HPC I/O Principles: Everything you always wanted to know about HPC I/O but were afraid to ask

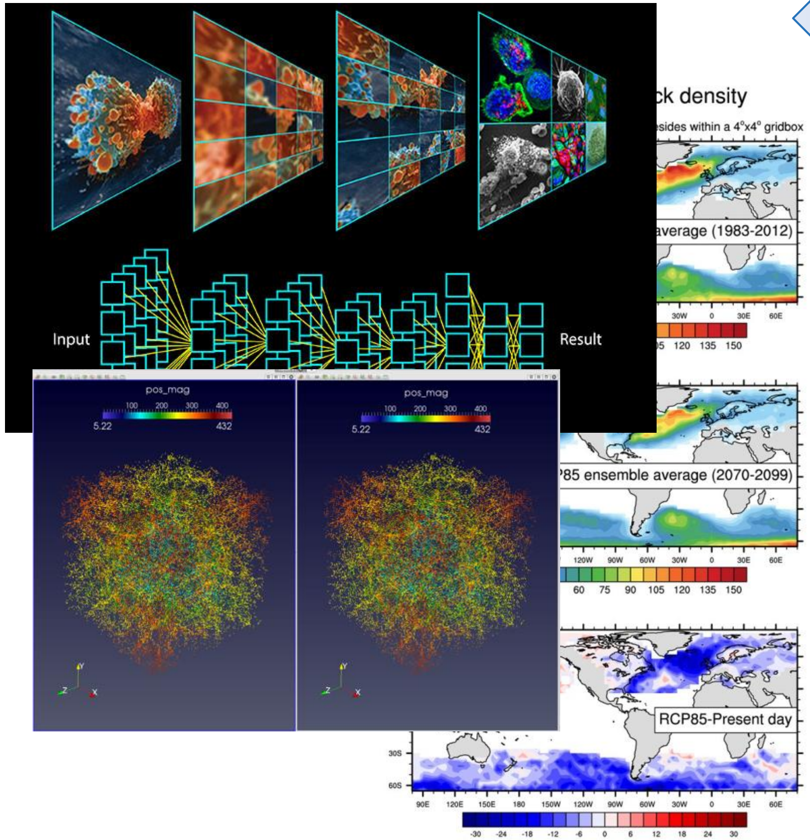
ATPESC 2018

Phil Carns
Mathematics and Computer Science Division
Argonne National Laboratory

Q Center, St. Charles, IL (USA)
July 29 – August 10, 2018

The role of data-intensive computer science research

(your lecturers’ day job)

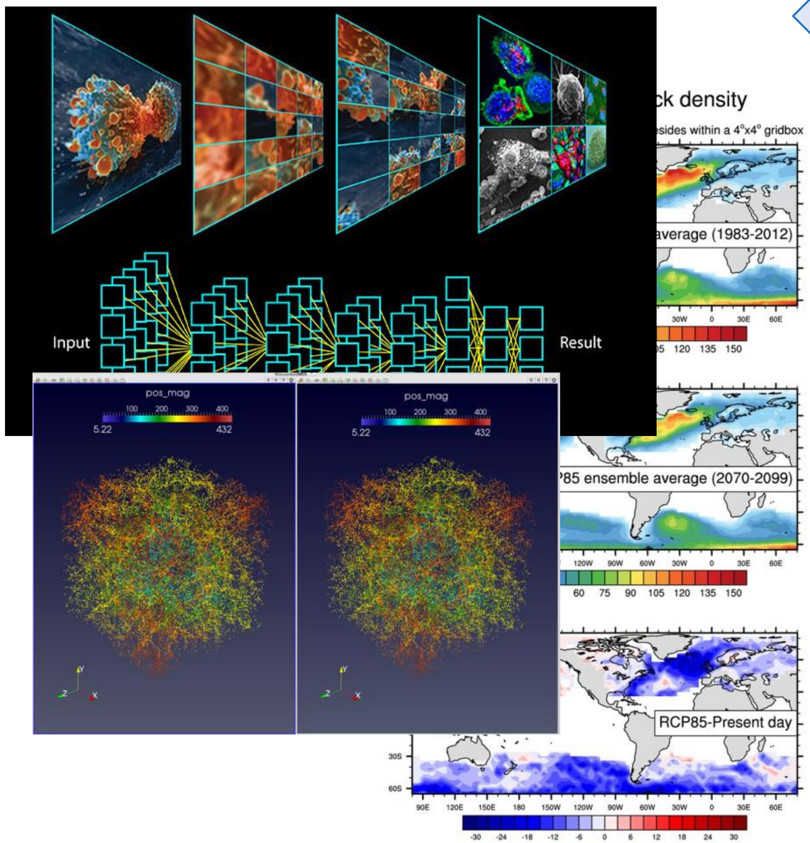


Techniques, algorithms, and software to bridge the “last mile” between scientific applications and storage systems

EXASCALE COMPUTING PROJECT

The role of data-intensive computer science research

(your lecturers’ day job)



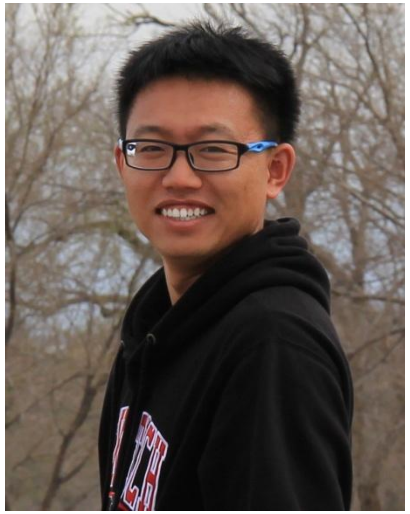
This means:

- Characterizing storage use
- Building and optimizing data services
- Modeling storage systems
- **Putting new technology into the hands of scientists**

Meet your lecturers

Rob Latham is a principal software development specialist at ANL who strives to make applications use I/O more efficiently. He has played a prominent role in the ROMIO MPI-IO implementation, the PVFS file system, and the PnetCDF high level library.

