Storage Architectures and Abstractions for Exascale Systems

Rob Ross, Pete Beckman, Phil Carns, Jason Cope, Kevin Harms, Kamil Iskra, Dries Kimpe, Rob Latham, Rusty Lusk, Tom Peterka, Katherine Riley, Seung Woo Son, Rajeev Thakur, Venkat Vishwanath, and Justin Wozniak

Mathematics and Computer Science Division

Argonne National Laboratory

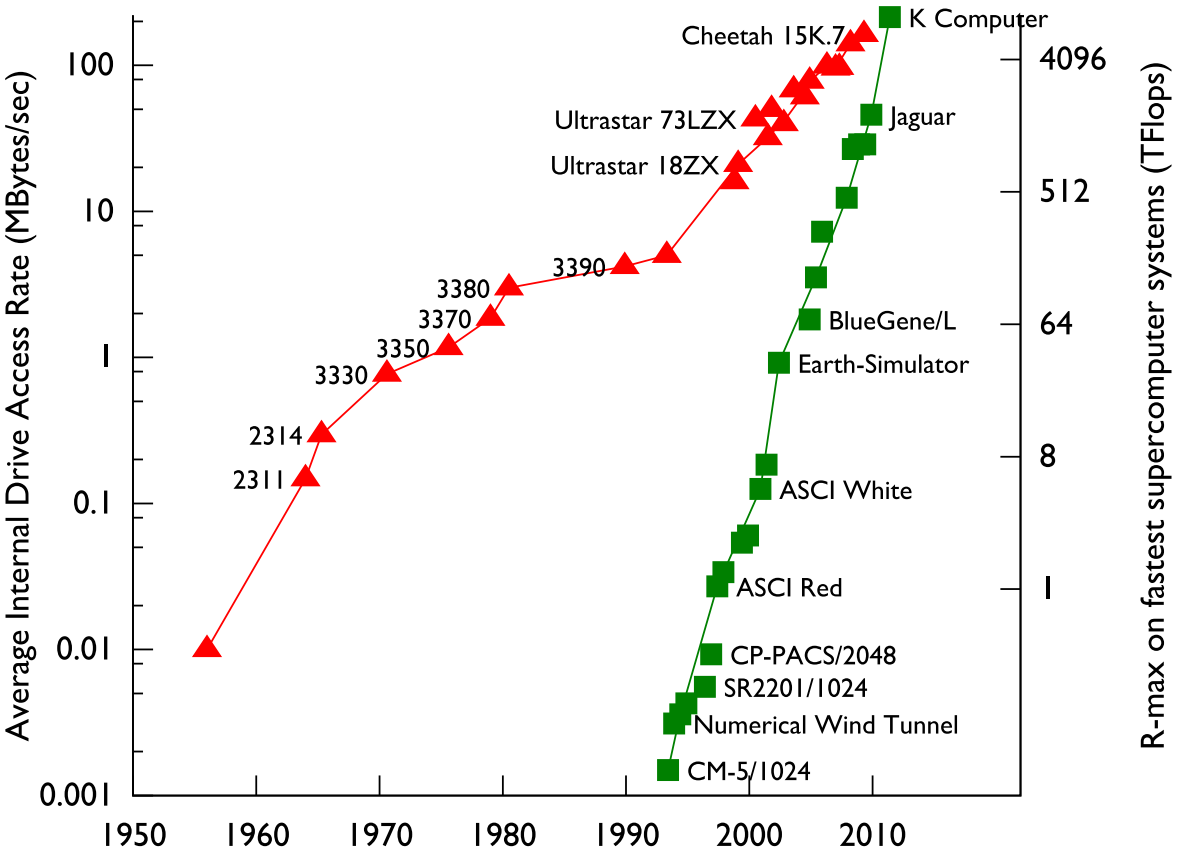
rross@mcs.anl.gov

Complexity and Scale

Complexity and Scale of Hardware Deployments

Data volumes and rates are resulting in storage systems that must serve unprecedented numbers of clients and incorporate massive numbers of devices with very different performance, capacity, and reliability traits.

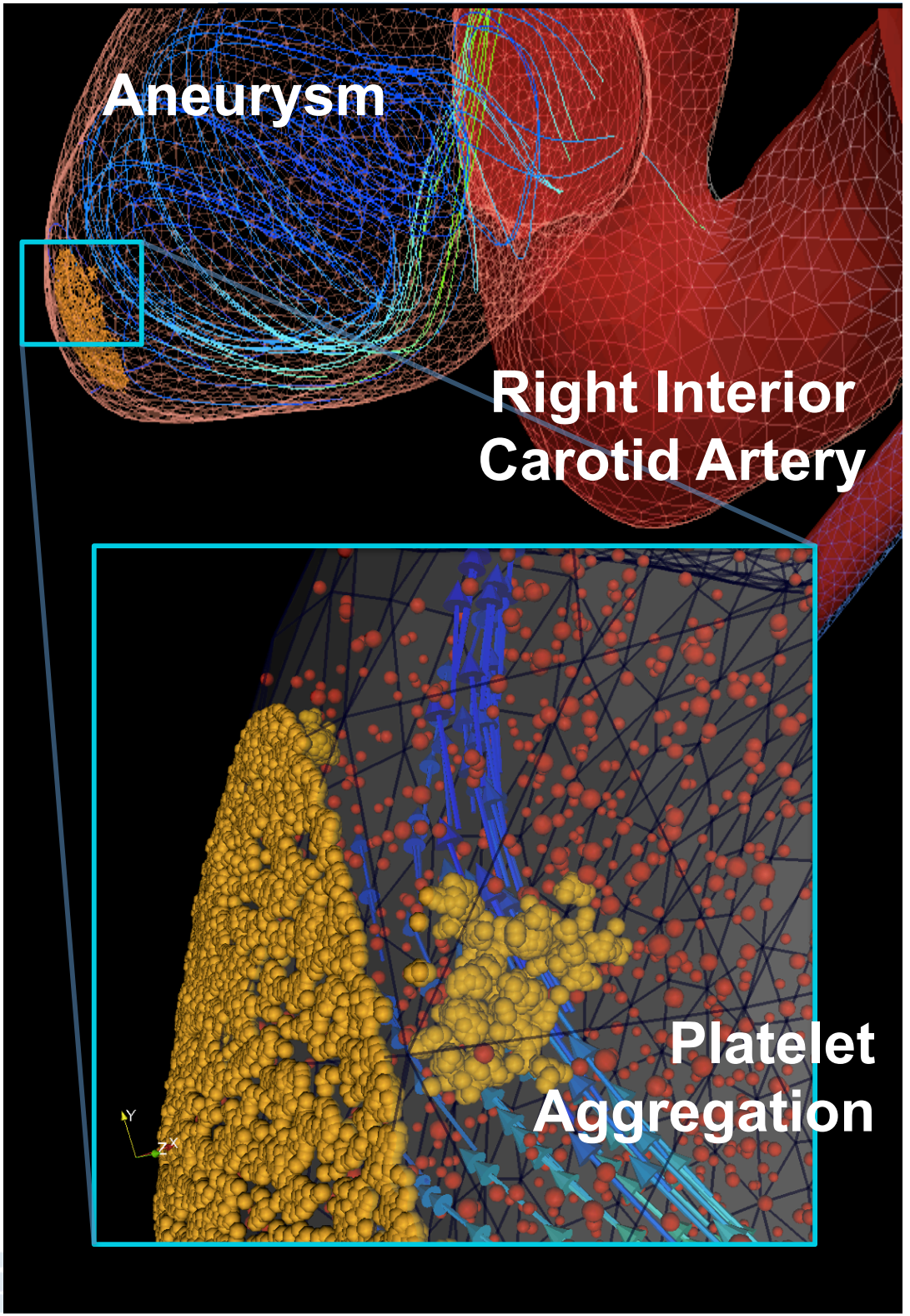
- Trajectory of disk access rate improvements has led to more disks at each HPC system generation
- Projections indicate disk-only storage for exascale would require ~175K disks and cost ~$200M
- NVRAM helps, but existing software not well suited to heterogeneous components
- No good archive story…



