HPC Storage and Data: Current State and Future Directions

Rob Ross
Mathematics and Computer Science Division
Argonne National Laboratory
rross@mcs.anl.gov

Quick Intro to HPC Storage

HPC I/O Systems

HPC I/O system is the hardware and software that assists in accessing data during simulations and analysis and retaining data between these activities.

- Hardware: disks, disk enclosures, servers, networks, etc.
- Software: parallel file system, libraries, parts of the OS

- Two "flavors" of I/O from applications:
  - **Defensive**: storing data to protect results from data loss due to system faults
  - **Productive**: storing/retrieving data as part of the scientific workflow
  - Note: Sometimes these are combined (i.e., data stored both protects from loss and is used in later analysis)
- "Flavor" influences priorities:
  - Defensive I/O: Spend as little time as possible
  - Productive I/O: Capture provenance, organize for analysis

HPC I/O Software Stack

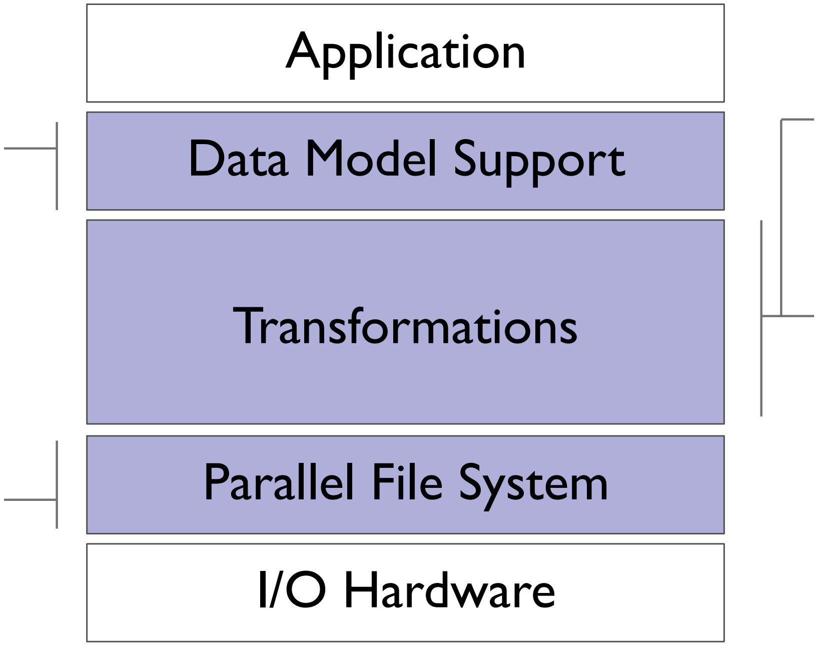
The software used to provide data model support and to transform I/O to better perform on today's I/O systems is often referred to as the *I/O stack.*

Data Model Libraries map application abstractions onto storage abstractions and provide data portability.

HDF5, Parallel netCDF, ADIOS

Parallel file system maintains logical file model and provides efficient access to data.

PVFS, PanFS, GPFS, Lustre



I/O Middleware organizes accesses from many processes, especially those using collective I/O.

MPI-IO, PLFS

I/O Forwarding transforms I/O from many clients into fewer, larger request; reduces lock contention; and bridges between the HPC system and external storage.

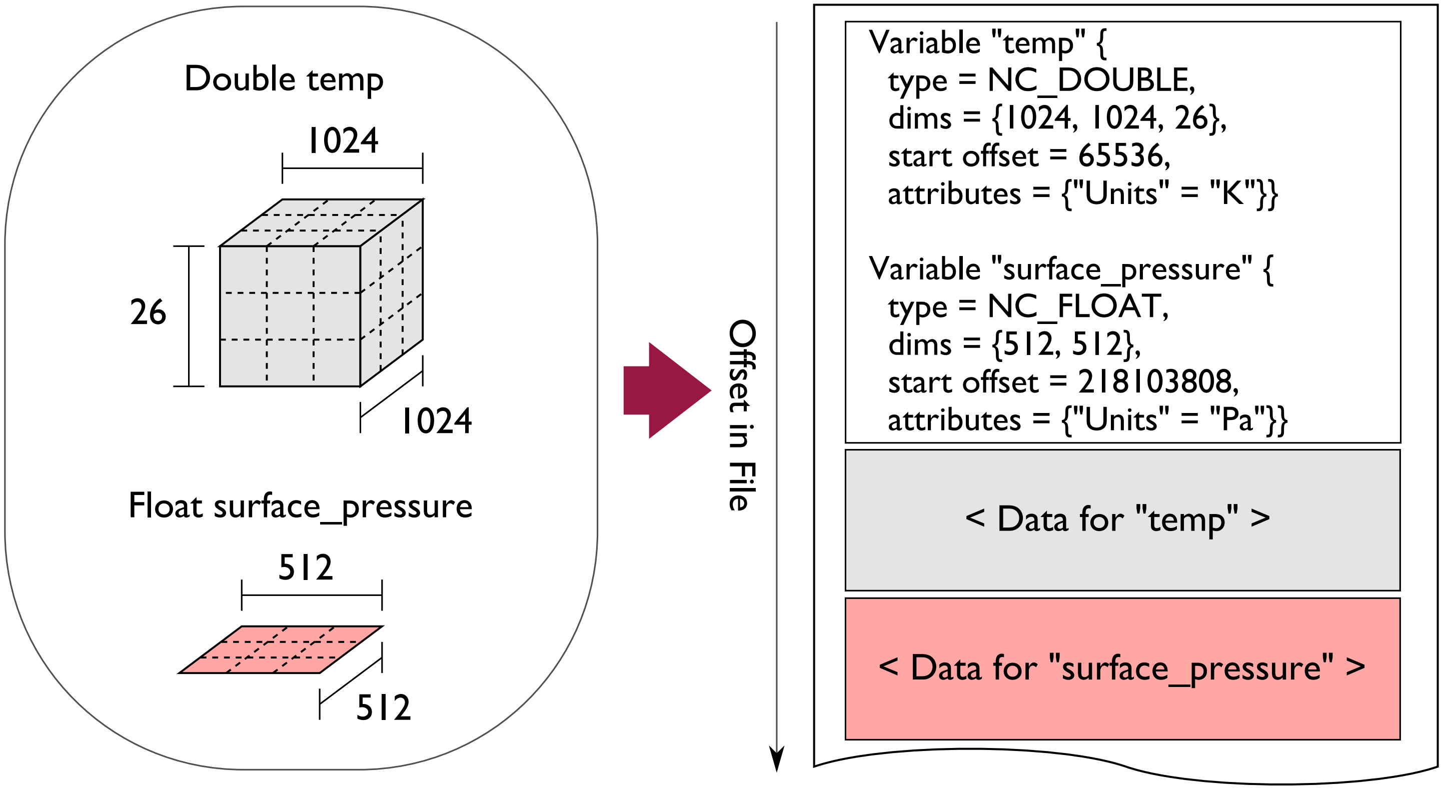
IBM ciod, IOFSL, Cray DVS

Storing and Organizing Data: Application Model(s)

Application data models are supported via libraries that map down to files (and sometimes directories).