Quincey Koziol is a principal data architect at LBNL where he drives scientific data architecture discussions and participates in NERSC system design activities. He was the principal architect for the HDF5 project and a founding member of the HDF Group.

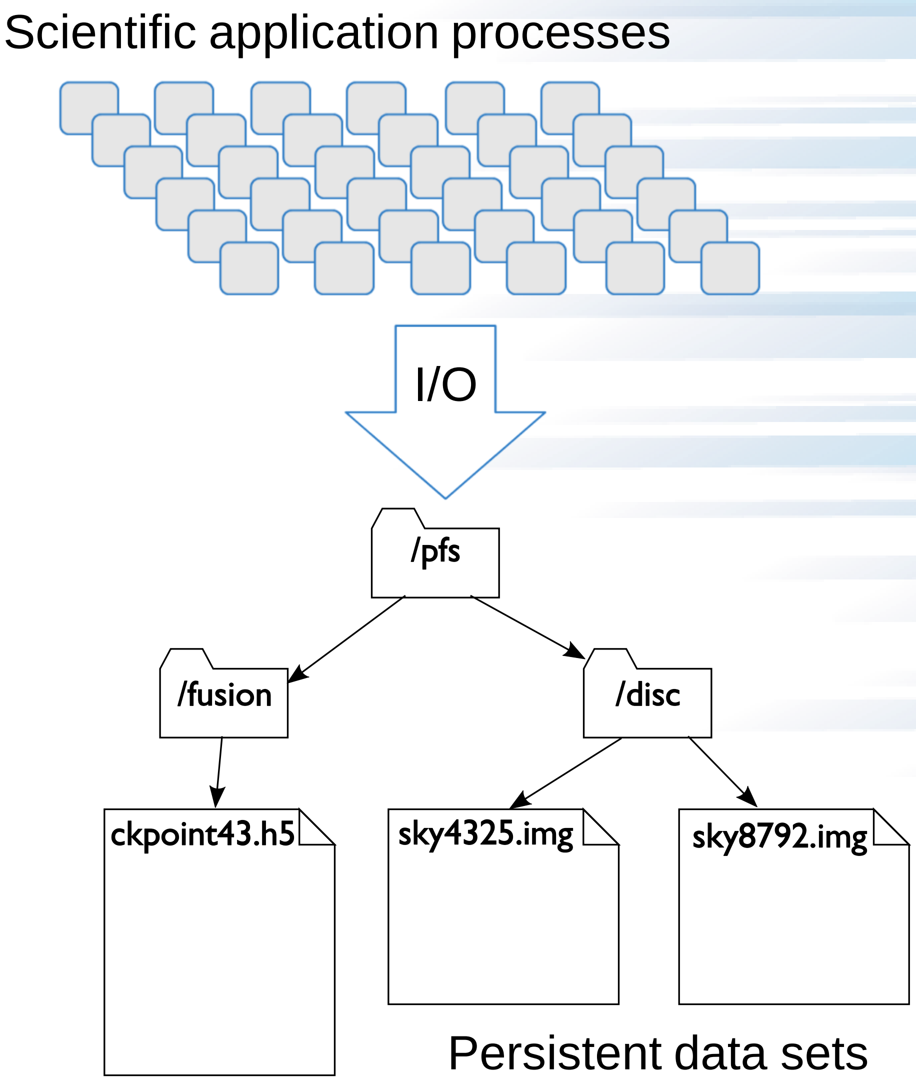


Jialin Liu is a HPC engineer at NERSC with a focus on parallel I/O, file systems, and productive data analytics on HPC. He has extensive experience in object storage systems and scalable HDF data access in Spark via H5Spark.

Phil Carns is a principle software development specialist at ANL who works on measurement, modeling, and development of data services. He has been a key contributor to the Darshan performance tool, CODES simulation toolkit, and PVFS file system.

HPC I/O 101

- HPC I/O: storing and retrieving persistent scientific data on a high performance computing platform
  - Data is usually stored on a **parallel file system**
  - The parallel file system must be capable of storing and accessing enormous volumes of data quickly!
  - Requires coordination between hardware components, system software, and applications
  - Otherwise compute resources will be idle waiting for data
- Today's material will largely be about the proper care and feeding of parallel file systems to get the most out of them.



EXASCALE COMPUTING PROJECT

Parallel file systems

- Looks just like a file system on your laptop: directories and files, open/close/read/write
- But **a parallel file system does not behave like a conventional file system**
- We'll highlight 5 key, high-level differences in this presentation
- The objective is to provide some background and motivation for the more specific optimizations and usage tips that we will cover later.

What is unique about HPC I/O?
#1: Multiple file systems to choose from on each platform

Suppose you want to pick a vehicle:

- To hold a *lot* of material
- To go as fast as possible
- To let your friends join you
- To be as safe as possible
- For a quick, short trip

It’s immediately obvious which one is going to be best for each scenario.

