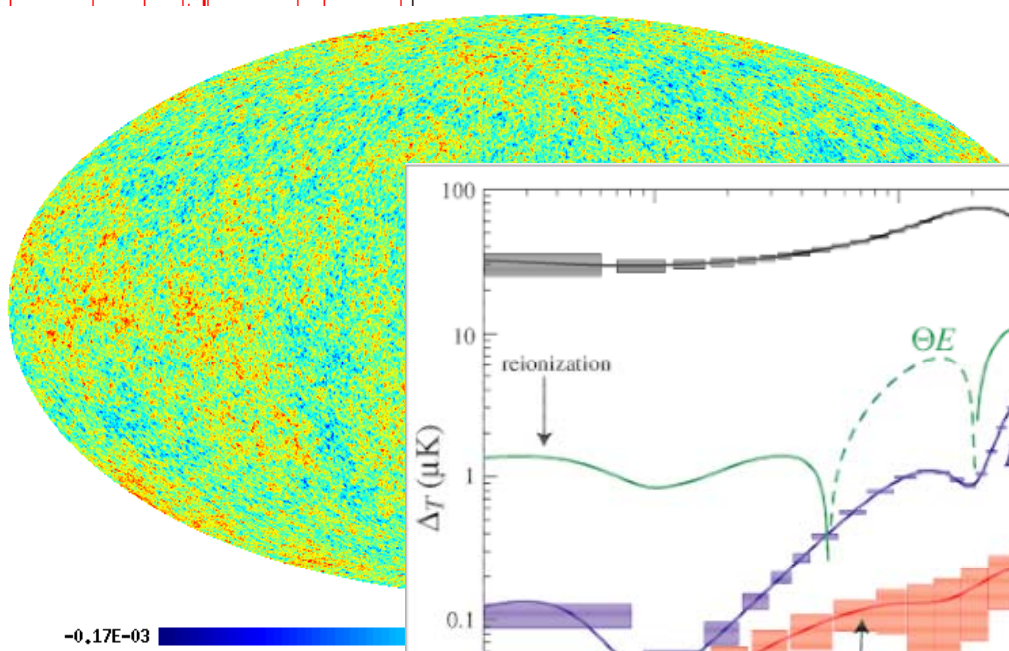
Cosmic *- primordial photons filling all space*

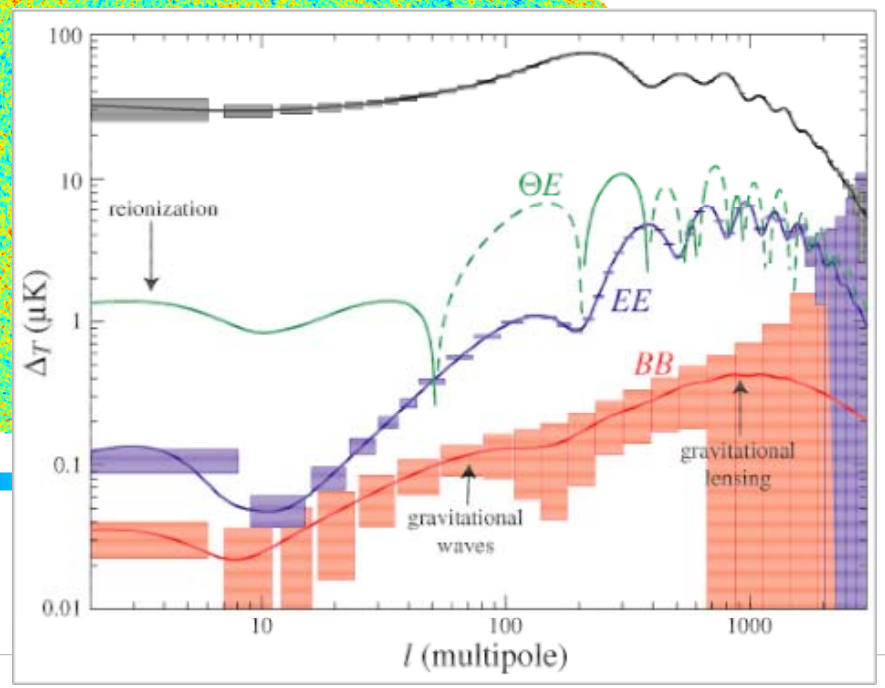
Microwave *- red-shifted by the continued expansion of the Universe from 3000K at last scattering to 3K today*

Background *- coming from "behind" all astrophysical sources.*

CMB Data Analysis







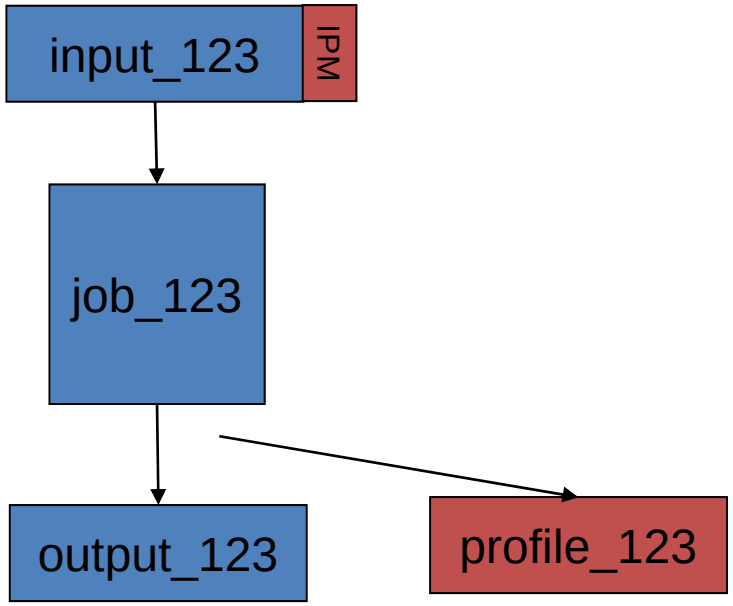
- HEC has therefore l

MADbench2 Overview

- Lightweight version of MADCAP: maximum likelihood CMB angular power spectrum estimation code
  - Unlike most I/O benchmarks, MADbench2 is derived directly from important app
- Benchmark retains complexity and requirements of the full science code
  - Eliminated special-case features, preliminary data checking, etc.
- Out-of-core calculation due to large size/number of matrices pix-pix matrices
  - Holds at most three matrices in memory at any one time
- MADbench2 used for
  - Procuring supercomputers and filesystems
  - Benchmarking and optimizing performance of realistic scientific applications
  - Comparing various computer system architectures
- Computation can be replaced with tunable busy work
  - Allows runs on experimental systems
  - Avoids waiting for large computation runs
  - Allows exploration balance between computation and I/O rate

IPM-I/O

- Extended capabilities of IPM to measure I/O characteristics
- IPM-I/O is an interposition library that wraps I/O calls with tracing instructions
- Intercepts POSIX-I/O calls, records call arguments, duration, time stamp, etc
- Provides a complete picture of I/O performance –
  Including application, middleware, file system



IPM Design Goals

- Provide high level performance profile ✓
- Fixed memory footprint ✓
- Minimal CPU overhead ✓
- Parallel aware ✓
- Easy to use ✓
- Portable ✓
- Recently extended to perform I/O tracing

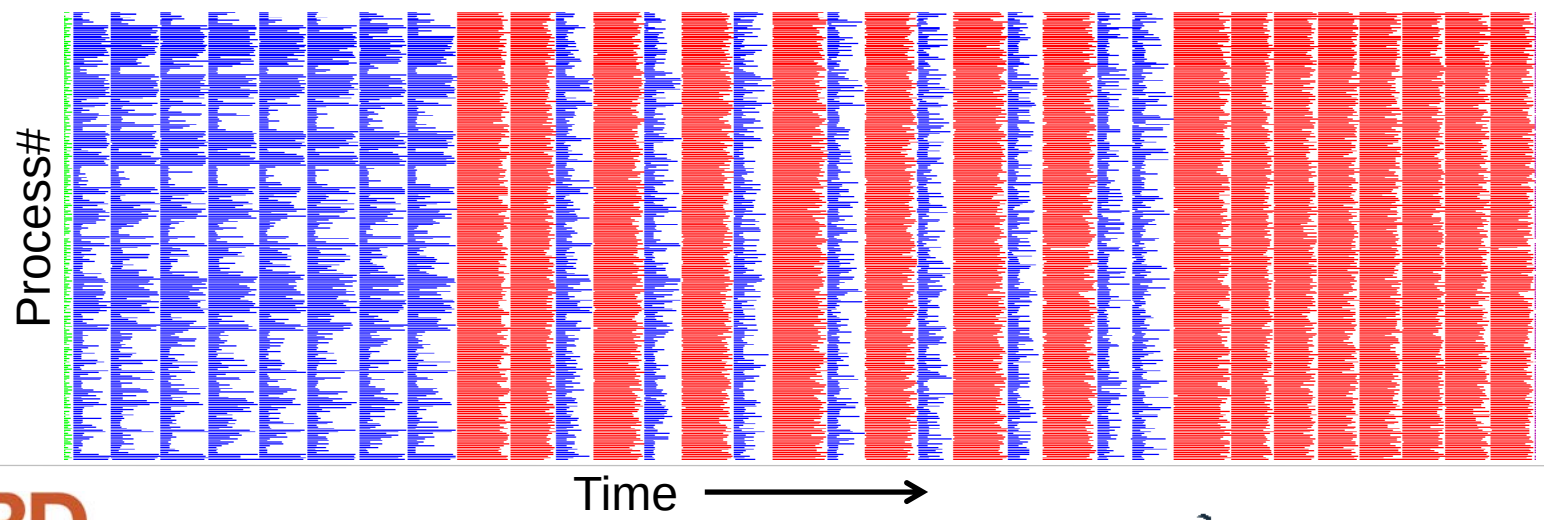
Computational Structure

Derive spectra from sky maps by:

- Compute, Write (Loop): Recursively build sequence of Legendre polynomial based CMB signal pixel-pixel correlation component matrices
- Compute/Communicate: Form and invert CMB signal & noise correlation matrix
- Read, Compute, Write (Loop): Read each CMB component signal matrix, multiply by inverse CMB data correlation matrix, write resulting matrix to disk
- Read, Compute/Communicate (Loop): In turn read each pair of these result matrices and calculate trace of their product

Recast as benchmarking tool: all scientific detail removed, allows varying busy-work component to measure balance between computational method and I/O

- The generic I/O access patterns are applicable to numerous applications

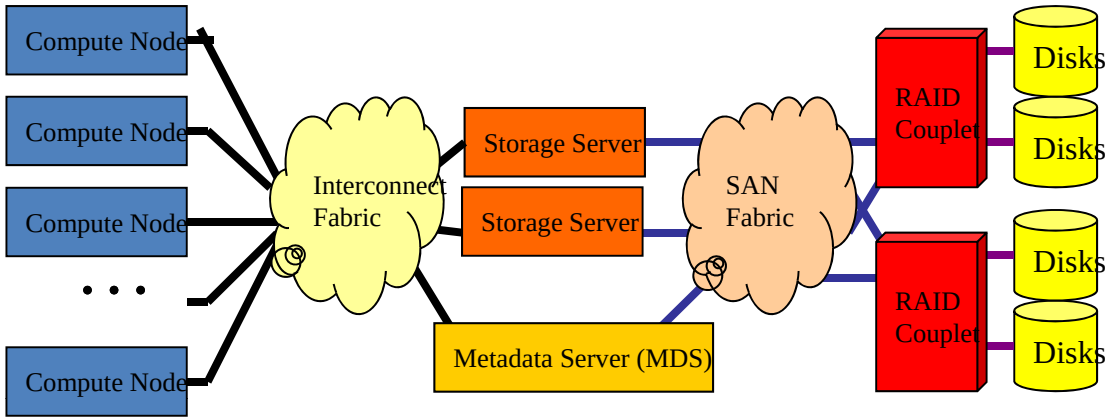


Parallel Filesystem Overview

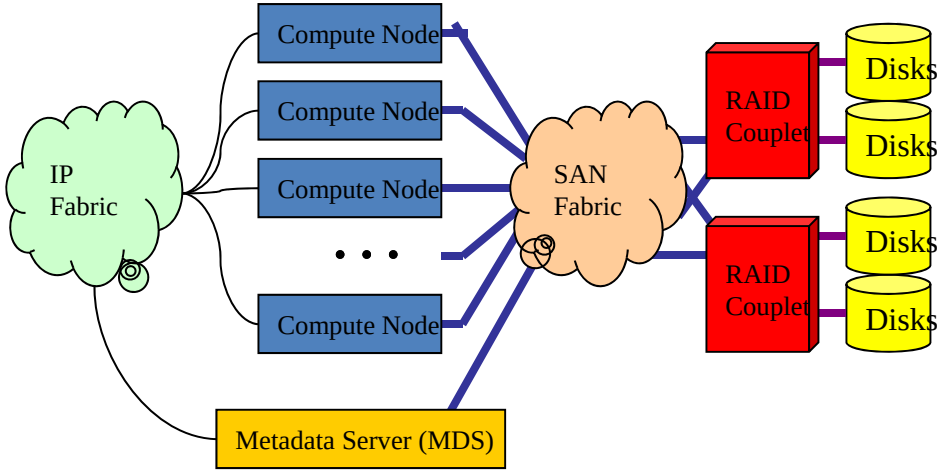
Machine Name	Center	Parallel File System	Proc Arch	Total Procs	Compute to I/O Node	Max Node BW to IO	MPI Node BW (GB/s)	I/O Servers/ Clients	Max Disk BW (GB/s)	Total Disk (TB)
Jaguar XT3	ORNL	Lustre	AMD	10,400	SeaStar-1	6.4	1.2	1:105	22.5	100
Franklin XT4	LBNL	Lustre	AMD	19,320	SeaStar-2	6.4	1.2	1:112	10.2	350
Thunder	LLNL	Lustre	IA64	4,096	Quadrics	0.9	0.4	1:64	6.4	185
Bassi	LBNL	GPFS	Pwr5	976	Federation	8.0	6.1	1:16	6.4	100
Jacquard	LBNL	GPFS	AMD	640	Infiniband	2.0	1.2	1:22	6.4	30
BG/P	ANL	GPFS	PPC	32,768	Tree/10GigE	0.35	0.3	1:64	8	233
BG/P	ANL	PVFS2	PPC	4096	Tree/10GigE	0.2	0.2	1:64	1.0	88
Columbia	NASA	CXFS	IA64	10,240	FC4	1.6	N/A	N/A	1.6	600

- Examine a variety of parallel filesystem implementations. Each with its own balance between computation and I/O performance
- <u>Cost is always a critical metric, but is not examined in our study</u>

Filesystem Schematics



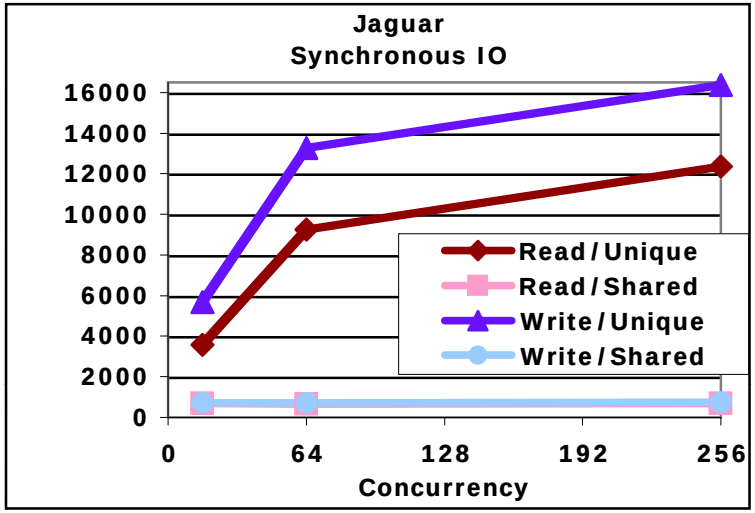
Abstract Schematic
Lustre, GPFS, and PVFS Storage

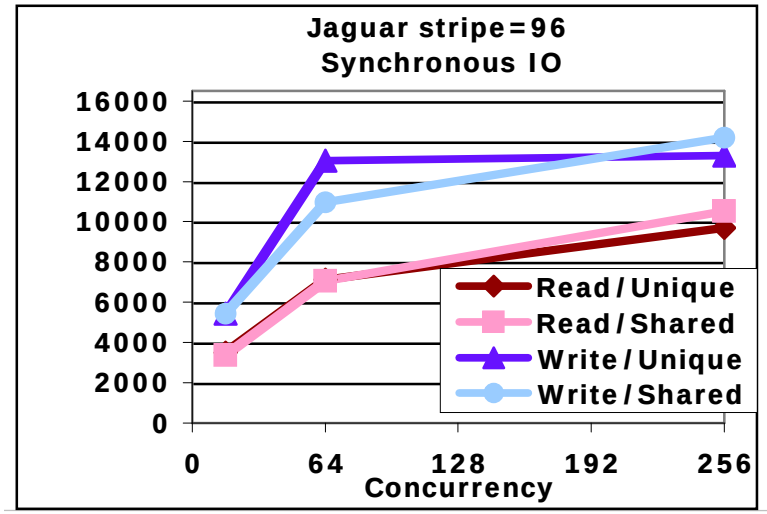


Abstract Schematic
SAN Storage (Columbia): clients connect directly to I/O without storage server. However (MDS) are connected to the clients via GigE.