Application Data Structures

netCDF File "checkpoint07.nc"



```
Variable "temp" {
  type = NC_DOUBLE,
  dims = {1024, 1024, 26},
  start offset = 65536,
  attributes = {"Units" = "K"}}

Variable "surface_pressure" {
  type = NC_FLOAT,
  dims = {512, 512},
  start offset = 218103808,
  attributes = {"Units" = "Pa"}}
```

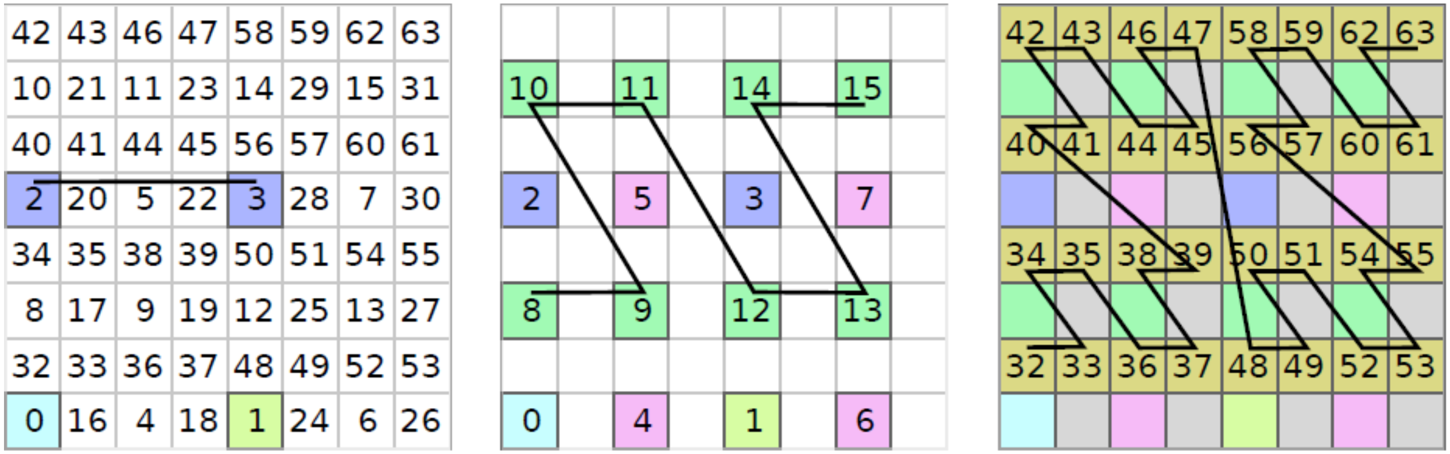
netCDF header describes the contents of the file: typed, multi-dimensional variables and attributes on variables or the dataset itself.

Data for variables is stored in contiguous blocks, encoded in a portable binary format according to the variable's type.

Example: Multi-Resolution Data Organization

- IDX is a library for storing multidimensional data in a multi-resolution format
  - Enables fast browsing of very large datasets
  - PIDX version enables writing of data directly into this format from simulation codes, streaming from simulation to analysis codes

Contact: V. Pascucci <pascucci@sci.utah.edu> and S. Kumar (U. of Utah)

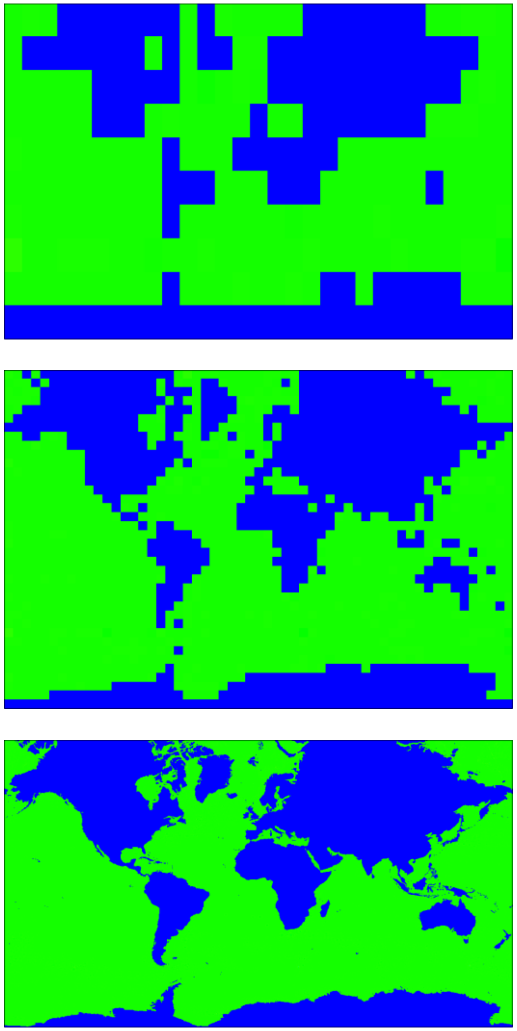


42	43	46	47	58	59	62	63
10	21	11	23	14	29	15	31
40	41	44	45	56	57	60	61
2	20	5	22	3	28	7	30
34	35	38	39	50	51	54	55
8	17	9	19	12	25	13	27
32	33	36	37	48	49	52	53
0	16	4	18	1	24	6	26

10		11		14		15	
2		5		3		7	
8		9		12		13	
0		4		1		6	

42	43	46	47	58	59	62	63
40	41	44	45	56	57	60	61
34	35	38	39	50	51	54	55
32	33	36	37	48	49	52	53

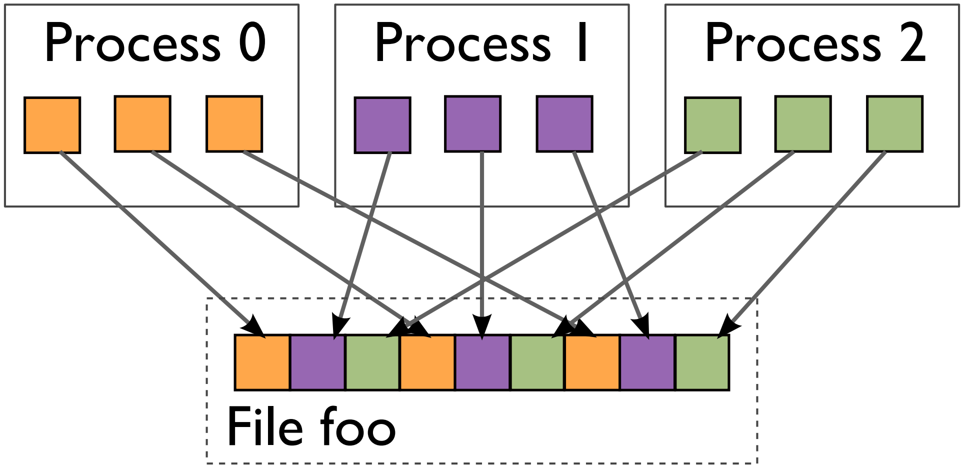
Hierarchical z-order data organization enables multi-resolution access while retaining spatial locality.