ECP EXASCALE COMPUTING PROJECT

#1: Multiple file systems to choose from on each platform (examples from Cori @ NERSC and Theta @ ALCF)

/project

/global

/projects

/local

/home

$HOME

HPSS

/scratch

$DW_JOB_STRIPED

$SCRATCH

Suppose you want to pick a storage system:

- To hold a *lot* of material
- To go as fast as possible
- To let your friends join you
- To be as safe as possible
- For a quick, short trip

#1: Multiple file systems to choose from on each platform (examples from Cori @ NERSC and Theta @ ALCF)

/project

/global

/projects

/local

/home

$HOME

HPSS

/scratch

$DW_JOB_STRIPED

$SCRATCH

Suppose you want to pick a storage system:

- To hold a *lot* of material
- To go as fast as possible
- To let your friends join you
- To be as safe as possible
- For a quick, short trip

Is it still obvious?

EXASCALE COMPUTING PROJECT

#1: Multiple file systems to choose from on each platform (examples from Cori @ NERSC and Theta @ ALCF)

/project
/projects
/global
/home
/local
$HOME
HPSS
/scratch
$SCRATCH
$DW_JOB_STRIPED

Suppose you want to pick a storage system:

- To hold a *lot* of material
- To go as fast as possible
- To let your friends join you
- To be as safe as possible
- For a quick, short trip

By the way, these aren’t the same thing.

Neither are these.

Use facility documentation!

https://www.alcf.anl.gov/user-guides/data-storage-file-systems
http://www.nersc.gov/users/storage-and-file-systems/file-systems/

ECP EXASCALE COMPUTING PROJECT

How to *use* available file systems

At least the differences are obvious once you sit down to use one of the options, right?

EXASCALE COMPUTING PROJECT

How to *use* available file systems

open()
close()
read()
write()

open()
close()
read()
write()

open()
close()
read()
write()

open()
close()
read()
write()

open()
close()
read()
write()

At least the differences are obvious once you sit down to use one of the options, right?

How to *use* available file systems

open()
close()
read()
write()

open()
close()
read()
write()

open()
close()
read()
write()

open()
close()
read()
write()

open()
close()
read()
write()

At least the differences are obvious once you sit down to use one of the options, right?

Not so much. This is good for portability though!

Just be alert that an application will just as easily run on a poor file system choice as it will a good file system choice.

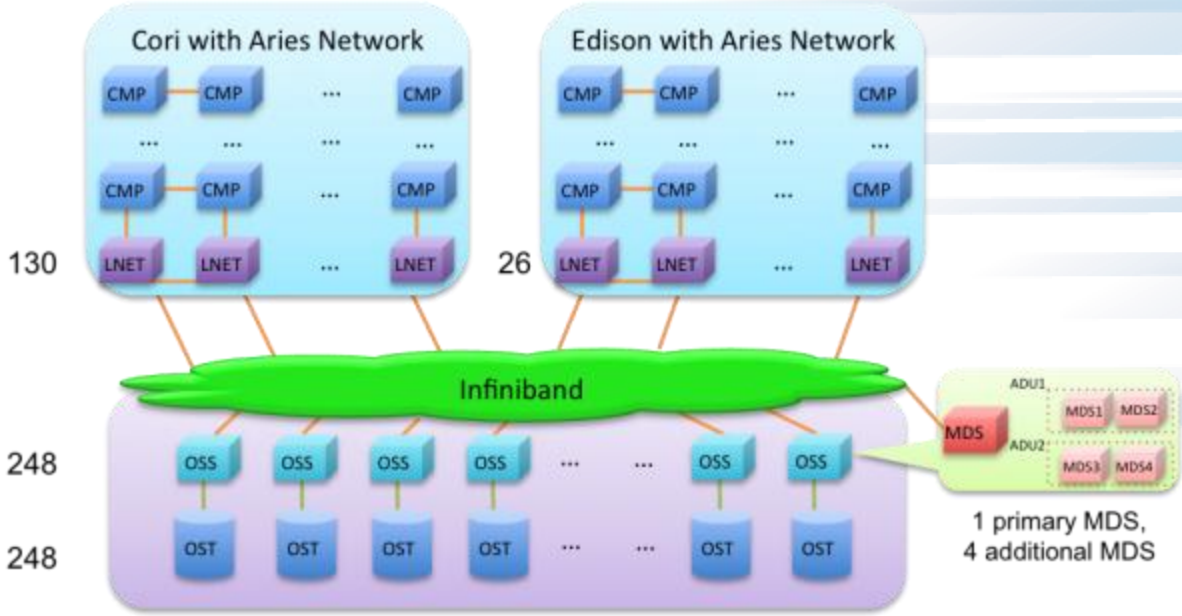
Rely on facility documentation and support team to help you pick the correct storage resources for your work.

EXASCALE COMPUTING PROJECT

What is unique about HPC I/O?
#2: the storage system is large and complex

- It looks like any other file system
- But there are 10,000 or more disk drives!
- This means that an HPC file system will often behave differently