Jaguar Performance

LUSTRE / AMD / XT3



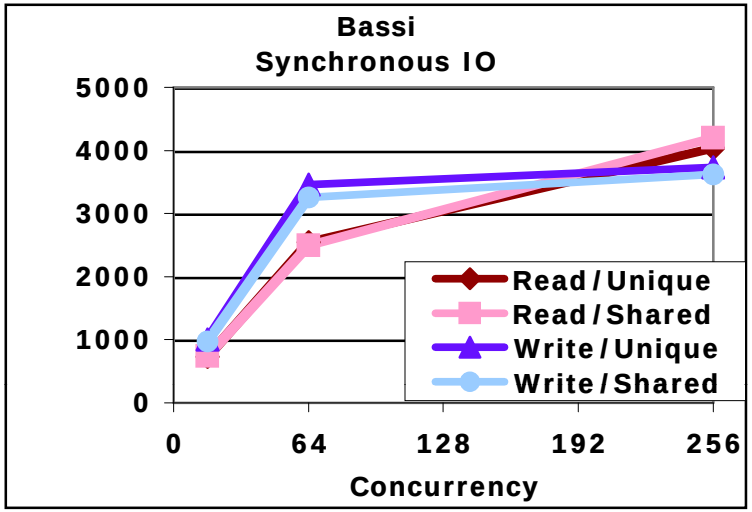


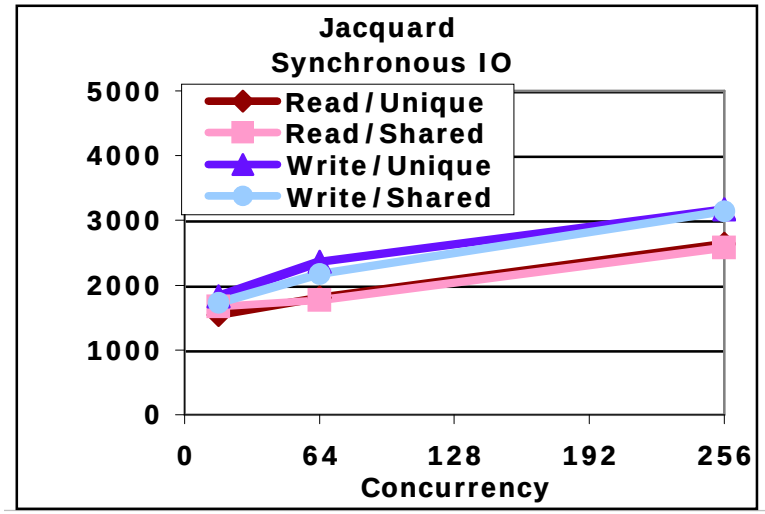
- Highest synchronous *unique* performance
- Reading is slower than writing due to buffering
- Unlike *unique* files, *shared* files performance is uniformly poor:
  - Default I/O traffic only uses 8 of 96 OSTs
- OST restriction allows consistent performance, but limits single job access to full throughput
- Using 96 OSTs (lstripe) allows comparable performance between unique and shared
- Same is true for Franklin
- OST 96 is not default due to:
  - Increase risk job failure
  - Exposes jobs to more I/O interference
  - Reduce performance of *unique* file access

Lustre 5,200 dual-AMD node XT3 @ ORNL Seastar-1 via HyperTransport in 3D Torus Catamount: compute PE, Linux: Pes 48 OSS, 1 MDS, 96 OST, 22.5 GB/s I/O peak

Bassi & Jacquard Performance

GPFS / Power5 & AMD / Federation & IB



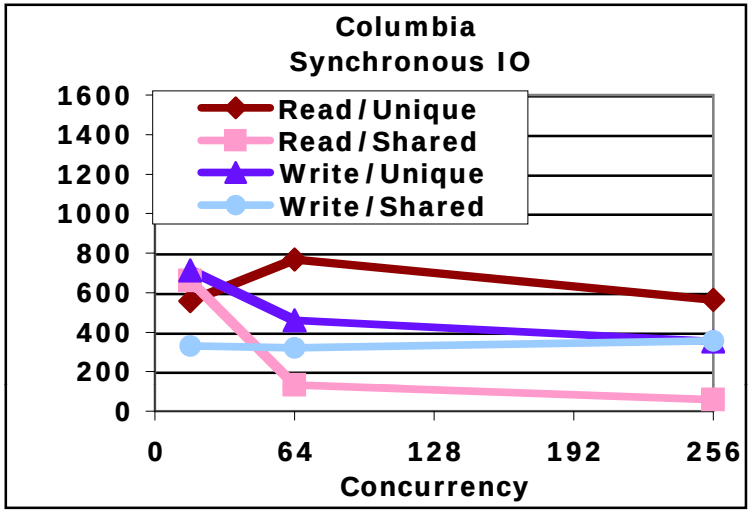


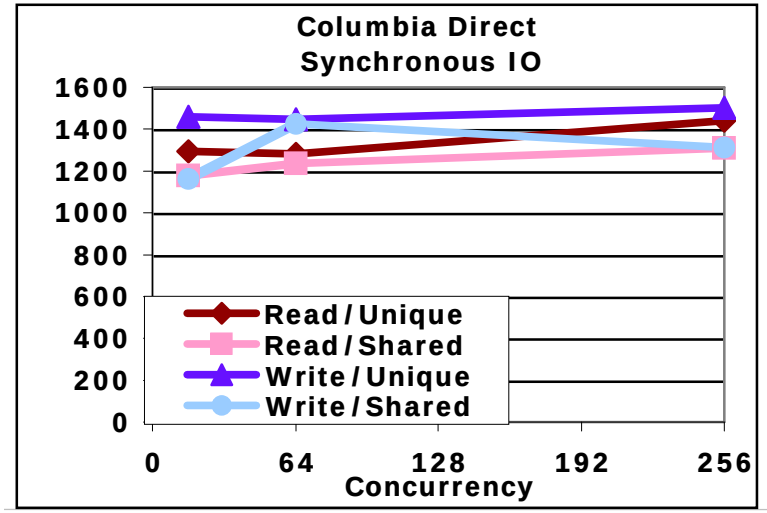
- Unlike Lustre, Bassi and Jacquard's attain similar *shared* and *unique* performance
  - with no special optimization
- Bassi quickly saturates I/O due to high BW node to I/O interconnect
- Bassi higher read I/O could be prefetching
- Jacquard continues to scale at 256 indicating that GPFS NFS has not been saturated
- Bassi outperforms Jacquard due to superior node to I/O BW (8 vs 2 GB/s)

Bassi GPFS 122 8-way Power5, AIX, Federation, fat-tree
6 VSD, 16 FC links, 6.4 GB/s peak @ LBNL
Jacquard GPFS 320 dual-AMD, Linux, Infiniband, fat-tree
IB4X,12x (leaves, spine), peak 4.2 GB/s (IP over IB) @ LBNL