Thanks to Richard Freitas of IBM Almaden Research for providing historical drive data.

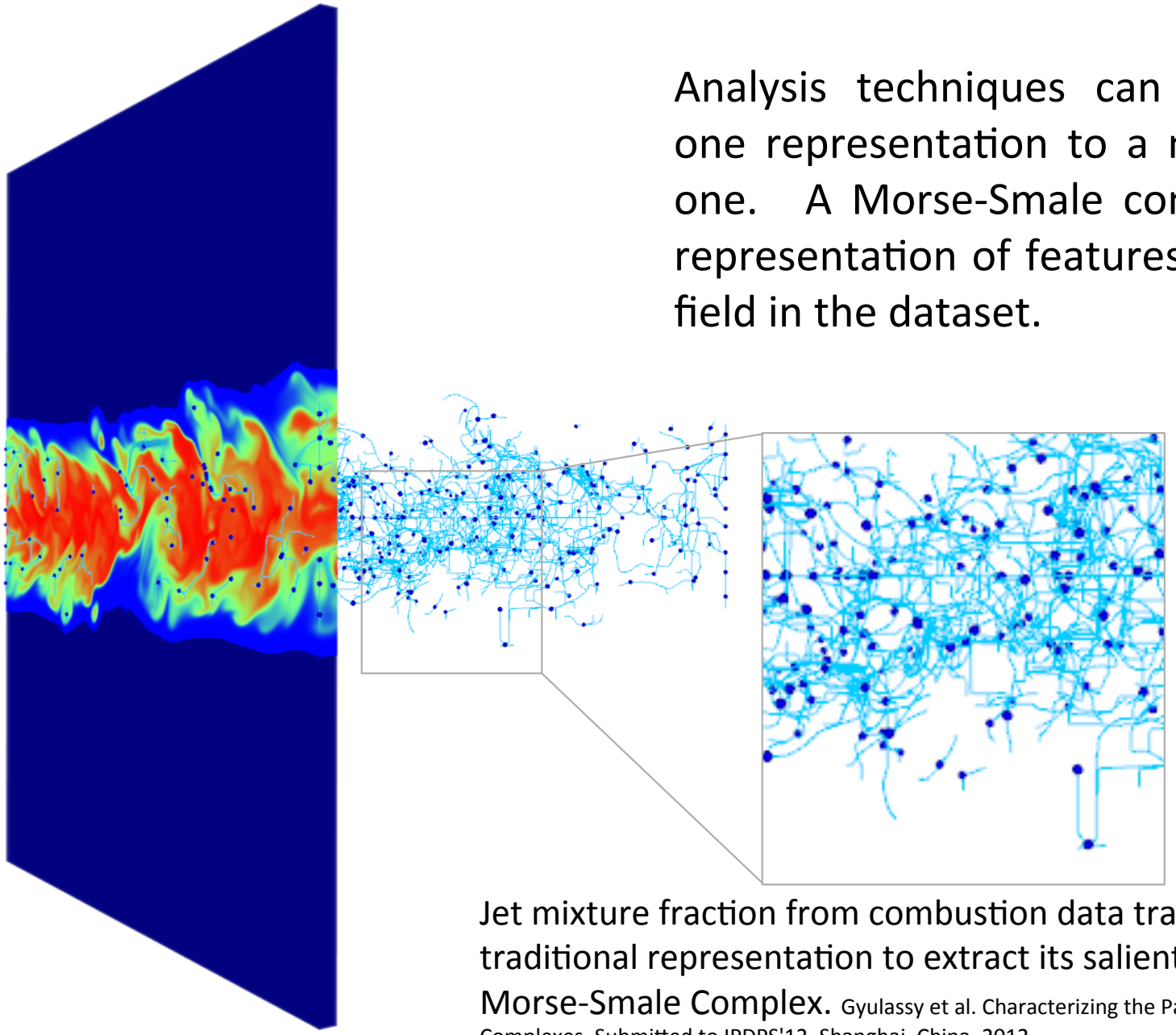
Dataset Complexity

- Complexity as an artifact of science problems and codes:
  - Coupled multi-scale simulations generate multi-component dataset.
  - Atomistic data representations for plasma, red blood cells, and platelets from MD simulation.
  - Field data for ensemble average solution generated by spectral element method hydrodynamics code [Grinberg 2011, Insley 2011]



Thanks to M. Hereld (ANL) for this slide.

Complexity in Analysis



Analysis techniques can transform from one representation to a more meaningful one. A Morse-Smale complex is a graph representation of features in the gradient field in the dataset.

Jet mixture fraction from combustion data transformed from its traditional representation to extract its salient features via the Morse-Smale Complex. Gyulassy et al. Characterizing the Parallel Computation of Morse-Smale Complexes. Submitted to IPDPS'12, Shanghai, China, 2012.

The Need for Revolution in Data Storage

The HPC data storage infrastructure developed over the past two decades needs to be replaced.

- Assumes that faults (transient and persistent) will be rare
  - Algorithms/approaches for managing faults are not scalable (e.g., heartbeat)
  - Service degradation is not graceful
  - Many faults must be handled by (expensive) hardware
- Makes poor use of available and upcoming storage technologies
  - Assumes uniform performance from devices
  - Unaware of underlying resource locations
- Presents a cumbersome model for building scientific storage
  - Data layout is obfuscated, making locality difficult to exploit
  - Limited ability to describe relationships between datasets (i.e., directory tree)
  - Concurrency control (e.g., block/page locking) unrelated to user constructs
- Interfaces between storage software layers limit knowledge of behavior
  - Prevents many classes of optimization

The Need for Revolution in Data Storage

The HPC data storage infrastructure developed over the past two decades needs to be replaced.