Cori scratch file system diagram
NERSC, 2017

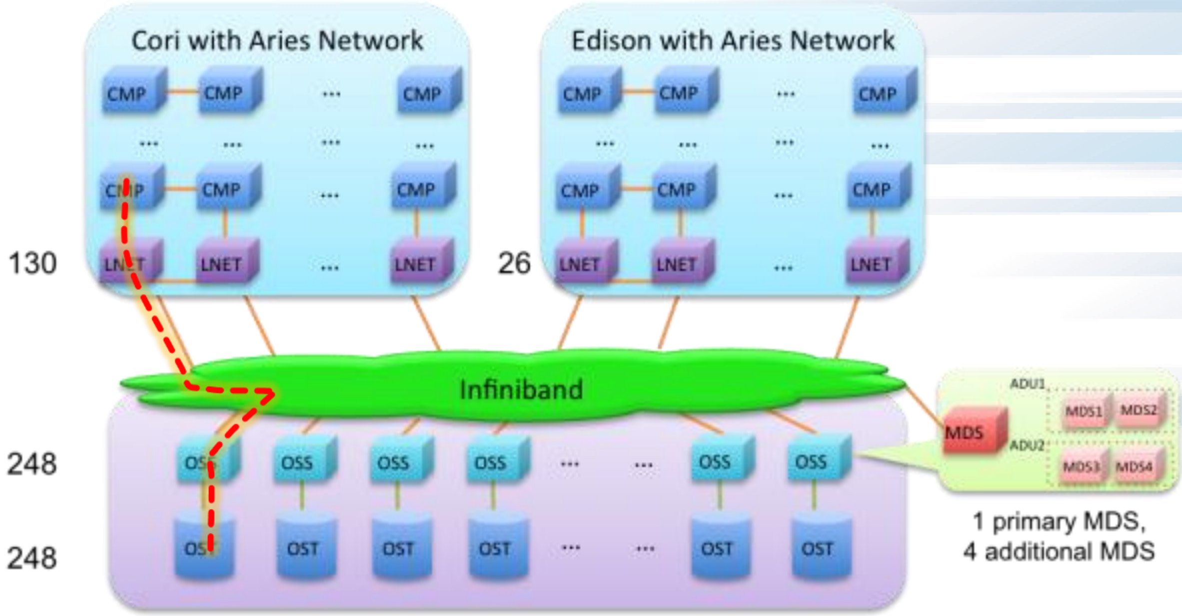


Each OSS controls one OST. The Infiniband connects the MDS, ADUs and OSSs to the LNET routers on the Cray XC System. The OSTs are configured with GridRAID, similar to RAID6, (8+2), but can restore failure 3.5 times faster than traditional RAID6. Each OST consists of 41 disks, and can deliver 240TB capacity.

What is unique about HPC I/O?
#2: the storage system is large and complex

- Moving data from one compute node to a disk drive requires several "hops"
- Therefore, the *latency*, or time to complete a single small operation by itself, is often quite poor

Cori scratch file system diagram
NERSC, 2017

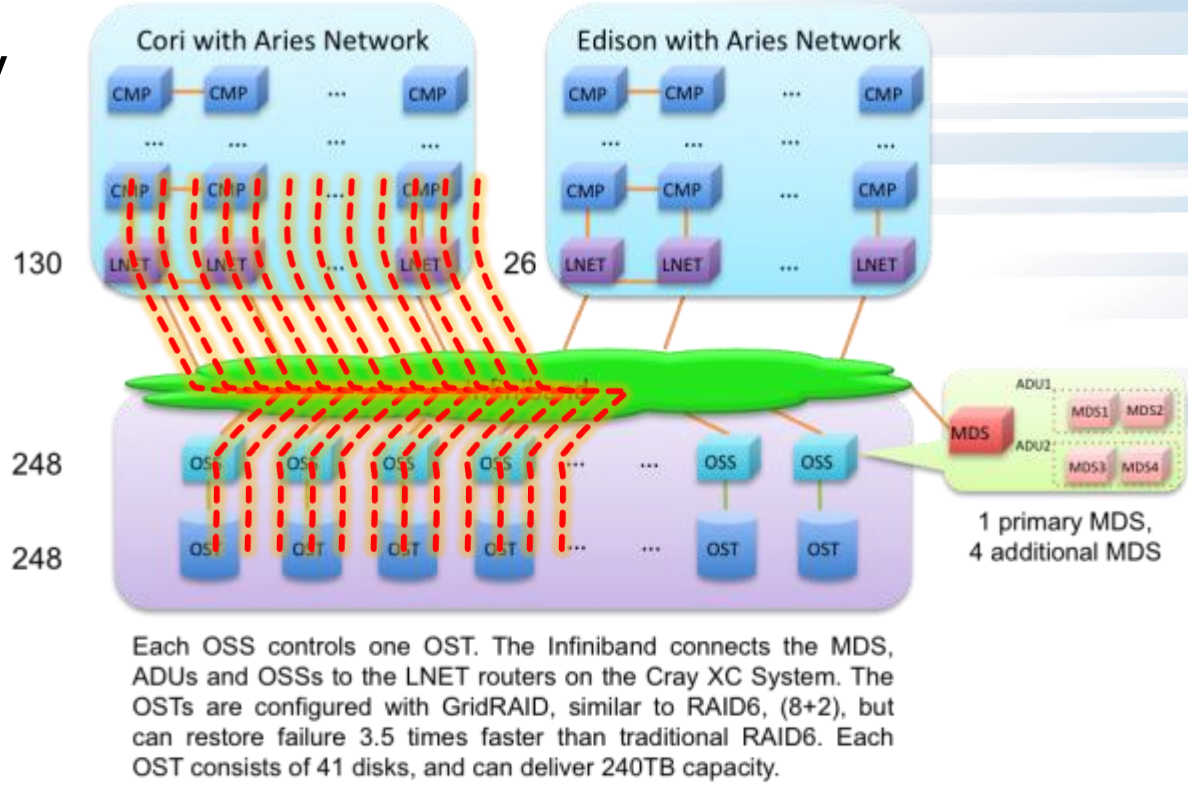


Each OSS controls one OST. The Infiniband connects the MDS, ADUs and OSSs to the LNET routers on the Cray XC System. The OSTs are configured with GridRAID, similar to RAID6, (8+2), but can restore failure 3.5 times faster than traditional RAID6. Each OST consists of 41 disks, and can deliver 240TB capacity.