I/O Transformations

Software between the application and the PFS performs transformations, primarily to improve performance.

- Goals of transformations:
  - Reduce number of operations to PFS (avoiding latency)
  - Avoid lock contention (increasing level of concurrency)
  - Hide number of clients (more on this later)
- With "transparent" transformations, data ends up in the same locations in the file
  - i.e., the file system is still aware of the actual data organization



When we think about I/O transformations, we consider the mapping of data between application processes and locations in file.

Storing and Organizing Data: Storage Model

HPC I/O systems are built around a *parallel file system* that organizes storage and manages access.

- Parallel file systems (PFSes) are distributed systems that provide a file data model (i.e., files and directories) to users
- Multiple PFS servers manage access to storage, while PFS client systems run applications that access storage
- PFS clients can access storage resources in parallel!

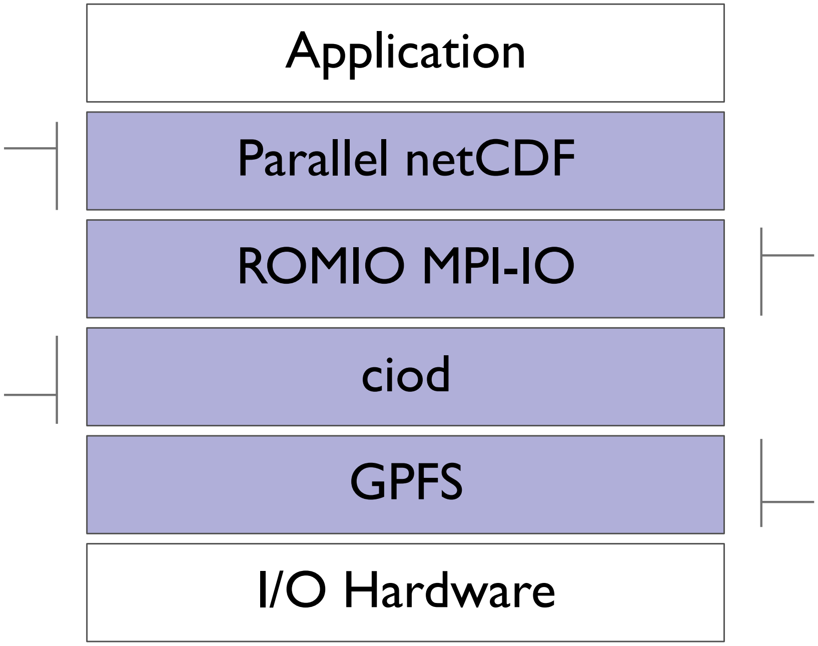
An Example HPC I/O Software Stack

This example I/O stack captures the software stack used in some applications on the IBM Blue Gene/Q system at Argonne.

Parallel netCDF is used in numerous climate and weather applications running on DOE systems.

Built in collaboration with NWU.

ciod is the I/O forwarding implementation on the IBM Blue Gene/P and Blue Gene/Q systems.



Application

Parallel netCDF

ROMIO MPI-IO

ciod

GPFS

I/O Hardware

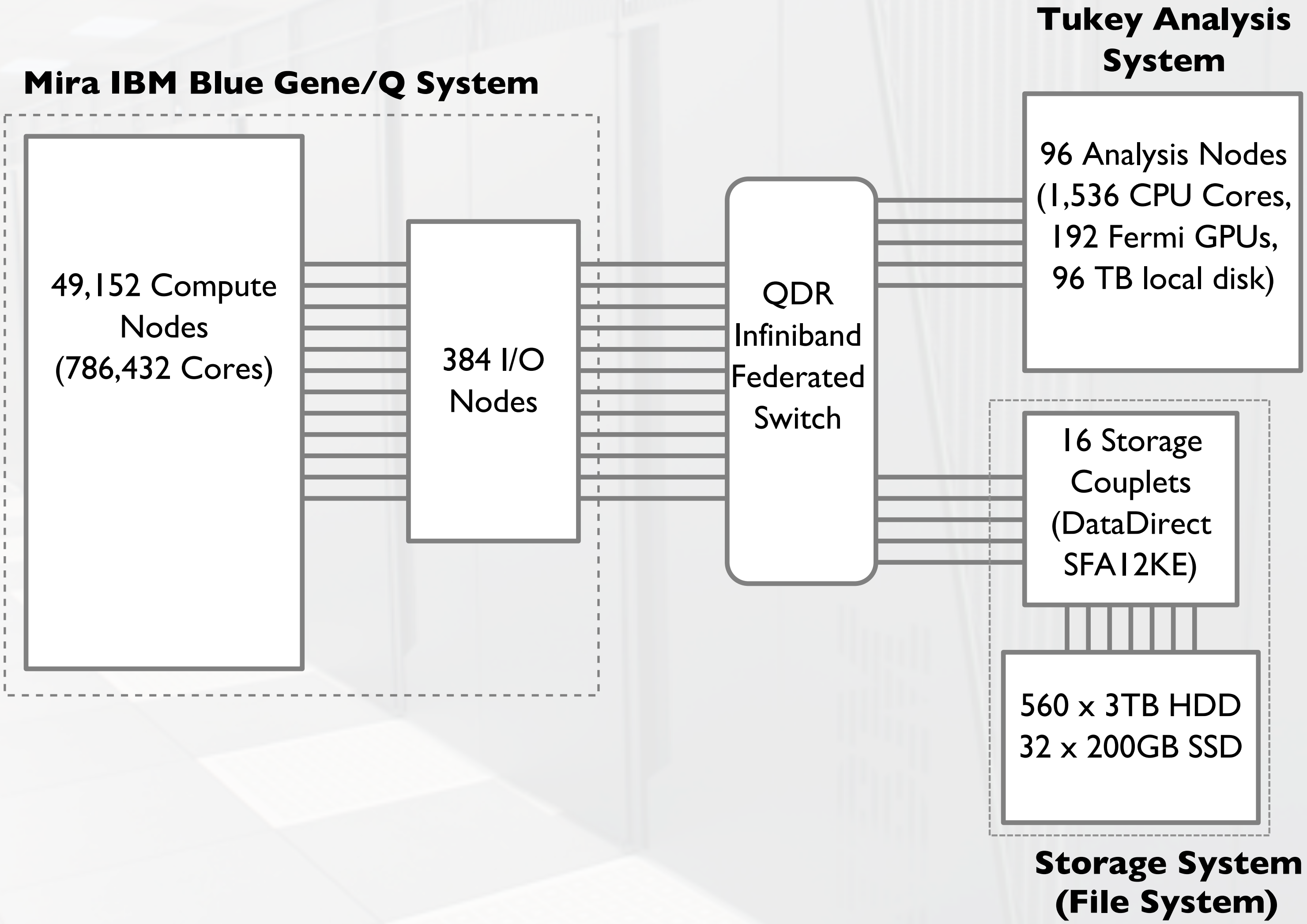
ROMIO is the basis for virtually all MPI-IO implementations on all platforms today and the starting point for nearly all MPI-IO research.