- Assumes that faults (transient and persistent) will be rare
  - Algorithms/approaches for managing faults are not scalable (e.g., heartbeat)
  - Service degradation is not graceful
  - Many faults must be handled by (expensive) hardware
- Makes poor use of available and upcoming storage technologies
  - Assumes uniform performance from devices
  - Unaware of underlying resource locations

Architectural Mismatch

- Presents a cumbersome model for building scientific storage
  - Data layout is obfuscated, making locality difficult to exploit
  - Limited ability to describe relationships between datasets (i.e., directory tree)
  - Concurrency control (e.g., block/page locking) unrelated to user constructs
- Interfaces between storage software layers limit knowledge of behavior
  - Prevents many classes of optimization

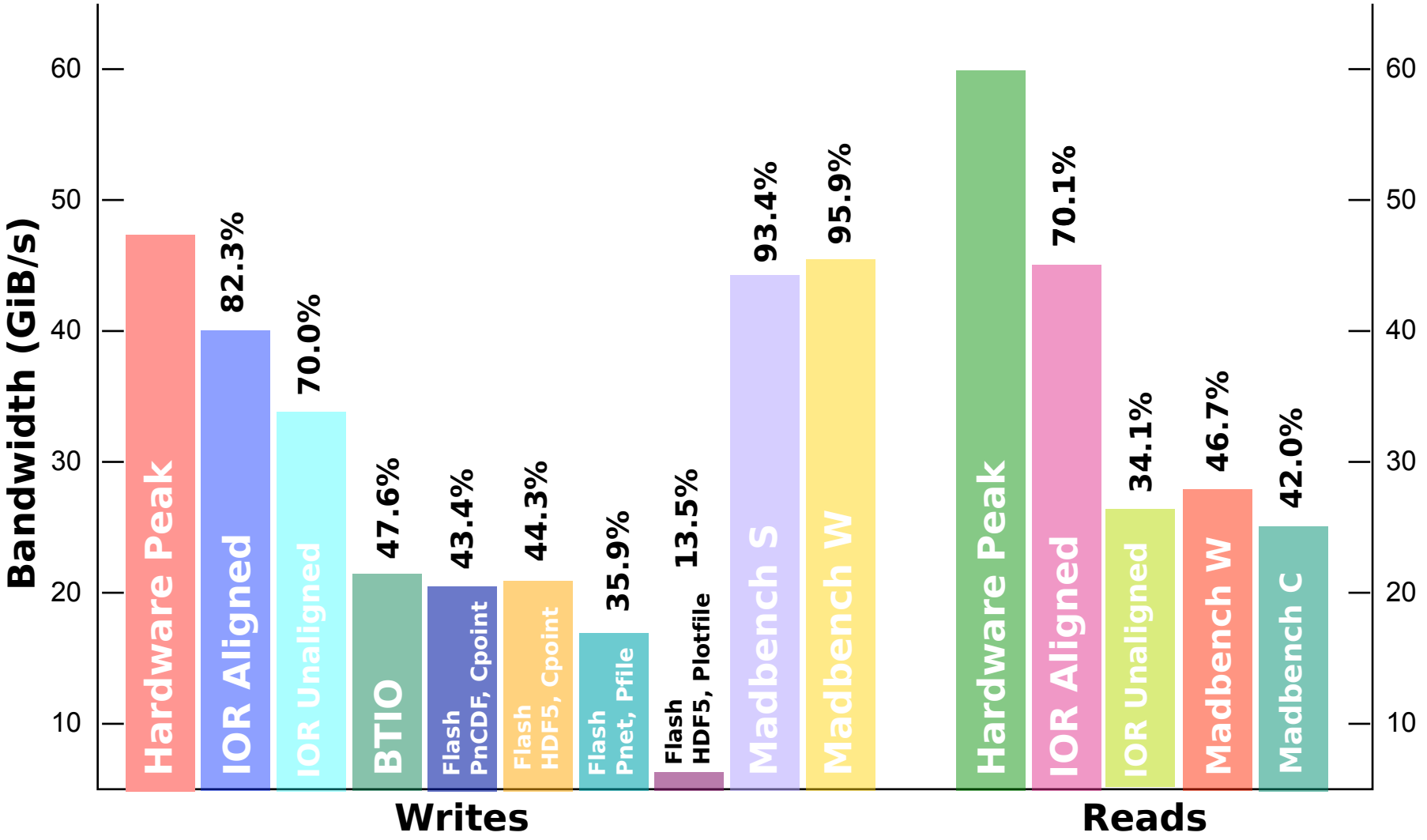
Abstraction Mismatch

R&D Activities in Anticipation of Exascale

- The “exascale picture” is still fuzzy
  - Hardware options for data storage are rapidly changing
  - Failure characteristics of systems are unclear
  - Application drivers could change
- Storage systems take many years to develop and mature, so we must begin making progress now
- Three activities we can engage in today (and are engaged in):
  - Better understanding and tracking application I/O behavior
  - Developing tools to explore the storage system design space (architecture)
  - Building better support for computational science data models (abstraction)

Understanding I/O Behavior

I/O Benchmarks on ALCF Blue Gene/P



See [Lang 2009] for more details.

Characterizing Application I/O

How are are applications using the I/O system, and how successful are they at attaining high performance?

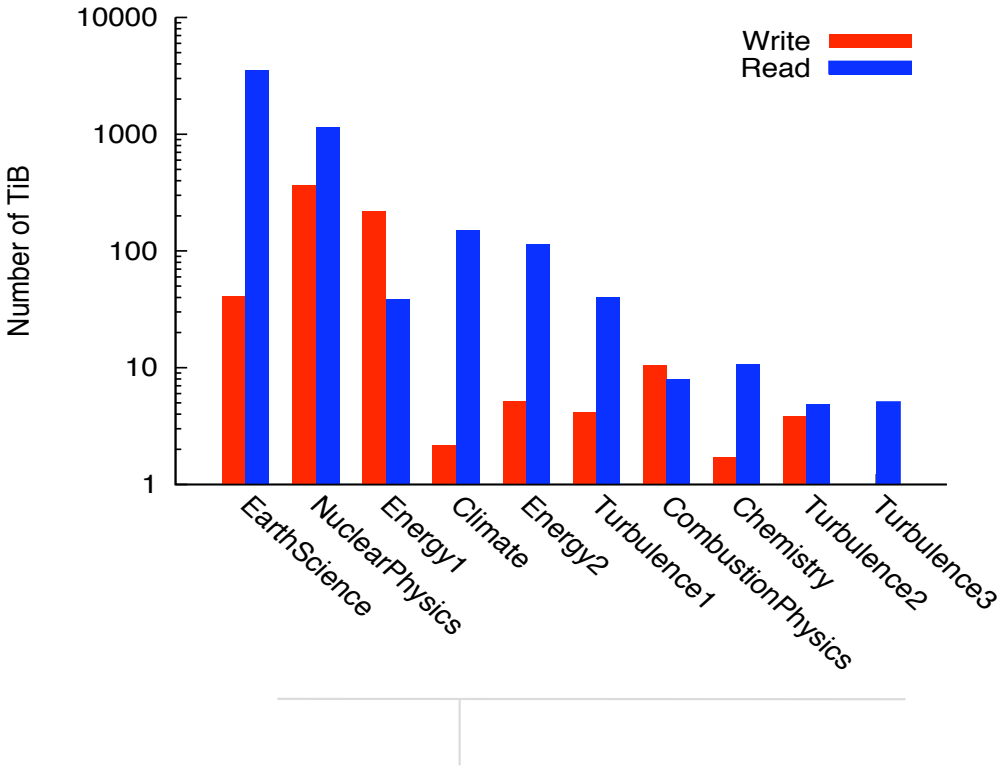
Darshan (Sanskrit for "sight") is a tool we developed for I/O characterization at extreme scale [Carns 2009]:

- No code changes, small and tunable memory footprint (~2MB default)
- Captures:
  - Counters for POSIX and MPI-IO operations
  - Counters for unaligned, sequential, consecutive, and strided access
  - Timing of opens, closes, first and last reads and writes
  - Cumulative data read and written
  - Histograms of access, stride, datatype, and extent sizes
- Aggregated and compressed output
  - 32K processes writing a shared file leads to 203 bytes of output
  - 32K processes writing a total of 262,144 files leads to 13.3MB of output

Two Months of Application I/O on ALCF Blue Gene/P

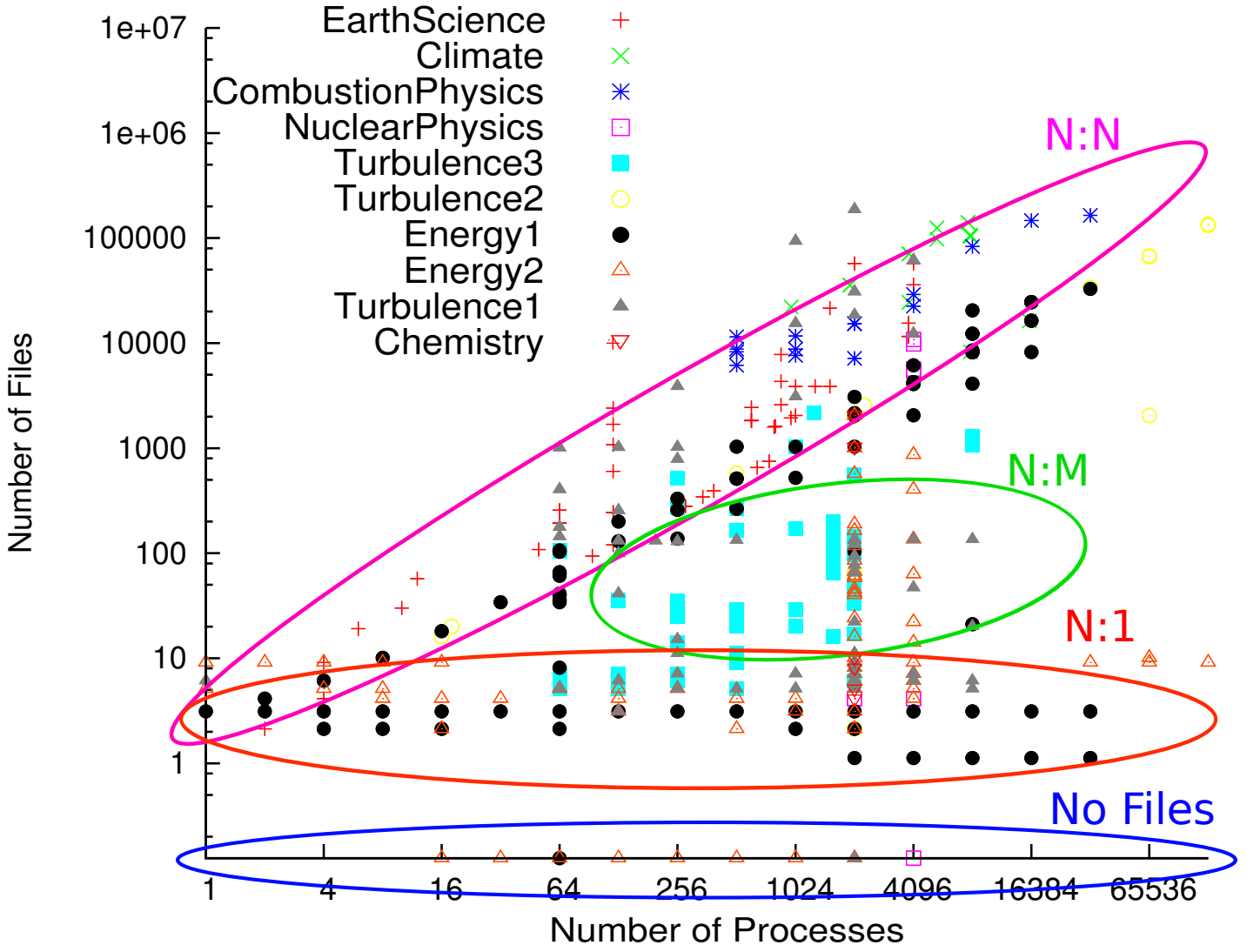
- After additional testing and hardening, Darshan installed on Intrepid
- By default, all applications compiling with MPI compilers are instrumented

- Data captured from late January through late March of 2010 [Carns 2011]
- Darshan captured data on 6,480 jobs (27%) from 39 projects (59%)
- Simultaneously captured data on servers related to storage utilization



Top 10 data producers and/or consumers shown. Surprisingly, most "big I/O" users read more data during simulations than they wrote.