What is unique about HPC I/O?
#2 the storage system is large and complex

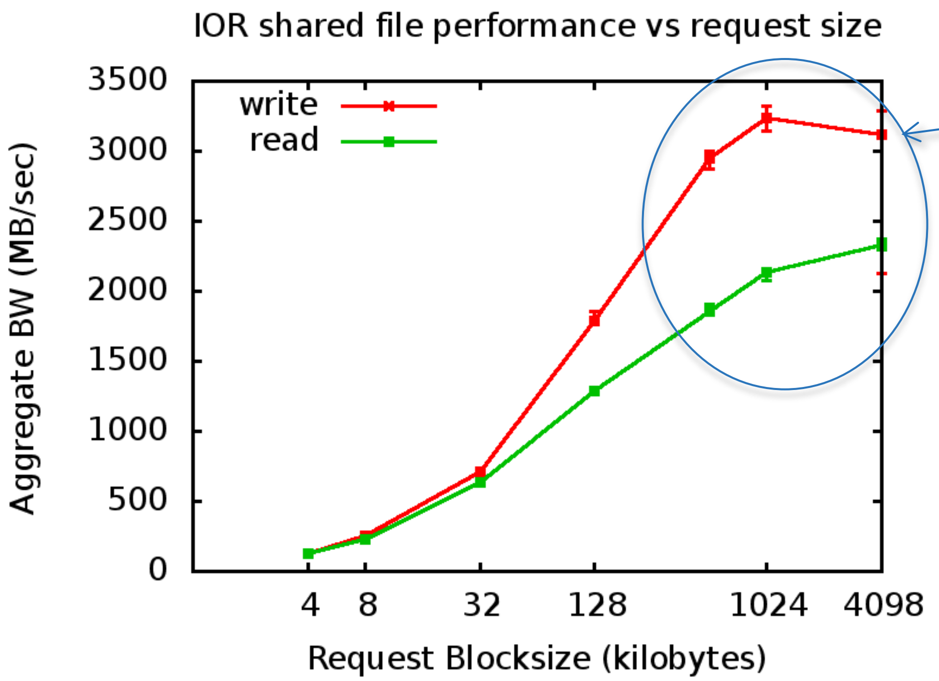
- But the network is fast, and you can do many I/O operations simultaneously
- Therefore, the *aggregate bandwidth,* or rate of parallel data access, is tremendous
- Parallel I/O tuning is all about playing to the system's strengths:
  - Move data in parallel with big operations
  - Avoid waiting on individual small operations

Cori scratch file system diagram
NERSC, 2017



More on the bandwidth and latency of parallel file systems.

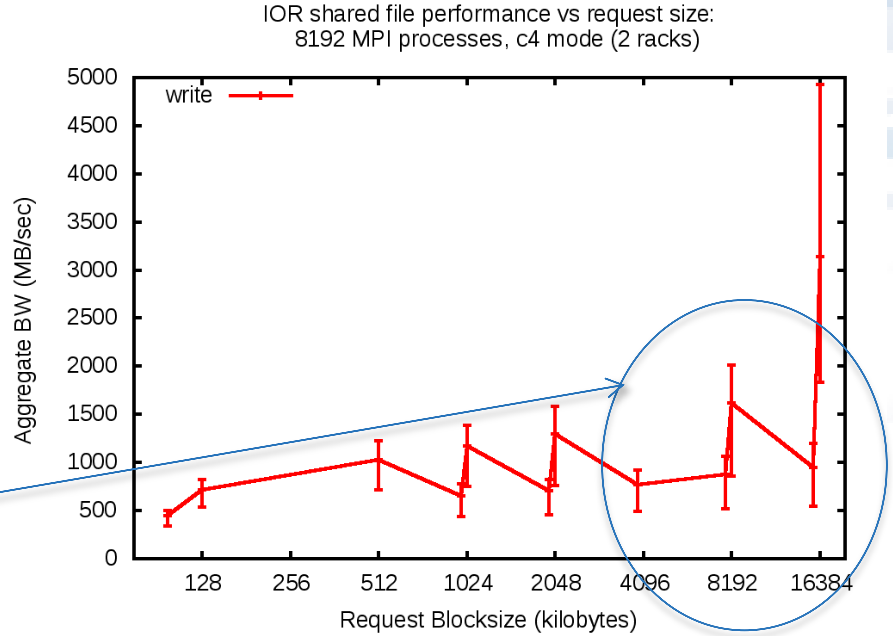
Latency has a significant impact on effective rate of I/O. The system performs best with operations in the O(Mbytes) range.



Large operations get an order of magnitude higher throughput.

2K processes of IBM Blue Gene/P at ANL.

Good alignment can quadruple throughput



8k processes of IBM Blue Gene /Q at ANL

Small operations spend too much time handshaking for the amount of work performed.

Poor alignment causes large operations to be split into smaller operations or read/modify/write operations.

Today we will talk about libraries and tools that help you hit the “sweet spot”.