Incorporates research from NWU and patches from vendors.

GPFS is a production parallel file system provided by IBM.

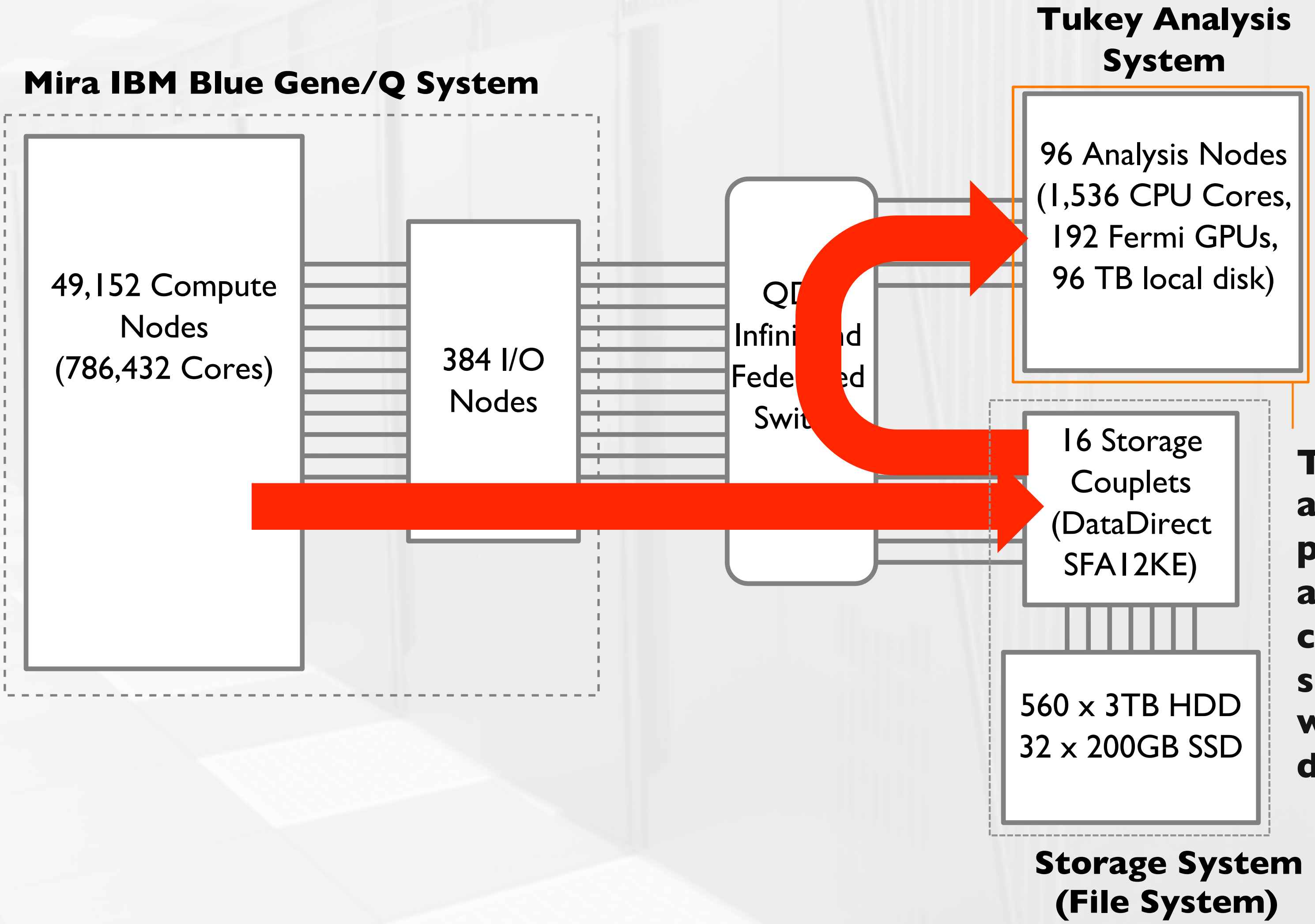
HPC Systems and Storage Hardware

An Example Leadership System Architecture


Mira IBM Blue Gene/Q System
49,152 Compute Nodes (786,432 Cores)
384 I/O Nodes
QDR Infiniband Federated Switch
Tukey Analysis System
96 Analysis Nodes (1,536 CPU Cores, 192 Fermi GPUs, 96 TB local disk)
16 Storage Couplets (DataDirect SFA12KE)
560 x 3TB HDD
32 x 200GB SSD
Storage System (File System)

High-level diagram of 10 Pflop IBM Blue Gene/Q system at Argonne Leadership Computing Facility

Analyzing Data: Traditional Post-Processing


Tukey Analysis System
Mira IBM Blue Gene/Q System
49,152 Compute Nodes (786,432 Cores)
384 I/O Nodes
QD
Infini nd
Fede ed
Swit
96 Analysis Nodes (1,536 CPU Cores, 192 Fermi GPUs, 96 TB local disk)
16 Storage Couplets (DataDirect SFA12KE)
560 x 3TB HDD
32 x 200GB SSD
Storage System (File System)

Typically analysis is performed on a separate cluster, after simulation has written to data to disk.

High-level diagram of 10 Pflop IBM Blue Gene/Q system at Argonne Leadership Computing Facility