Columbia Performance

CXFS / Itanium2 / Altix3700 / NUMAlink3

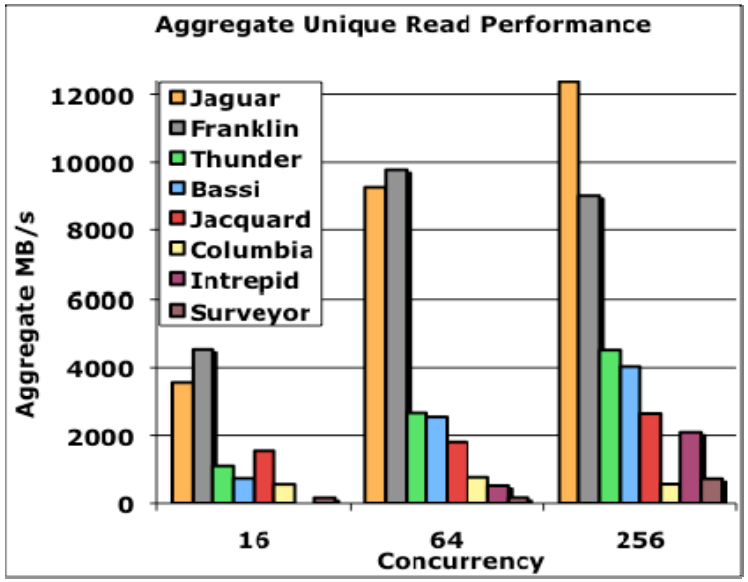


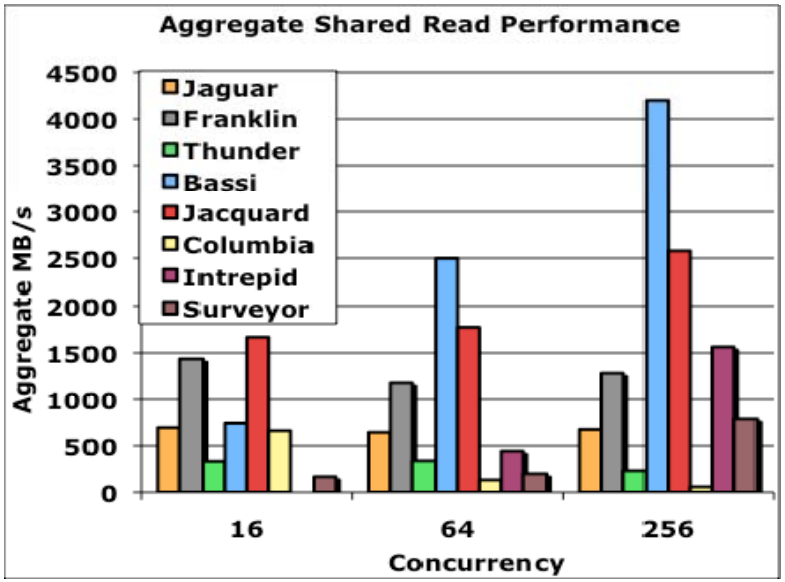


- Default I/O rate relatively poor
- With increasing concurrency:
  - Higher lock overhead (access buffer cache)
  - More contention to I/O subsystem
  - Potentially reduced coherence of I/O request
- DirectIO significantly improves I/O
  - Bypasses block-buffer cache
  - Prevents block buffer cache reuse
  - Complicates I/O, transaction must be block-aligned
  - Has restrictions on memory alignment
  - Forces programming in disk-block sized I/O as opposed to arbitrary size POSIX I/O
- Columbia CCNUMA also offers option of using idle procs for I/O buffering

CXFS, 20 Altix3700, 512-way IA64 @ NASA
10,240 procs, Linux, NUMAlink3, no storage server layer
3MDS via GigE, max 4 FC4, peak 1.6 GB/s

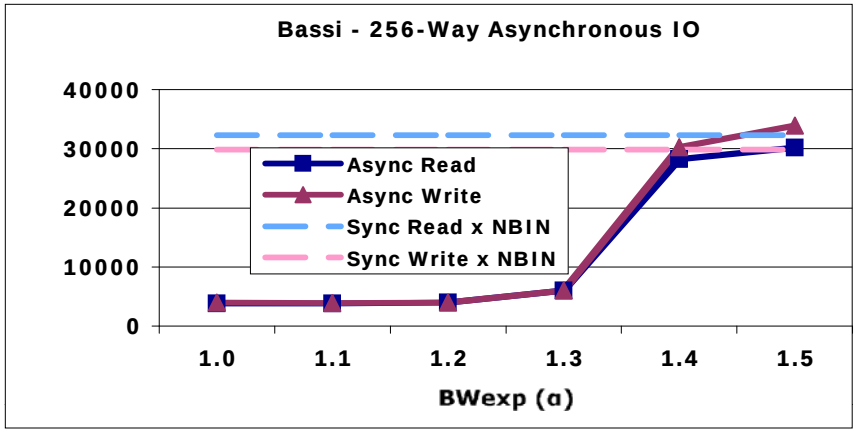
MADBench Synchronous Performance

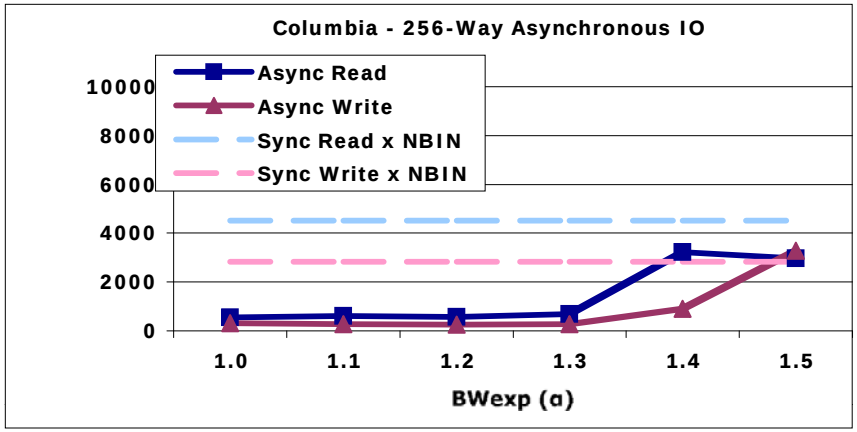




- For *unique* read/write Jaguar/Franklin attain highest performance
- For default *shared* read/write Bassi shows highest I/O rate
- It is possible to achieve similar behavior between shared and unique file access
  Default for all systems except Lustre which required trivial mod
- Almost all systems are close to I/O saturation at P=256
- Performance variation between systems up to 75x
- Future will explore larger concurrencies, MDS may reduce performance at higher P

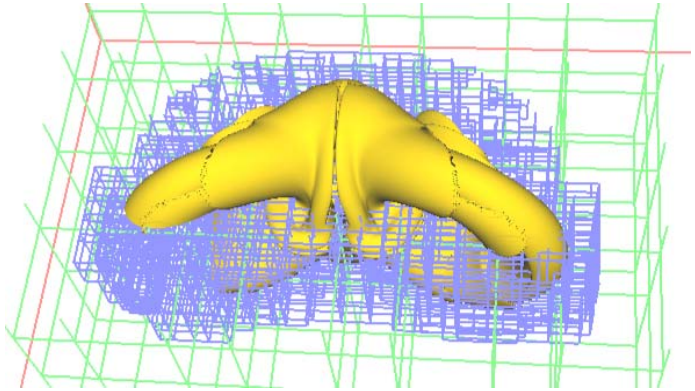
Asynchronous Performance



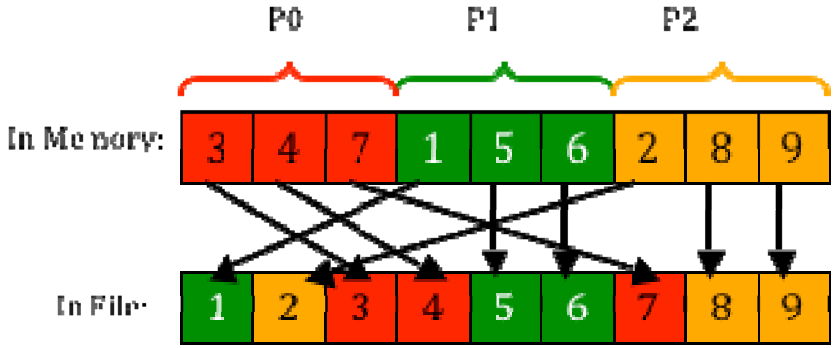


- Possible to hide I/O behind calc via MPI-2
- Only Bassi and Columbia (out of 9) support fully asynchronous MPI-I/O
- Develop busy-work exponent α, corresponds to $O(N^{\alpha})$ flops
- Bassi and Columbia improve I/O by almost 8x for high α (peak improvement)
  - Bassi now shows 2x the performance of Jaguar
  - As expected small α reproduced synchronous behavior
- Critical value for transition is α between 1.3-1.4 ie algorithms > $O(N^{2.6})$
- If comp-I/O balance continues to decline, effective α to hide I/O will increase: However, we are quickly approaching the practical limit of BLAS3 complexity

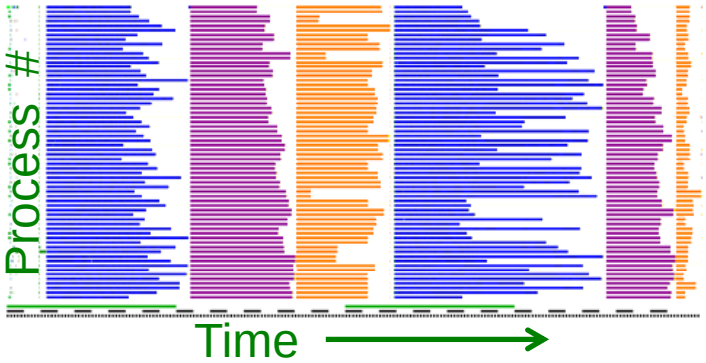
CHOMBO I/O Optimization



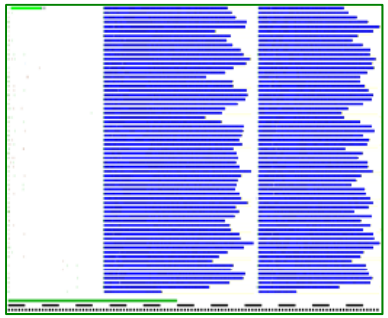
Chombo AMR vorticity-magnitude isosurface