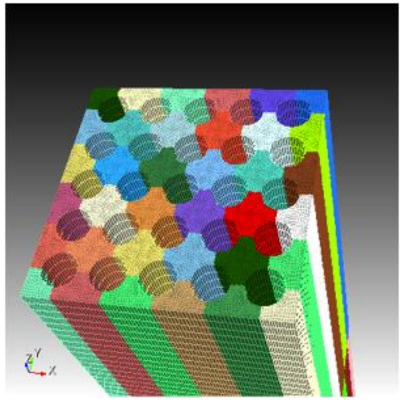
What is unique about HPC I/O?
#3 sophisticated application data models

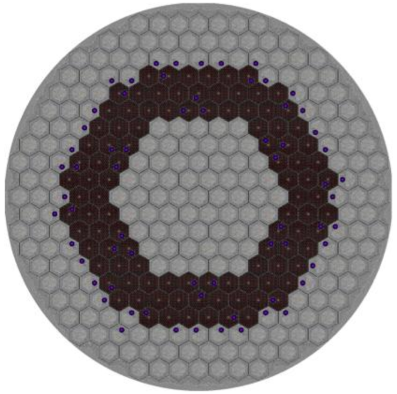
Images from T. Tautges (ANL) (upper left), M. Smith (ANL) (lower left), and K. Smith (MIT) (right).

- Applications use advanced data models that suite the problem at hand
  - Multidimensional typed arrays, images composed of scan lines, etc.
  - Headers, attributes on data
- I/O systems have very simple data models
  - Tree-based hierarchy of containers
  - Containers with streams of bytes (files)
  - Containers listing other containers (directories)

Data libraries help to map application data models to files and directories in an optimal, portable way.

We'll learn more about this as the day goes on too!





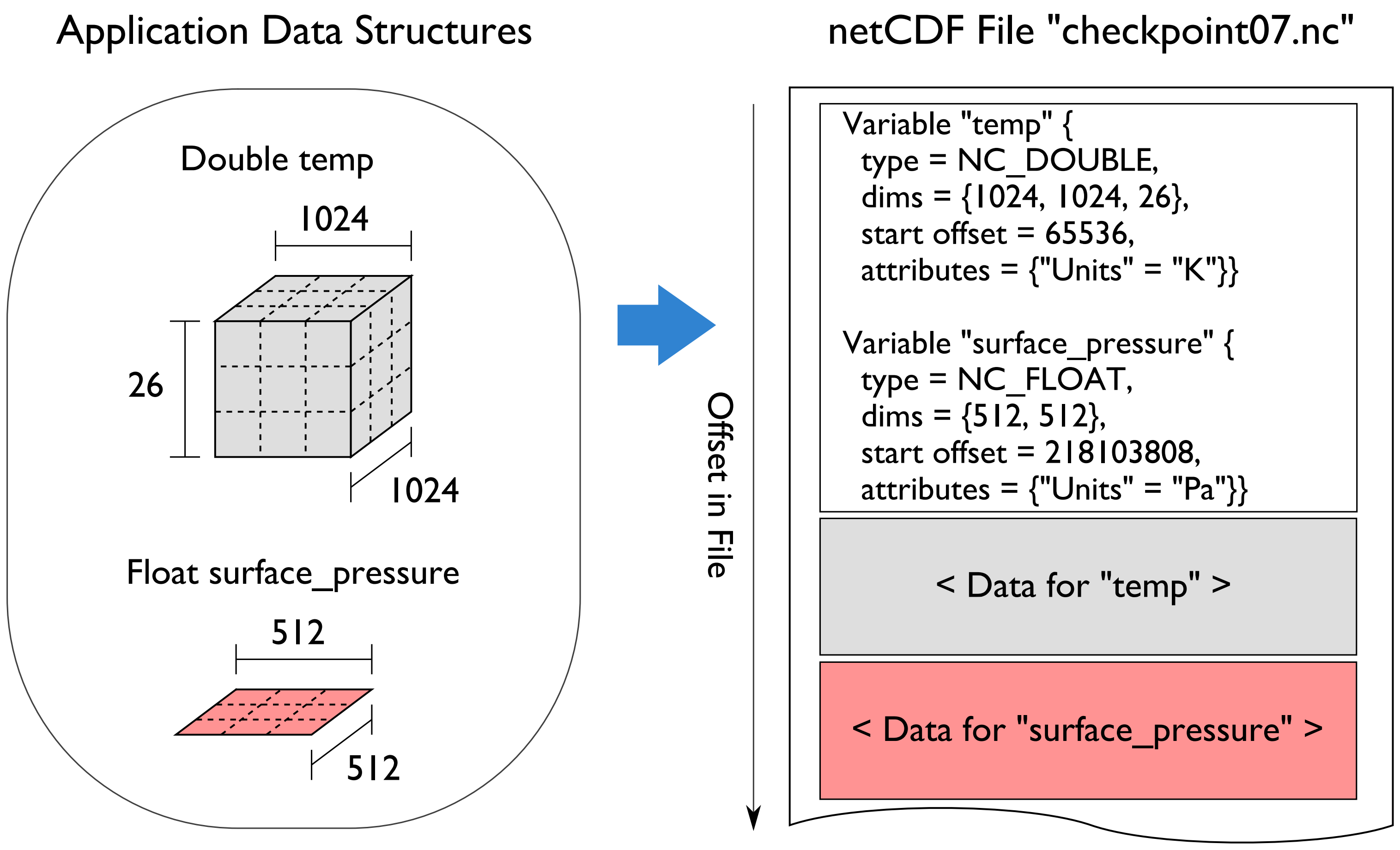
Model complexity:
Spectral element mesh (top) for thermal hydraulics computation coupled with finite element mesh (bottom) for neutronics calculation.

Scale complexity:
Spatial range from the reactor core in meters to fuel pellets in millimeters.

EXASCALE COMPUTING PROJECT

18

Example of organizing application data



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.