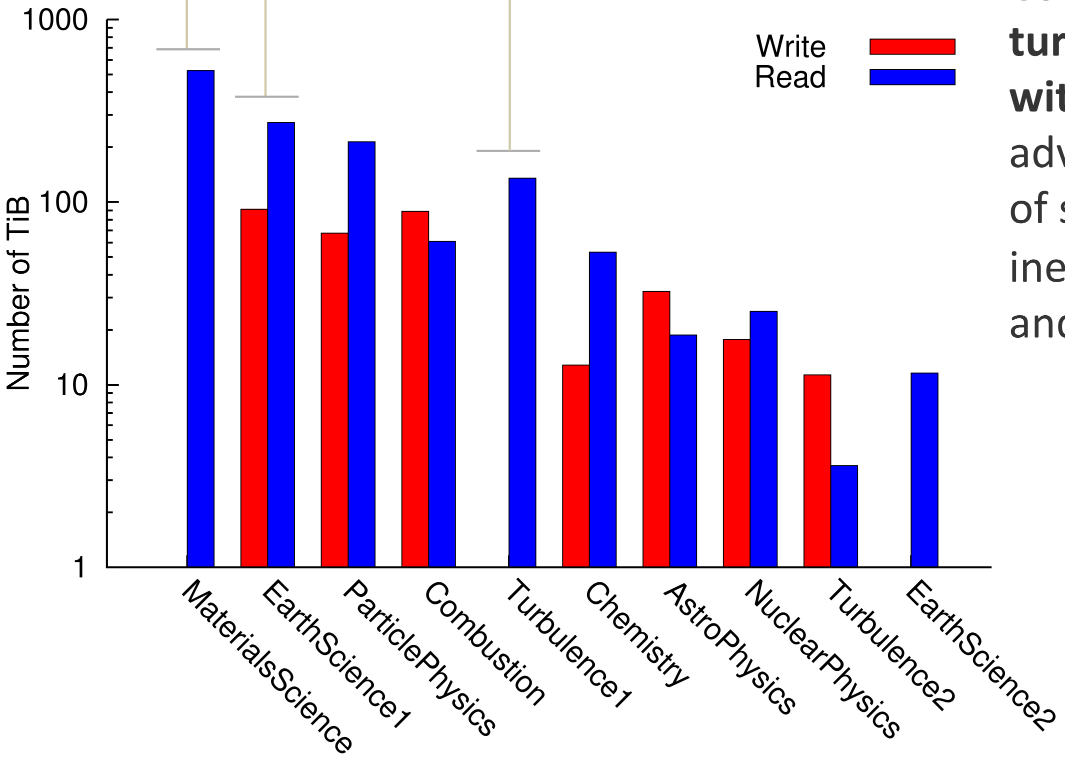
Data Analysis: Beyond Post-Processing

Data analysis is happening directly on HPC systems.

Matching large scale simulations of dense suspensions with empirical measurements to better understand properties of complex materials such as concrete.

Processing large-scale seismographic datasets to develop a 3D velocity model used in developing earthquake hazard maps.

Comparing simulations of turbulent mixing of fluids with experimental data to advance our understanding of supernovae explosions, inertial confinement fusion, and supersonic combustion.

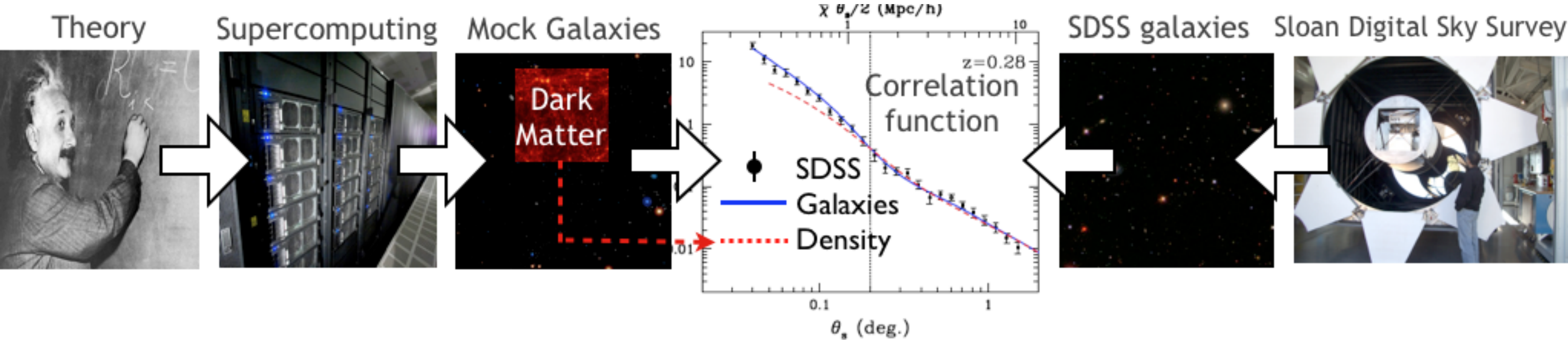


Top 10 data producer/consumers instrumented with Darshan over the month of July, 2011 on Intrepid BG/P system at Argonne. Surprisingly, three of the top producer/consumers almost exclusively read existing data.

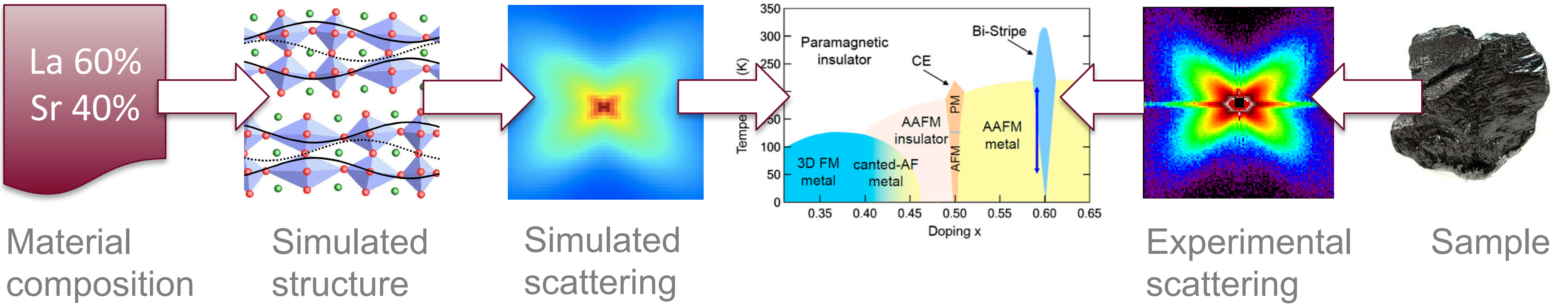
Data-Driven Science Examples

For many problems there is a deep coupling of observation (measurement) and computation (simulation)