Number of files generated by applications



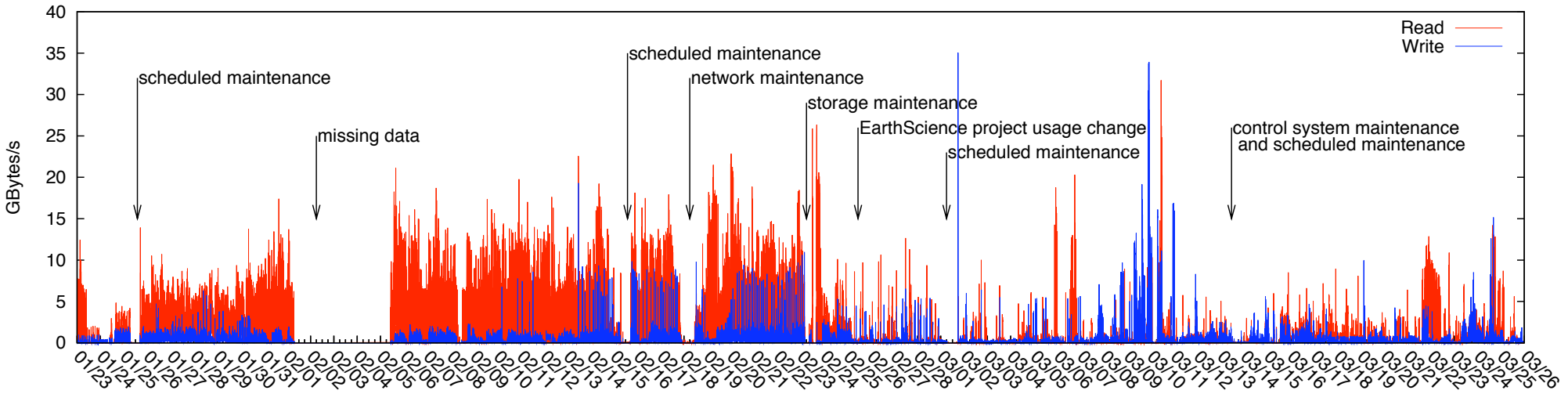
Results from Darshan study of ALCF BG/P system, looking at trends in terms of shared vs. independent file use across top data producer/consumers.

Real Application I/O on ALCF Blue Gene/P

Application	Mbytes/ sec/CN*	Cumulative MD	Files/ Proc	Creates/ Proc	Seq. I/O	Mbytes/ Proc
EarthScience	0.69	95%	140.67	98.87	65%	1779.48
NuclearPhysics	1.53	55%	1.72	0.63	100%	234.57
Energy1	0.77	31%	0.26	0.16	87%	66.35
Climate	0.31	82%	3.17	2.44	97%	1034.92
Energy2	0.44	3%	0.02	0.01	86%	24.49
Turbulence1	0.54	64%	0.26	0.13	77%	117.92
CombustionPhysics	1.34	67%	6.74	2.73	100%	657.37
Chemistry	0.86	21%	0.20	0.18	42%	321.36
Turbulence2	1.16	81%	0.53	0.03	67%	37.36
Turbulence3	0.58	1%	0.03	0.01	100%	40.40

* Synthetic I/O benchmarks (e.g., IOR) attain 3.93 - 5.75 Mbytes/sec/CN for modest job sizes, down to approximately 1.59 Mbytes/sec/CN for full-scale runs.

A System-side View of I/O



Aggregate I/O throughput on BG/P storage servers at one minute intervals.

- The I/O system is rarely **idle** at this granularity.
- The I/O system is also rarely at more than 33% of peak bandwidth.
- One particularly poor performing application can dramatically impact the system.

Reflecting on application I/O behavior…

- **Wide range of access patterns are seen**
  - No one of which is obviously the most successful
  - Speaks to need for different layouts for different purposes
- **High-level I/O libraries were rarely used in observed projects**, but the applications don't seem to have achieved higher performance by avoiding them
  - Applications shouldn't be tempted to optimize I/O on their own
  - Points to a need for immediate-term work to improve performance (e.g., ADIOS, PnetCDF subfiling, PLFS)
  - Selling point of high-level I/O libraries should be productivity, but need to convince users they are high performance also
- **I/O system is extremely underutilized**
  - "Room" for more data movement if we can enable asynchronous I/O

Storage System Designs

Vision: Adapting to Architectural Changes

- Highly adaptive system recognizes and responds to system perturbations
  - Low-overhead, scalable approach to information dissemination (e.g., gossip)
  - Ability to direct traffic to healthy targets, avoid overcommitted ones if possible
  - Allocation of bandwidth to simulations, analysis, and data replication/ reconstruction based on policy
- Incorporates new storage technologies into a multi-tier system
  - Aware of properties (e.g., capacity, performance, location, resilience) of storage targets
  - Integrates this information into service planning based on available (incomplete) knowledge of system state
  - Can leverage inexpensive, commodity storage devices to lower system cost