Mapping logical AMR blocks & file blocks



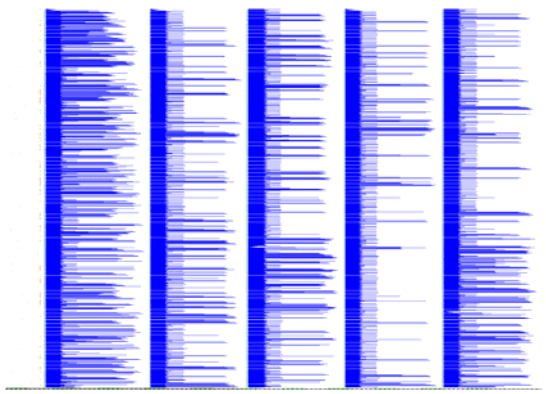
Original Chombo Trace on Franklin

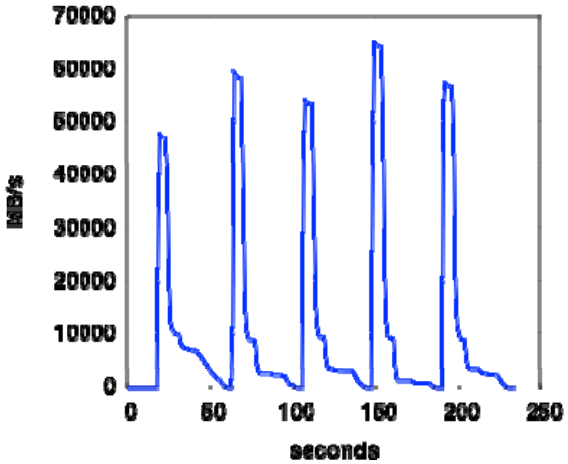


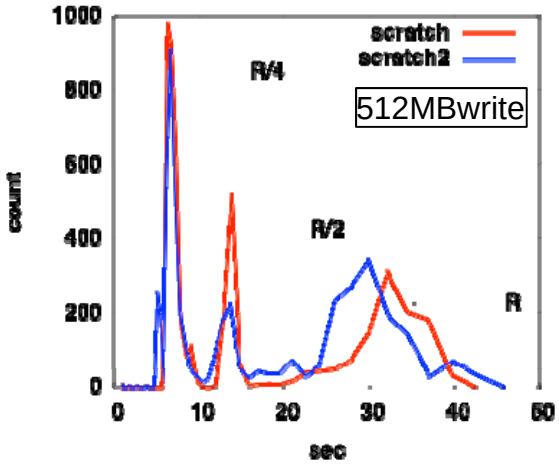
After modified HDF5: removal of *ftruncate* and *fsync*: 2x speedup

Lightweight, easy to use tools are essential for diagnosing these issues

Performance Events to Ensembles

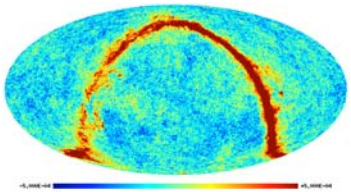




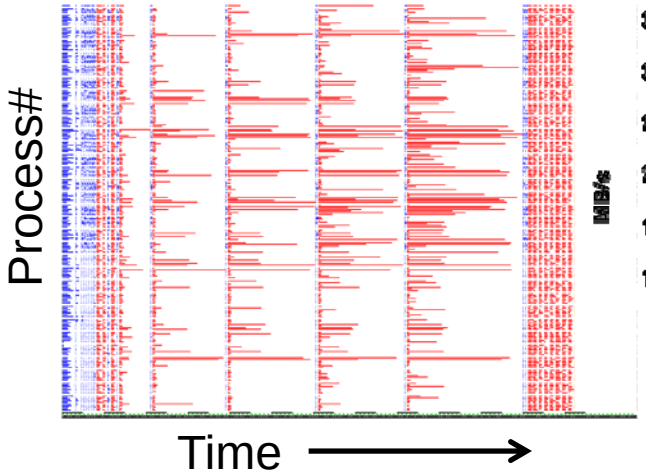


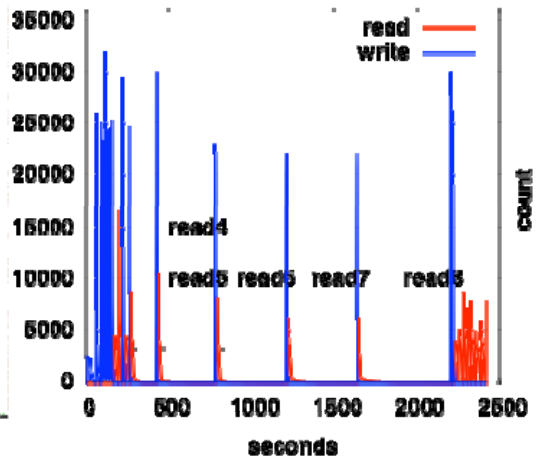
Transition from mechanistic analysis of isolated events to analysis of ensembles resembles the strategy of statistical physics - whereby large numbers of interacting systems can be described by the properties of their ensemble distributions such as moments, splitting and line-widths

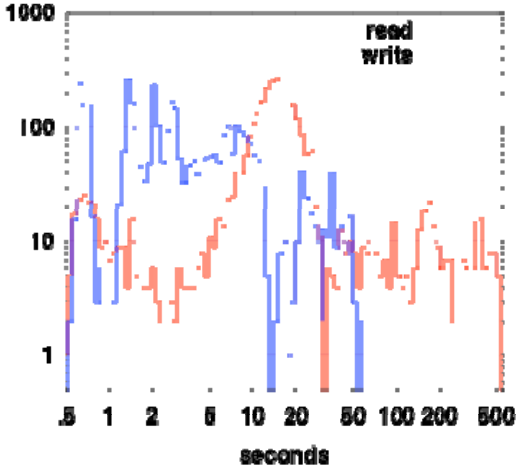
MADbench I/O Optimization

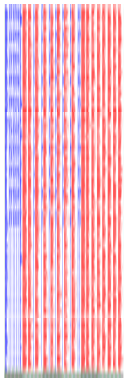


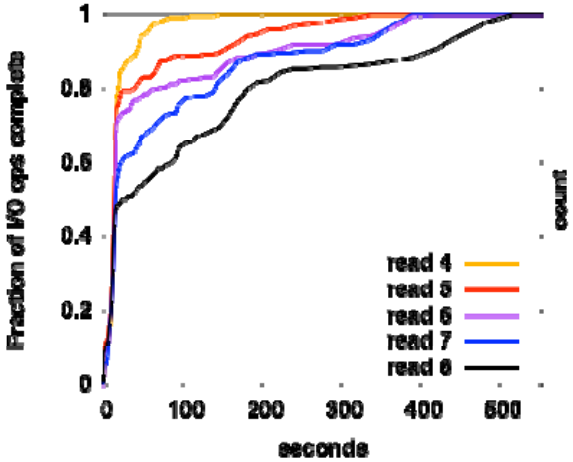
Before

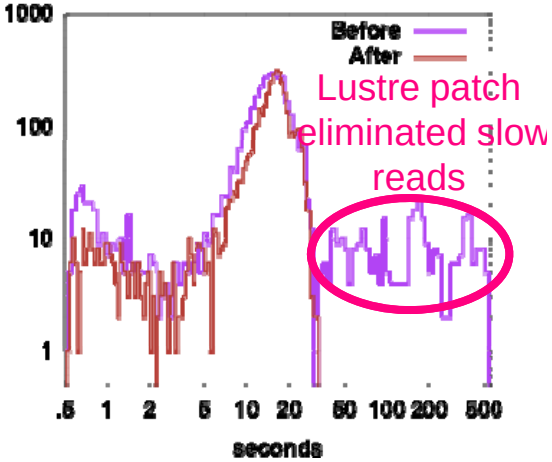












Statistical methods revealed a subtle Lustre system software bug relating to erroneous read-ahead buffer. <u>Patch improved performance by >4x</u>