Cosmology: The study of the universe as a dynamical system

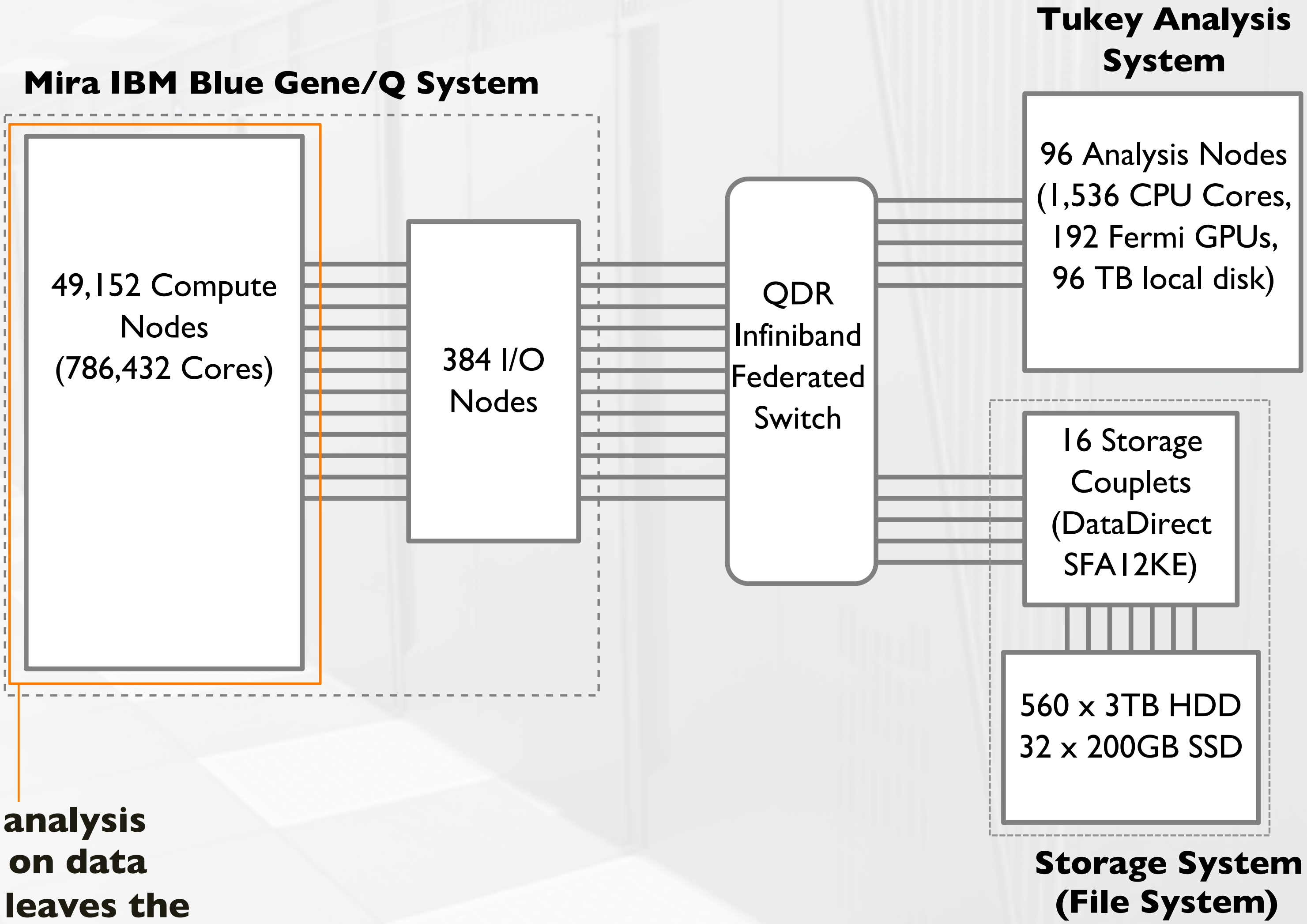


Materials science: Diffuse scattering to understand disordered structures



Images from Salman Habib et al. (HEP, MCS, etc.) and Ray Osborne et al. (MSD, APS, etc.)

Analyzing Data: In Situ Analysis



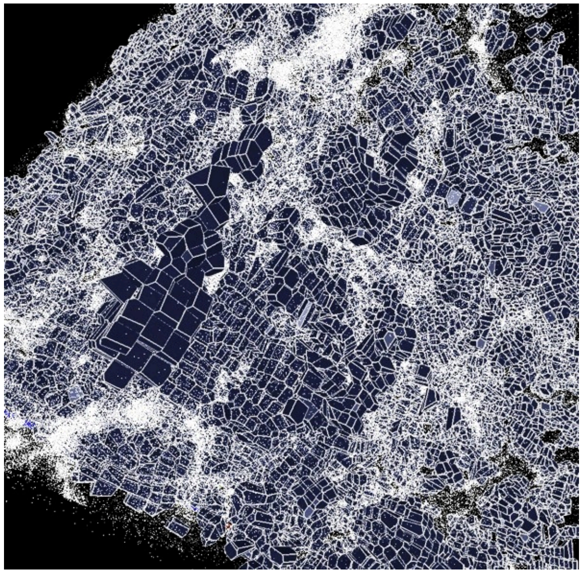
Mira IBM Blue Gene/Q System
49,152 Compute Nodes (786,432 Cores)
384 I/O Nodes
QDR Infiniband Federated Switch
Tukey Analysis System
96 Analysis Nodes (1,536 CPU Cores, 192 Fermi GPUs, 96 TB local disk)
16 Storage Couplets (DataDirect SFA12KE)
560 x 3TB HDD
32 x 200GB SSD
Storage System (File System)

"In situ" analysis operates on data before it leaves the compute nodes.

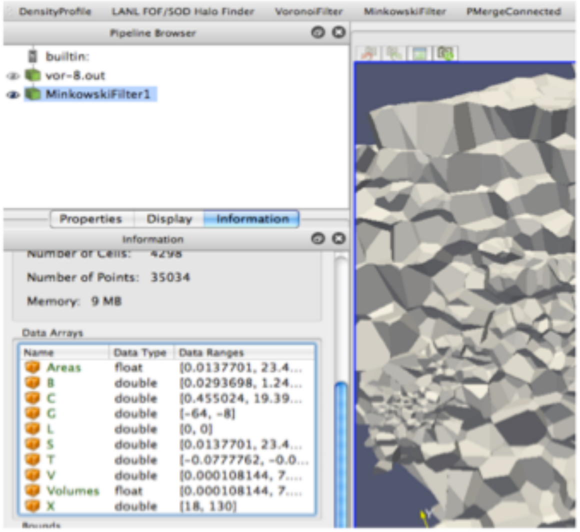
High-level diagram of 10 Pflop IBM Blue Gene/Q system at Argonne Leadership Computing Facility

Analysis at Scale: In Situ Analysis and Data Reduction

- On the HPC side, analysis is increasingly performed during runtime to avoid subsequent I/O
- HACC cosmology code employing Voronoi tessellation
  - Converts from particles into unstructured grid
  - Adaptive, retains full dynamic range of input
  - DIY toolkit used to implement analysis routines
- ParaView environment used for visual exploration, custom tools for analysis



Voronoi tessellation reveals regions of irregular low-density voids amid high-density halos.



Number of Cells: 4298

Number of Points: 35034

Memory: 9 MB

Name	Data Type	Data Ranges
Areas	float	[0.0137701, 23.4...
B	double	[0.0293698, 1.24...
C	double	[0.455024, 19.39...
G	double	[-64, -8]
L	double	[0, 0]
S	double	[0.0137701, 23.4...
T	double	[-0.0777762, -0.0...
V	double	[0.000108144, 7....
Volumes	float	[0.000108144, 7....
X	double	[18, 130]

ParaView plugin provides interactive feature exploration.