19

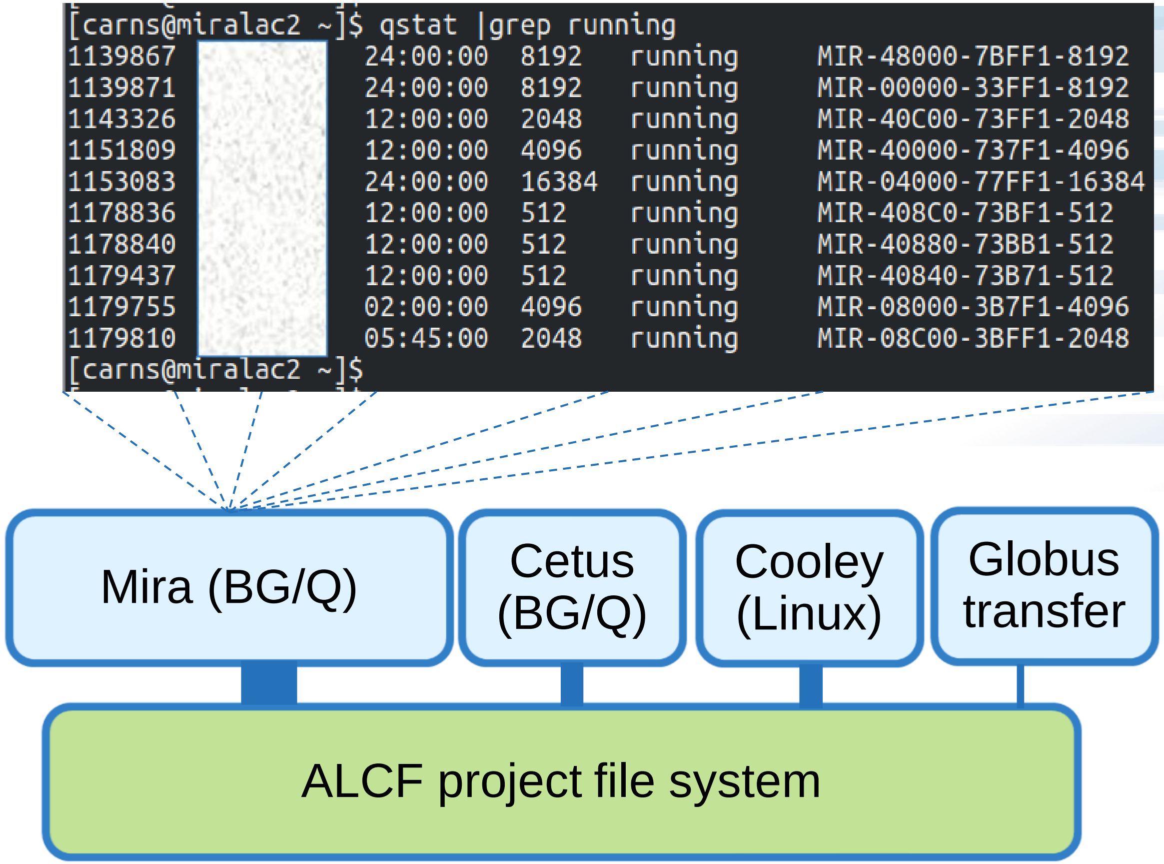
What is unique about HPC I/O?
#4: each HPC facility is different

… and more

- HPC systems are purpose-built by a few different vendors.
- Their storage systems are purpose-built as well, and each system has its own hardware, software, and performance characteristics.
- Use portable tools and libraries to handle portable platform optimizations, learn performance debugging basics (more later).

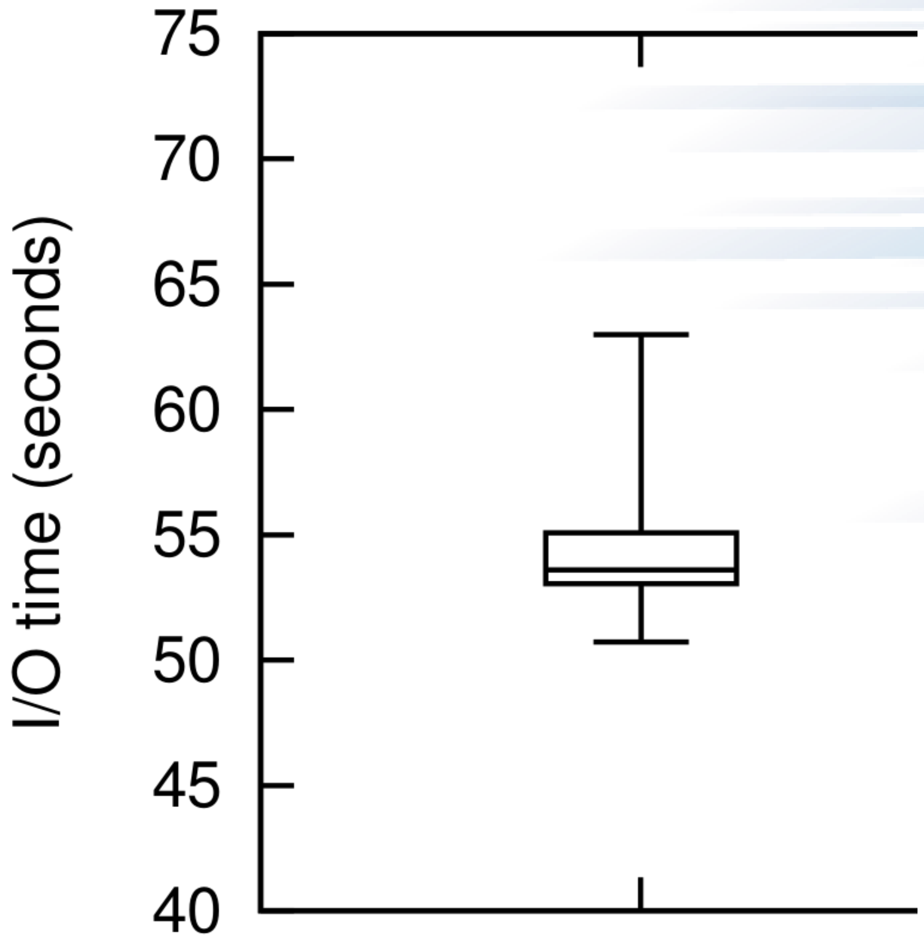
What is unique about HPC I/O?
#5: Expect some performance variability