Global Cloud System Resolving Climate Modeling

Individual cloud physics fairly well understood

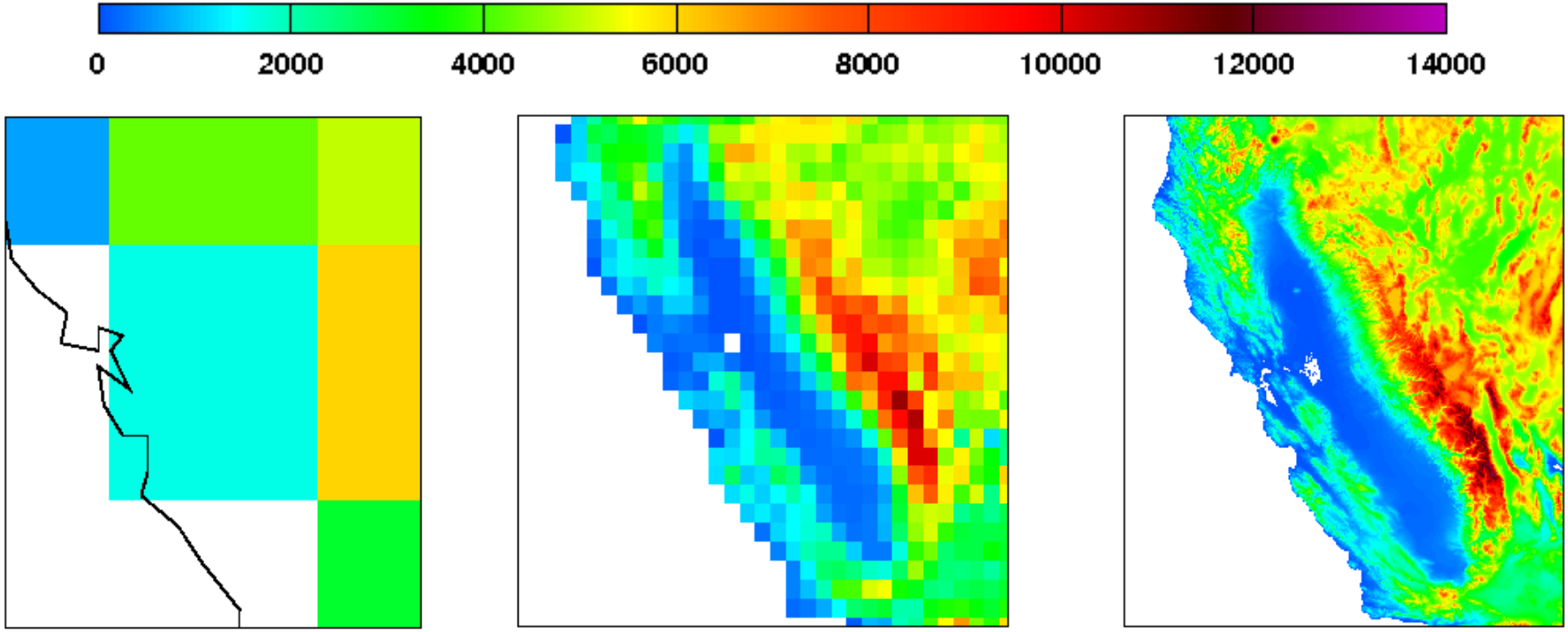
Parameterization of mesoscale cloud statistics performs poorly.

Direct simulation of cloud systems in global models requires exascale

- Cloud statistical parameterization is a leading source of errors in climate modeling
  - Impacts solar and terrestrial radiation, precipitation, etc
- Currently cloud systems are much smaller than model grid cells (unresolved)
- Direct cloud system simulation: top priority by the 1st UN WMO Modeling Summit.

High Resolution GCRM

Surface Altitude (feet)

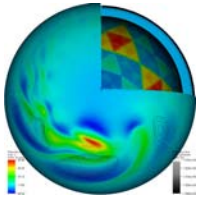


200km
Typical resolution of IPCC AR4 models

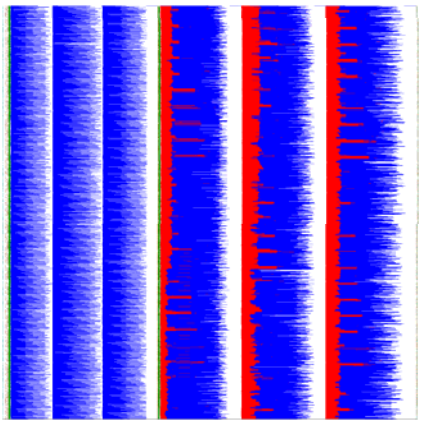
25km
Upper limit of climate models with cloud parameterizations

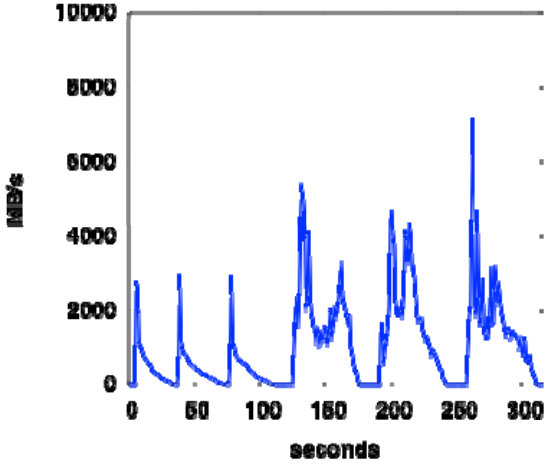
1km
Cloud system resolving models are a transformational change

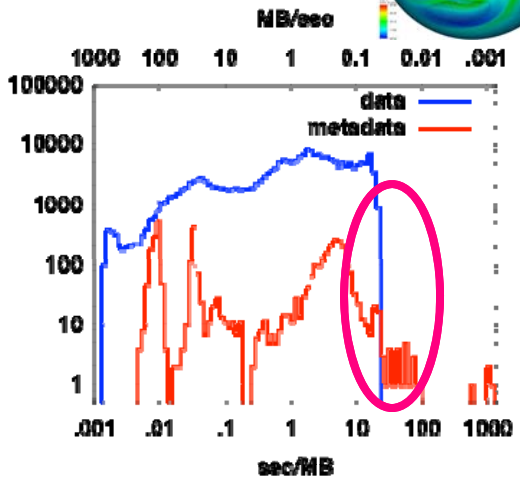
GCRM I/O Optimization

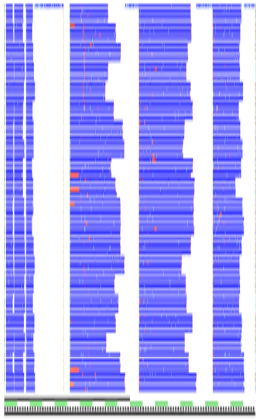


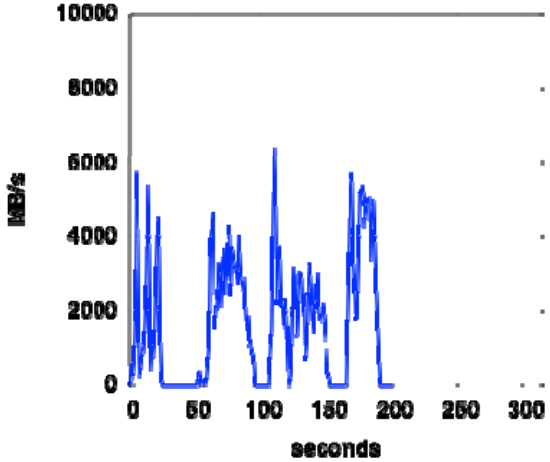
Before

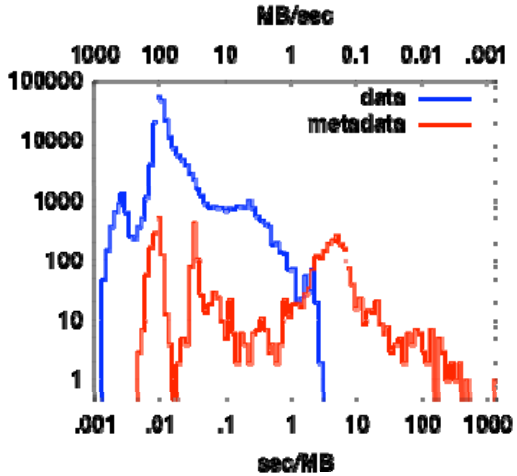






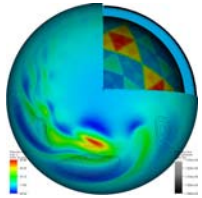




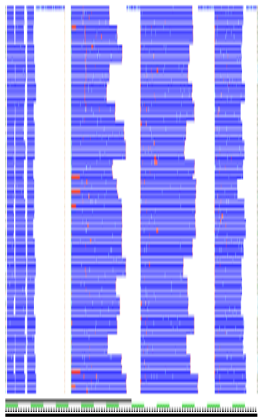


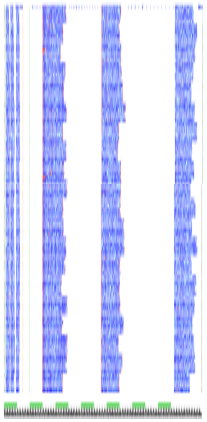
Collecting buffering – aggregating data from 10,240 to 80 task improves performance by <u>60%</u> (reduced contention, I/O server queue depth, etc)