Contact: Tom Peterka < tpeterka@mcs.anl.gov >
Collaboration with Berkeley Laboratory, Kitware, and U. of Tennessee

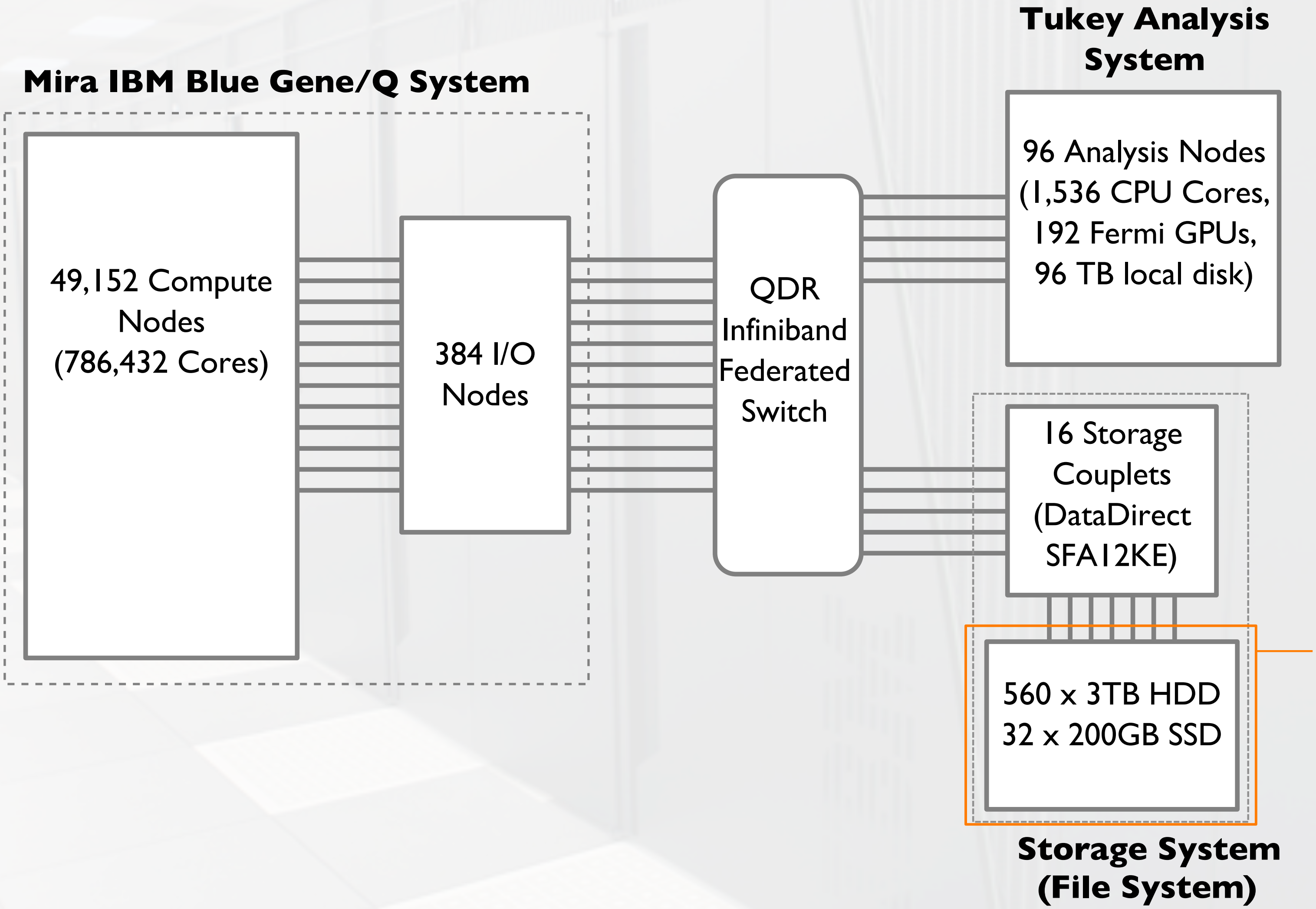
Nonvolatile Memory and HPC Architectures

Nonvolatile Memory

- There are a bunch of technologies that allow us to persistently store data at higher performance than disk
  - FLASH
  - Storage Class Memory technologies
- Properties vary by technology
  - FLASH has fast random reads, (somewhat) slower writes, endurance "issues"
- Not as expensive as DRAM, open up new options for storage in HPC systems

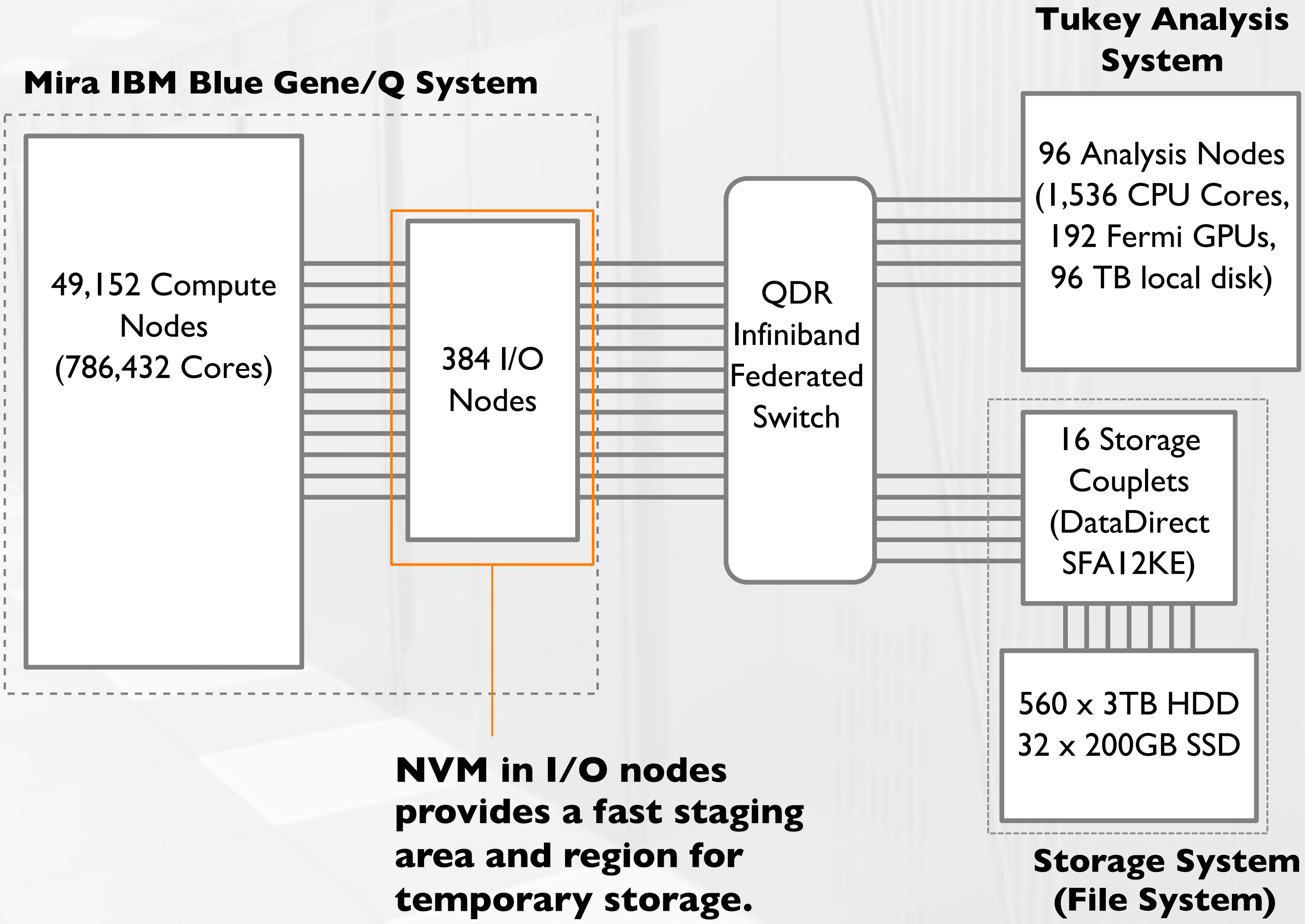
Where do we put it?