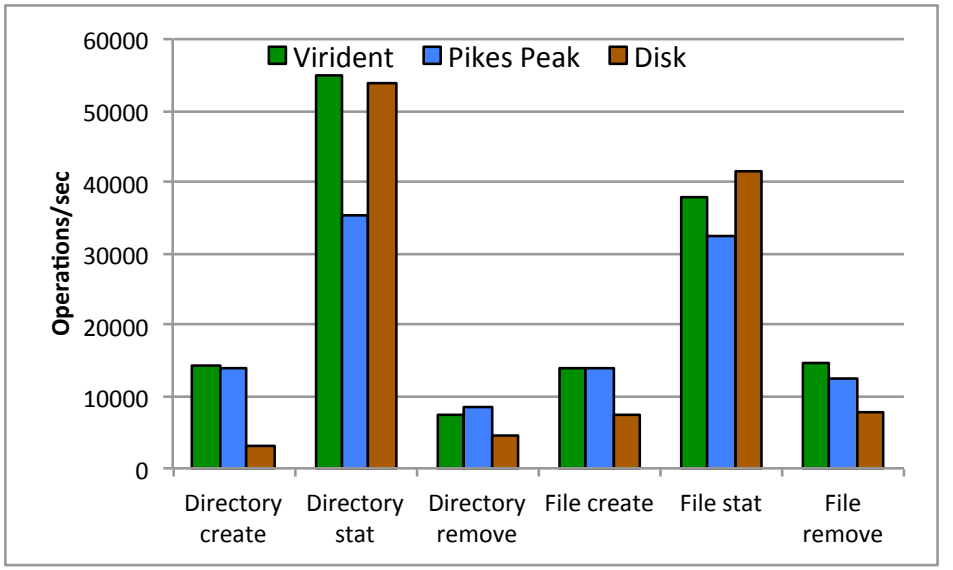


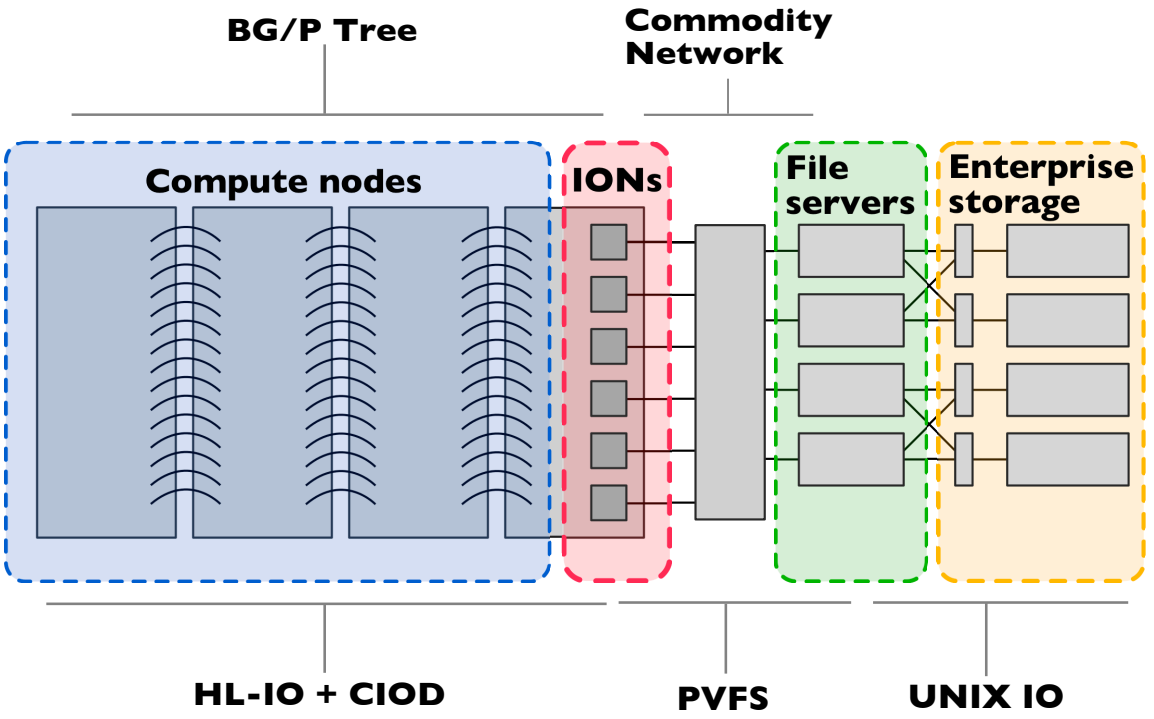
Figure 6: Lustre metadata results on Virident (SSD), Pikes Peak (SSD) and SATA disks as MDT for 64 MPI processors and a total of 300K files and directories.

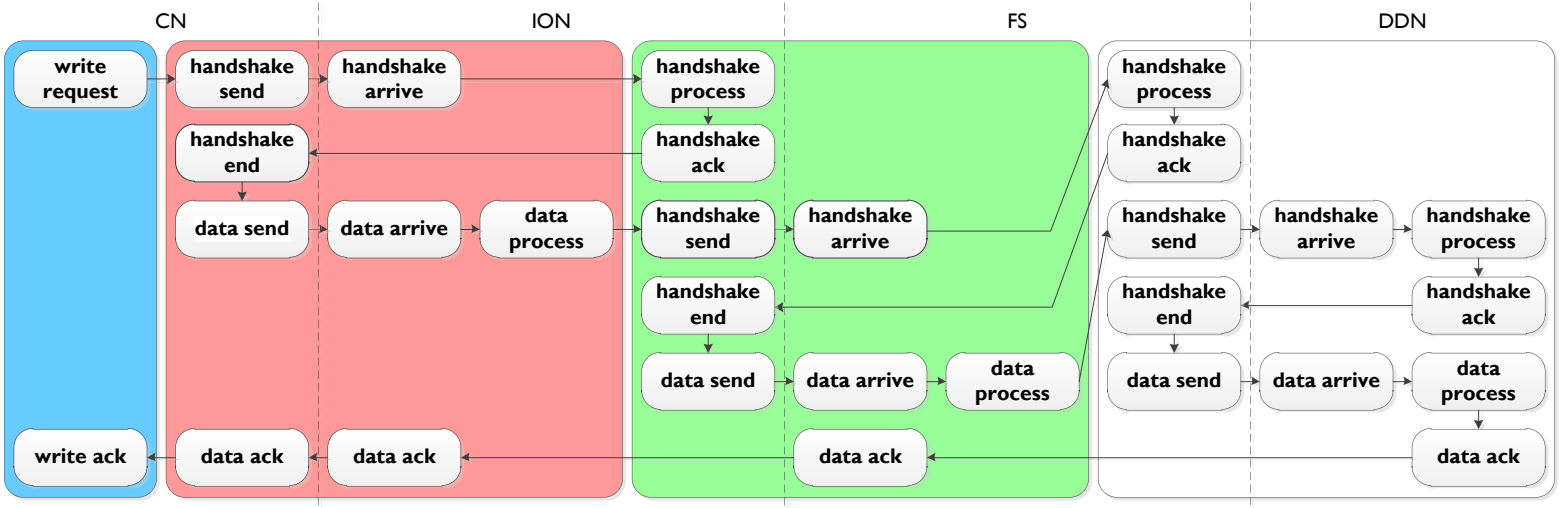
S. Alam et al. Parallel I/O and the metadata wall. Proceedings of the Parallel Data Storage Workshop, November 2011.

Assessing Extreme-Scale Storage via Simulation

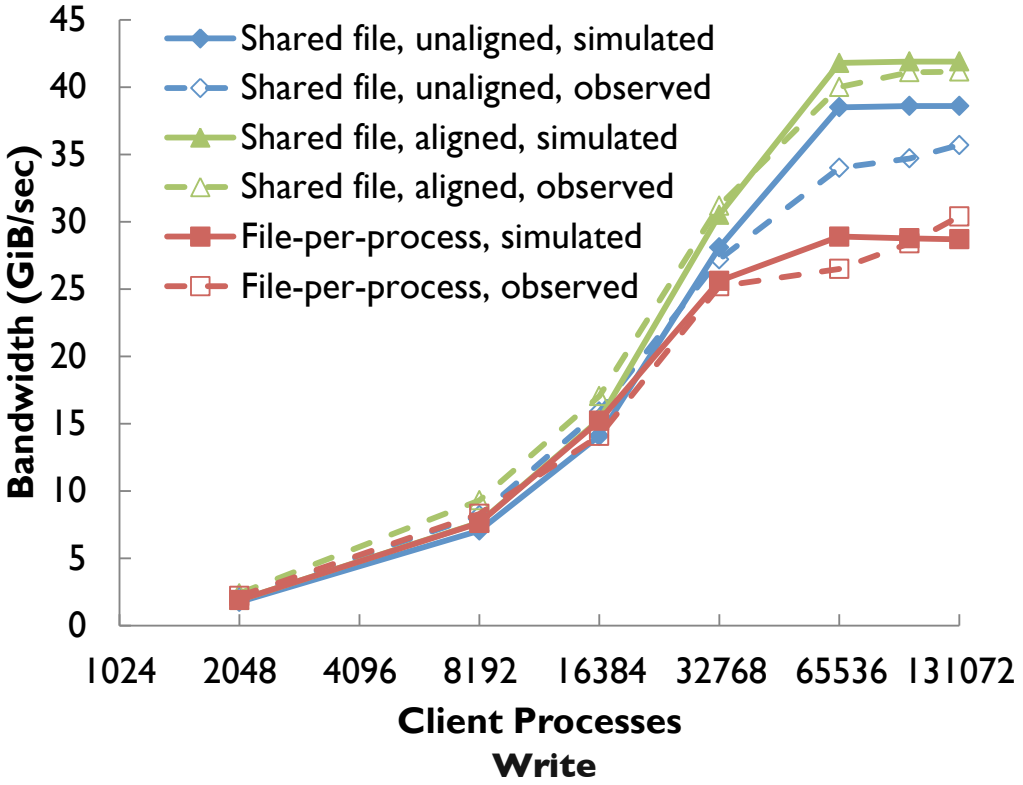
- Simulation is a critical tool for assessing future system designs
- Need scalable simulation capabilities so we can capture needed fidelity
- Working with C. Carothers, N. Liu (RPI) and A. Crume, C. Maltzhan (UCSC) to develop a simulation framework (CODES)
- Rensselaer Optimistic Simulation System (ROSS) as infrastructure
  - Parallel discrete-event simulator
  - Has been run on full BG/P system (as part of another project)
- Early work has focused on building a simulation of the Argonne BG/P system, so we can validate the approach and model [Liu 2011]