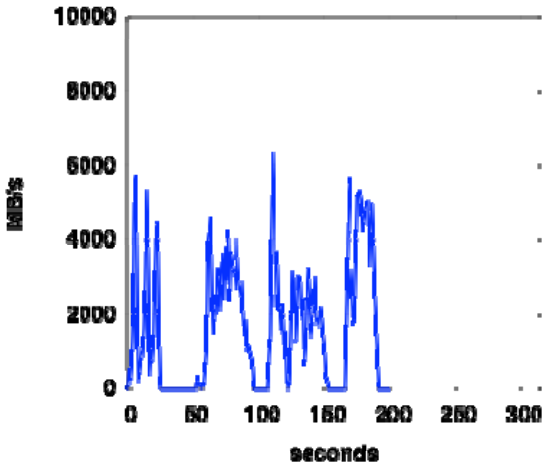
GCRM I/O Optimization

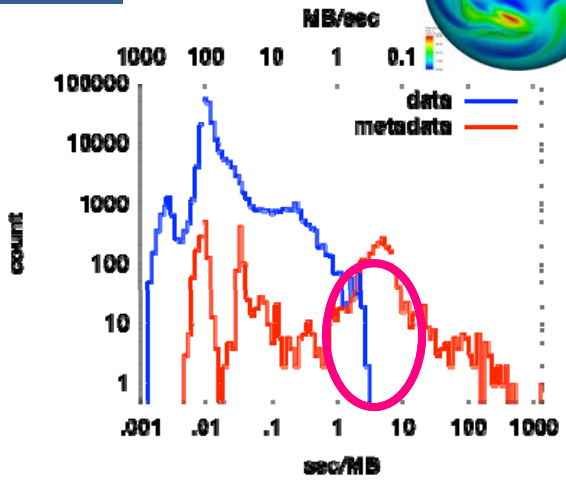


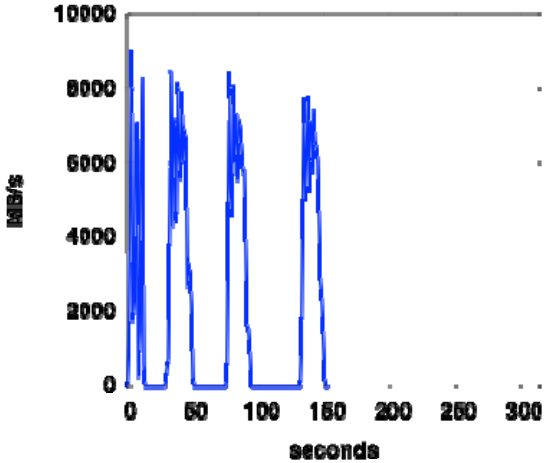
Before

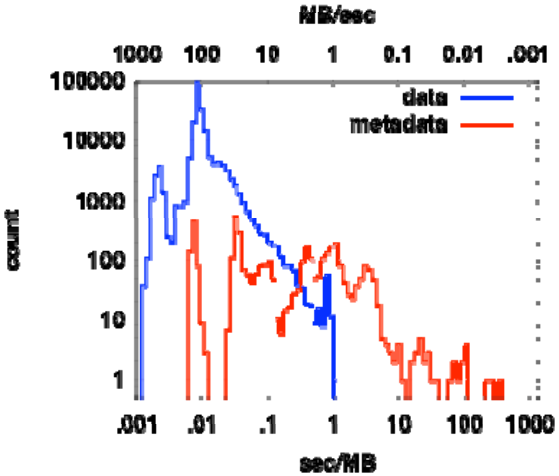






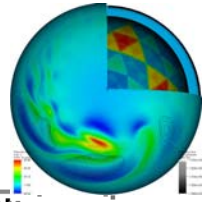




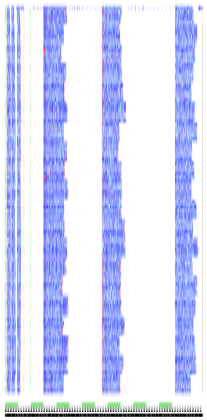


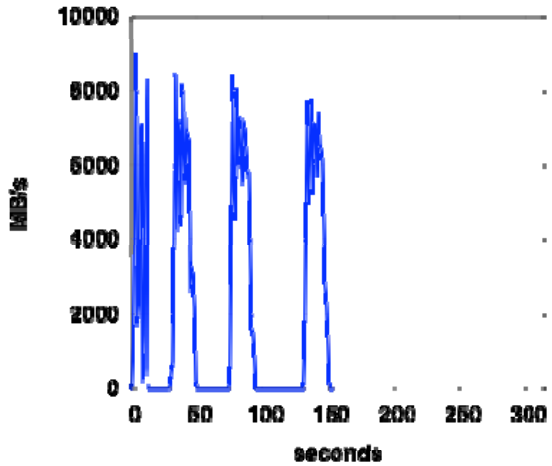
Using HDF5 library calls, padded and aligned writes to 1MB boundary
Worst case per-task rate is now 1MB/sec, also improves metadata
Overall improvement 50% reduction compared with original

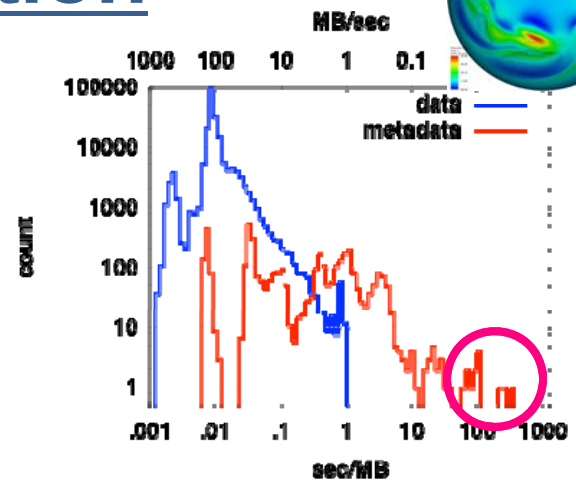
GCRM I/O Optimization

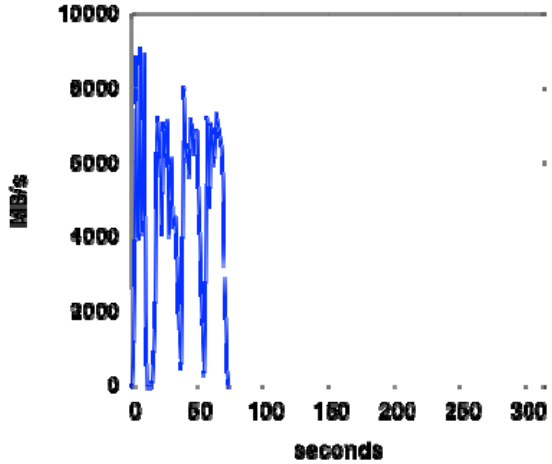


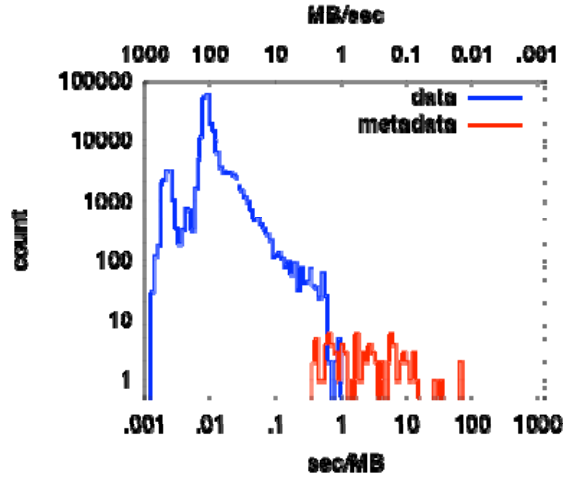
Before











Aggregate metadata <3KB writes into single 1MB write, deferred till file close
Removes large gaps caused by serialized writing on task0
Total runtime decreased by a total of 4x over baseline

Conclusions and Future Work

- Critical to quantify scientific I/O behavior on leading HPC systems
- **I/O bottleneck source may in application code, middleware library, filesystem, underlying architecture — or some combination thereof**
- Developed IPM I/O: a scalable, portable, lightweight framework for collecting, profiling, and aggregating HPC performance information.
- **Individual I/O events vary widely making bottleneck isolation difficult**
- Statistical methods give useful diagnostic insights into large datasets
  - Provides opportunities to improve I/O behavior
- **In the future, IPM will gather statistical information directly, improving scalability by reducing the data volume.**
- Future work: build automatic recognition of model and moments, allowing IPM to signal underlying system software with useful hints.