Nonvolatile Memory in Storage System


Mira IBM Blue Gene/Q System
49,152 Compute Nodes (786,432 Cores)
384 I/O Nodes
QDR Infiniband Federated Switch
Tukey Analysis System
96 Analysis Nodes (1,536 CPU Cores, 192 Fermi GPUs, 96 TB local disk)
16 Storage Couplets (DataDirect SFA12KE)
560 x 3TB HDD
32 x 200GB SSD
Storage System (File System)

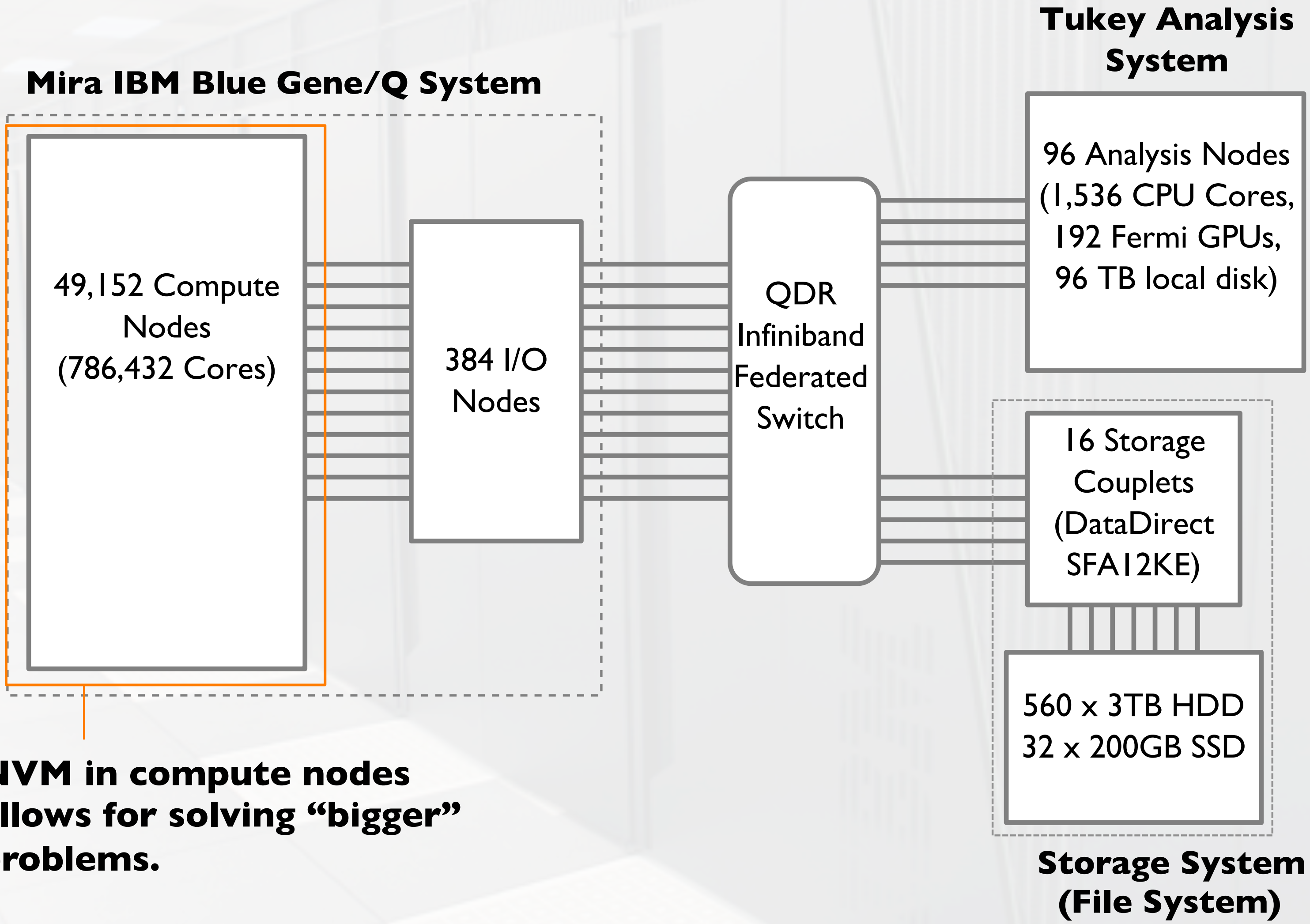
NVM in storage nodes serves as a PFS accelerator.

Nonvolatile Memory in I/O Nodes


Mira IBM Blue Gene/Q System
49,152 Compute Nodes (786,432 Cores)
384 I/O Nodes
QDR Infiniband Federated Switch
Tukey Analysis System
96 Analysis Nodes (1,536 CPU Cores, 192 Fermi GPUs, 96 TB local disk)
16 Storage Couplets (DataDirect SFA12KE)
560 x 3TB HDD
32 x 200GB SSD
Storage System (File System)

NVM in I/O nodes provides a fast staging area and region for temporary storage.

Nonvolatile Memory in Compute System


Mira IBM Blue Gene/Q System
49,152 Compute Nodes (786,432 Cores)
384 I/O Nodes
QDR Infiniband Federated Switch
Tukey Analysis System
96 Analysis Nodes (1,536 CPU Cores, 192 Fermi GPUs, 96 TB local disk)
16 Storage Couplets (DataDirect SFA12KE)
560 x 3TB HDD
32 x 200GB SSD
Storage System (File System)

NVM in compute nodes allows for solving “bigger” problems.

Wrapping Up

SDAV

Scalable Data Management, Analysis, and Visualization

Goal is to assist application scientists in using state-of-the-art data management, analysis, and visualization techniques to make new science discoveries:

- **Data Management** – infrastructure that captures the data models used in science codes, efficiently moves, indexes, and compresses this data, enables query of scientific datasets, and provides the underpinnings of in situ data analysis
- **Data Analysis** – application-driven, architecture-aware techniques for performing in situ data analysis, filtering, and reduction to optimize downstream I/O and prepare for in-depth post-processing analysis and visualization
- **Data Visualization** – exploratory visualization techniques that support understanding ensembles of results, methods of quantifying uncertainty, and identifying and understanding features in multi- scale, multi-physics datasets

- Lead by Arie Shoshani (LBNL)
- Focus is on users of largest DOE/ASCR computational resources
- http://www.sdav-scidac.org

Acknowledgments

This work was supported by the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357, including through the Scientific Discovery through Advanced Computing (SciDAC) Institute of Scalable Data Management, Analysis and Visualization.