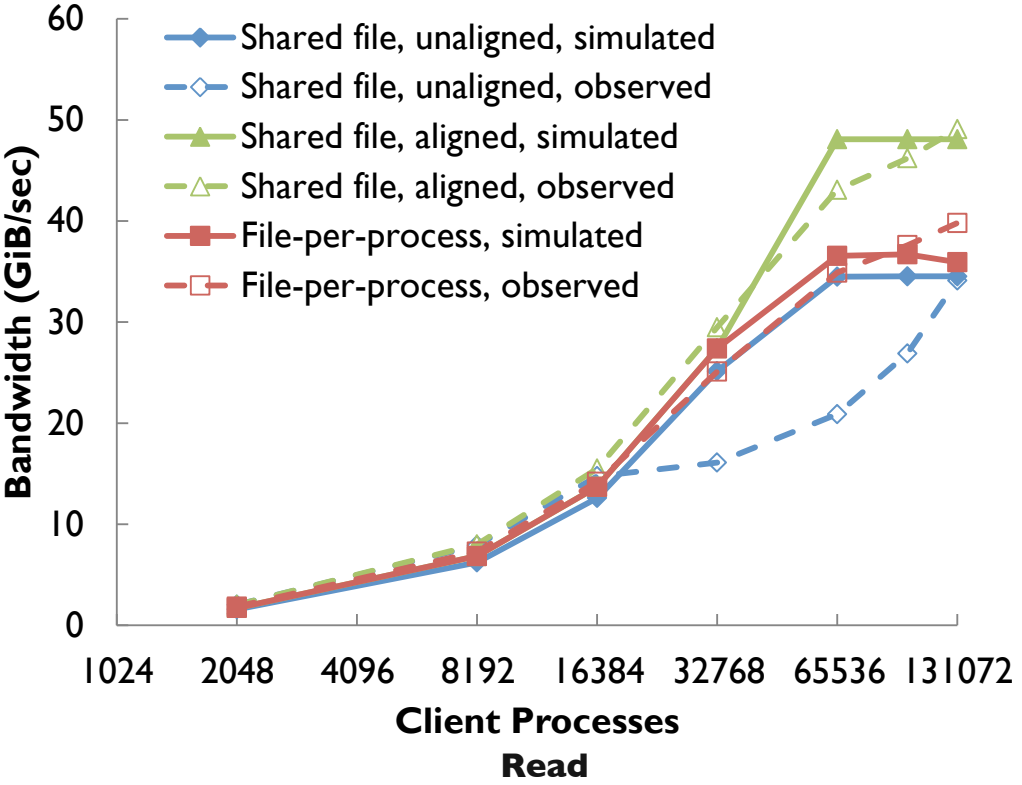
Simulating Storage: Components and Protocols





Early Results from Simulation of BG/P I/O System



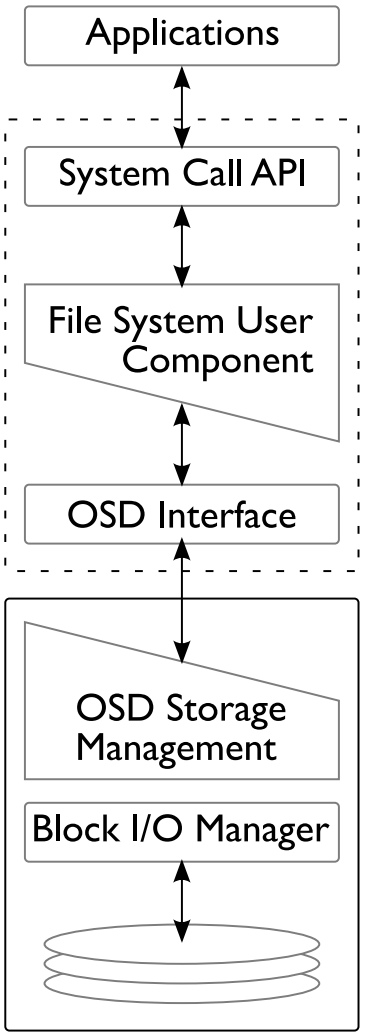


- Attempting to match results from SC09 paper detailing Intrepid I/O system
- Close!
- Unaligned performance is off (especially in read workload)
  - Network contention isn't accounted for
- Beginning to look at more complex I/O patterns now

Data Models and Storage Abstractions

Vision: Storage Abstractions for Computational Science

- Storage system exposes a data model meant for supporting many types of data
  - Containers with persistent references
  - Flexibility in how consistency semantics are applied (and a more sensible default)
  - Tunable resiliency that provides performance/safety/space trade-offs
  - Multiple options for defining name spaces
- Interfaces that provide rich descriptive capabilities in terms of concurrency, data movement, and future use
  - Capability to describe relationships and aggregate where possible, with minimal synchronization
  - Allowing the system to adapt in anticipation of future actions (e.g., migration)
  - Co-designed with applications (i.e., no shoehorning via ioctl())



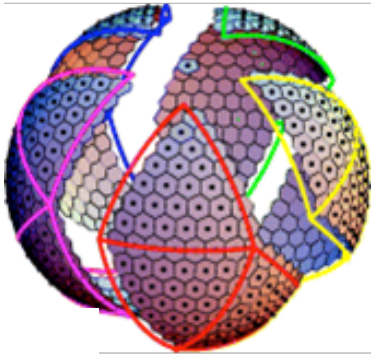
The object storage model is one step in the right direction in terms of storage abstractions for computational science.