Acknowledgements

Julian Borrill
Mark Howison
Karen Karavanic (PSU)
Noel Keen
John Shalf
Hongzhang Shan
David Skinner
Andrew Uselton
Nicholas Wright

This work was funded in part by the Advanced Scientific Computing Research (ASCR) in the DOE Office of Science under contract number DE-C02-05CH11231

EXTRA SLIDES

MADbench2 Parameters

Environment Variables:

IOMETHOD - either POSIX of MPI-IO data transfers

IOMODE - either synchronous or asyncronous

FILETYPE - either unique (1 file per proc) or shared (1 file for all procs)

BWEXP - the busy work exponent α

Command-Line Arguments:

NPIX - number of pixels (matrix size)

NBIN - number of bins (matrix count)

BScaLAPCK - ScaLAPACK blocksize

FBLOCKSIZE - file Blocksize

MOD_{RW} - IO concurrency control (only 1 MOD_{Rw} procs does IO simultaneously)

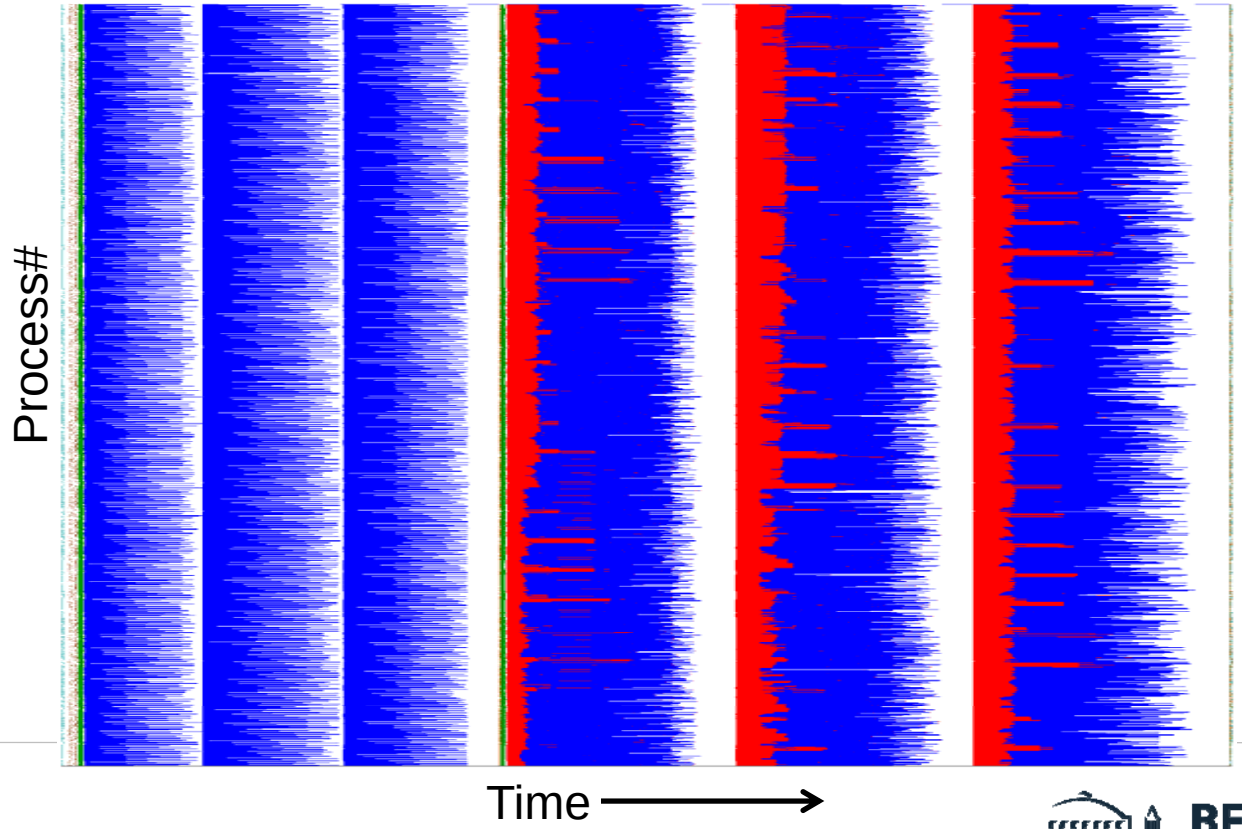
CPU	BG/L CPU	NPIX	NBIN	Mem (GB)	DISK (GB)
---	16	12,500	8	6	9
16	64	25,000	8	23	37
64	256	50,000	8	93	149
256	---	100,000	8	373	596

Events to Ensembles

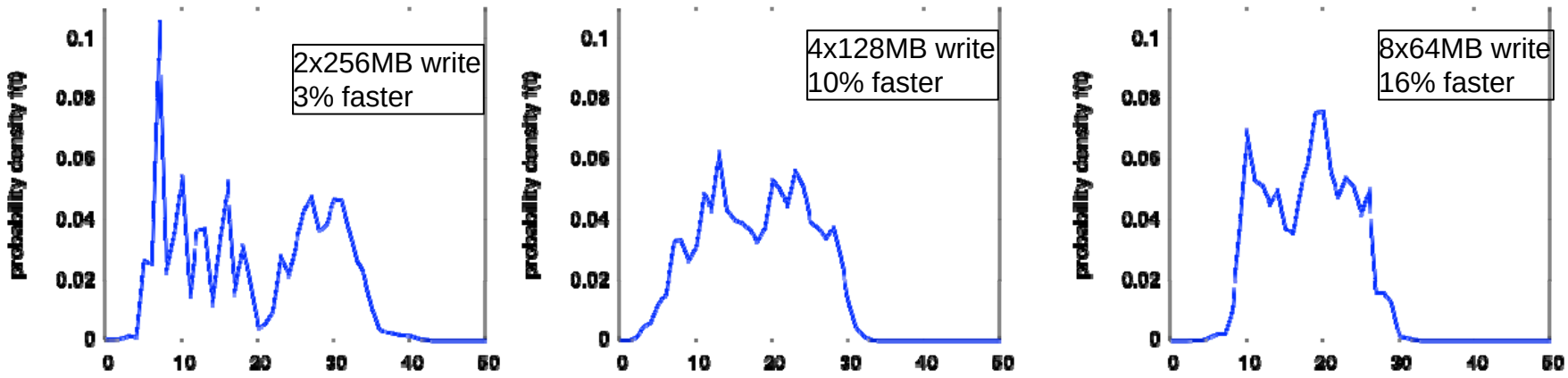
However, examining individual trace events is often insufficient

- Petascale I/O requires new techniques for analysis, visualization, diagnosis
- I/O that proceeds in phase/barrier is vulnerable to a one slow I/O event
- Data volume/variability of I/O events makes it difficult to work with directly

Statistical methods can be revealing

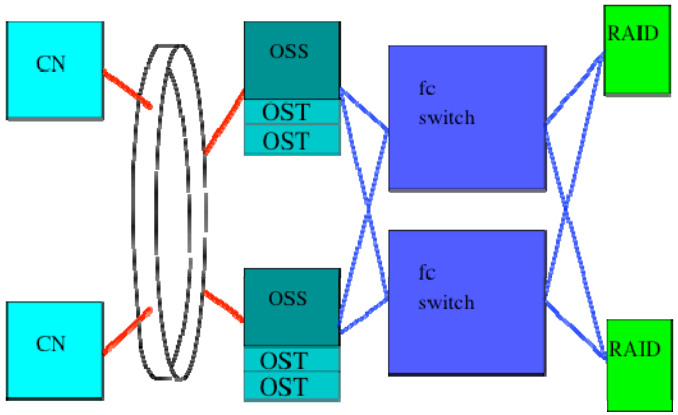


Ensembles: Law of Large Numbers

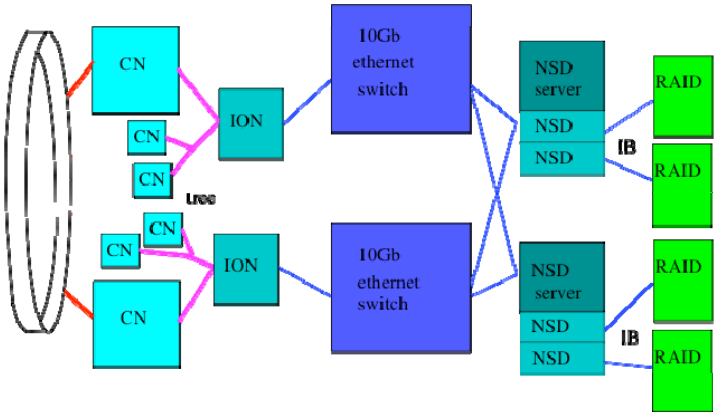


Transition from mechanistic analysis of isolated events to analysis of ensembles resembles the strategy of statistical physics - whereby large numbers of interacting systems can be described by the properties of their ensemble distributions such as moments, splitting and line-widths

Interconnect Overview



Frequently I/O nodes are connected directly to the interconnect



BG/P I/O nodes at the root of independent tree network connected to server via 10GigE