Vision: Supporting Computational Science Data Models

- Must cover the variety of models present in computational science codes
  - Improves productivity of scientists
  - Captures regularity where it is present, with the flexibility to describe unstructured datasets as well
  - Retains all relevant semantics during “flattening” to storage

Motif	Data Model/Structure	Examples
Dense Linear Algebra	Multidimensional Arrays	ScaLAPACK, S3D
Sparse Linear Algebra	Sparse Matrix	OSKI, SuperLU
Spectral Methods	Multidimensional Arrays	Nek5000
N-Body Methods	Trees, Unstructured Meshes	Molecular Dynamics
Structured Grids (+ AMR)	Multidimensional Arrays	FLASH, Chombo-based
Unstructured Grids (+AMR)	Unstructured Meshes	UNIC, Phasta
Graph Traversal	Sparse Matrix, DAG	Decision Trees (e.g., C4.5)

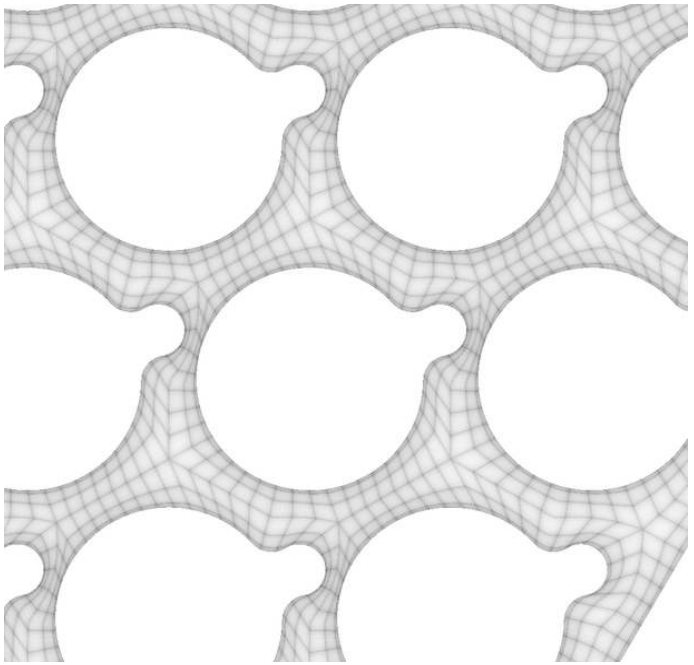
List of computational motifs and associated data models, derived from Berkeley Seven Dwarfs, modifications and additions by A. Choudhary, N. Samatova, Q. Koziol, T. Tautges, R. Latham, W. Liao, and R. Ross.



Geodesic grid used in global climate resolving model. From B. Palmer, A. Koontz, and K. Schuchardt, "An IO API for a Global Cloud Resolving Model". Environmental Modelling & Software (submitted).

Next-Generation Data Models

- Damsel project: supporting the complex models in computational science
  - A. Choudhary (NWU) is project PI
  - Support for structured and unstructured, regular and adaptive data models
- Storage and data movement
  - Mapping from science data models into novel storage data models [Gao 2009]
  - Supporting (many different) access patterns associated with model
  - Optimizing time to write in bandwidth limited environments [Kimpe 2007]
- Exploring the role of data models in analysis
  - Enabler for analysis at different locations in the data pipeline (GLEAN) [Vishwanath 2011]
  - Evaluating transformations for data analysis [Kumar 2011]



Cross-section of spectral element mesh used in large eddy simulation of 217-pin reactor subassembly.

Image from P. Fischer (ANL).

Concluding Remarks

- Ongoing activities in three areas:
  - Understanding and tracking application I/O behavior
  - Exploring the storage system design space
  - Building better support for computational science data models
- Need for strong connections with facilities
  - Deploying tools to understand I/O behavior
  - Gathering feedback on storage system designs
  - Building trust that revolutionary solutions are viable
- From HPC data to data intensive computing
  - What lessons can be learned from Internet services, observational and experimental sciences fields?
  - What can we teach them [Tantisiriroj 2011] ?
- Many opportunities for collaboration…

Acknowledgment

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

Our work is only possible with the help of our many collaborators, including:

- Alok Choudhary, Kui Gao, Wei-keng Liao, Arifa Nisar (NWU)
- Kwan-Liu Ma, Hongfeng Yu (UC Davis)
- Lee Ward (SNL)
- Gary Grider, James Nunez (LANL)
- Steve Poole, Terry Jones (ORNL)
- Yutaka Ishikawa, Kazuki Ohta (University of Tokyo)
- Javier Blas, Florin Isaila (University Carlos III of Madrid)
- Ning Liu, Chris Carothers (RPI)

Relevant Work

[Carns 2009] P. Carns, R. Latham, R. Ross, K. Iskra, S. Lang, and K. Riley. 24/7 characterization of petascale I/O workloads. In Proceedings of the First Workshop on Interfaces and Abstractions for Scientific Data Storage (IASDS), New Orleans, LA, September 2009.

[Carns 2011] P. Carns, K. Harms, W. Allcock, C. Bacon, R. Latham, S. Lang, and R. Ross. Understanding and improving computational science storage access through continuous characterization. In Proceedings of 27th IEEE Conference on Mass Storage Systems and Technologies (MSST 2011), May 2011. (Best Paper)

[Gao 2009] K. Gao, W. Liao, A. Nisar, A. Choudhary, R. Ross, and R. Latham. Using subfiling to improve programming flexibility and performance of parallel shared-file I/O. In Proc. ICPP 09, Vienna, Austria, September 2009.

[Kimpe 2007] D. Kimpe, R. Ross, S. Vandewalle, and S. Poedts. Transparent log-based data storage in MPI-IO applications. In Proc. of the 14th European PVM/MPI Users' Group Meeting (Euro PVM/MPI 2007), September 2007.

[Kumar 2011] S. Kumar, V. Vishwanath, P. Carns, B. Summa, G. Scorzelli, V. Pascucci, R. Ross, J. Chen, H. Kolla, and R. Grout. PIDX: Efficient parallel I/O for multi-resolution multi-dimensional scientific datasets. In Proceedings of IEEE Cluster 2011, Austin, TX, September 2011.

[Lang 2009] S. Lang, P. Carns, R. Latham, R. Ross, K. Harms, and W. Allcock. I/O performance challenges at leadership scale. In Proceedings of Supercomputing, November 2009.

[Liu 2011] N. Liu, C. Carothers, J. Cope, P. Carns, R. Ross, A. Crume, and C. Maltzhan. Modeling a leadership-scale storage system. Proceedings of the 9th International Conference on Parallel Processing and Applied Mathematics 2011 (PPAM 2011), September 2011 (paper to appear).

[Tantisiriroj 2011] W. Tantisiriroj, S. Patil, G. Gibson, S. W. Son, S. J. Lang, and R. B. Ross. On the duality of data-intensive file system design: Reconciling HDFS and PVFS. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC11), Seattle, WA, November 2011.

[Vishwanath 2011] V.Vishwanath, M. Hereld, and M. E. Papka, Toward Simulation-time data analysis and I/O acceleration on leadership-class systems using GLEAN. IEEE Symposium on Large Data Analysis and Visualization (LDAV), Providence, RI, USA, October 2011.

[Wozniak 2010] J. Wozniak, S. W. Son, and R. Ross. Distributed object storage rebuild analysis via simulation with GOBS. In Workshop on Fault-Tolerance for HPC at Extreme Scale, June 2010.