- Why:
  - Thousands of hard drives will *never* perform perfectly at the same time.
  - You are sharing storage with many other users.
  - You are sharing storage with remote transfers, tape archives, and other data management tasks.
  - You are sharing storage across multiple HPC systems.
- Some performance variance is normal



What is unique about HPC I/O?
#5: Expect some performance variability

- When measuring I/O performance, take multiple samples and look for trends over time.
- Example shows 15 samples of I/O time from a 6,000 process benchmark on Edison system, with a range of 51 to 63 seconds
- We will have a hands-on exercise later in the day that you can use to investigate this phenomenon first hand.



EXASCALE COMPUTING PROJECT

Putting it all together for HPC I/O happiness

1. Consult your facility documentation to find appropriate storage resources
2. Move big data in parallel, and avoid waiting for individual small operations
3. Use I/O libraries that are appropriate for your data model
4. Learn some performance debugging tools and techniques that you can reuse across systems
5. Be aware that I/O performance fluctuates on individual jobs for reasons that you cannot control

But wait, there’s more!

#6: Improving I/O performance is an ongoing process.

You have to monitor and adapt periodically over time. Contributors or students modify your application, your science objectives change, you are awarded hours on a new machine, etc.

One way to think of this: the OODA loop concept from strategy and control theory.

- **Observe:** instrument applications and systems
- **Orient:** interpret performance data in context
- **Decide:** determine how improve
- **Act:** implement improvements

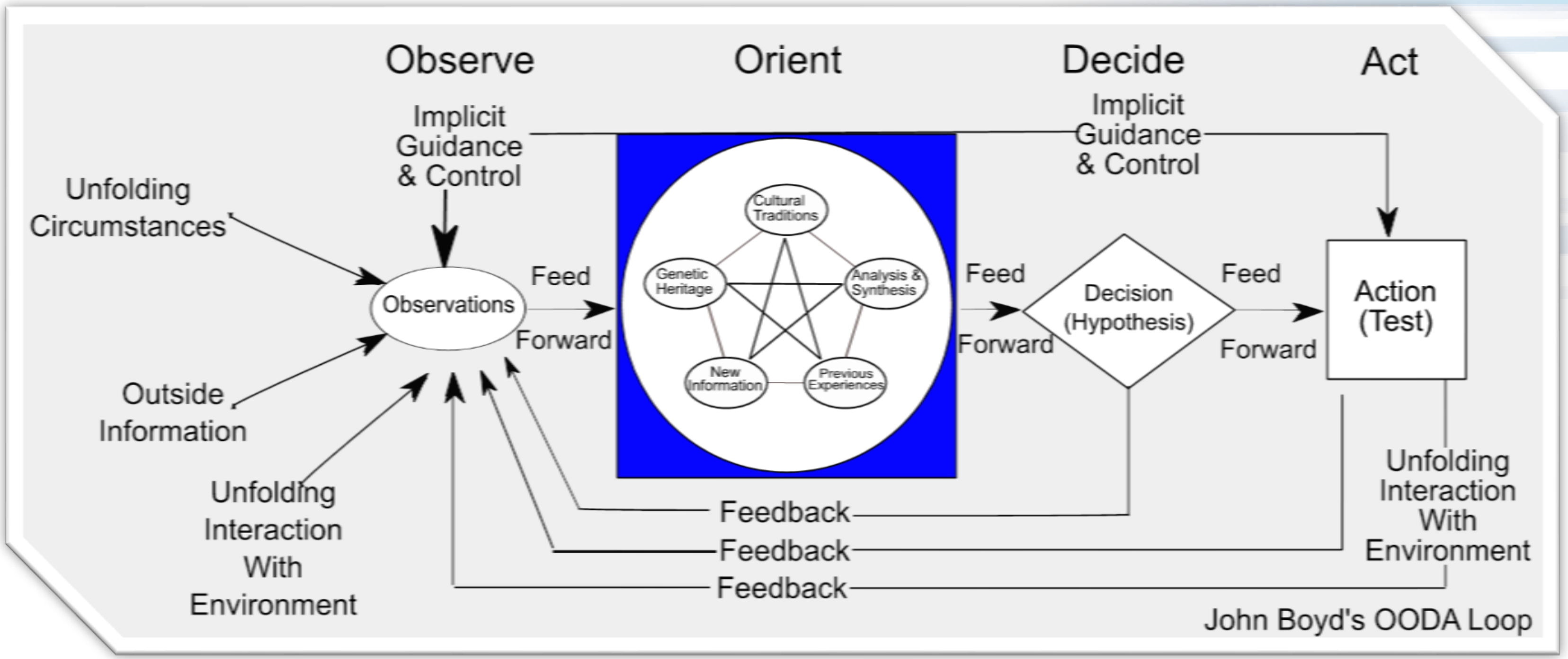


Figure by Patrick Edwin Moran

Improving I/O performance is an ongoing process

We will try to equip you with the tools you need to monitor and improve your I/O performance.

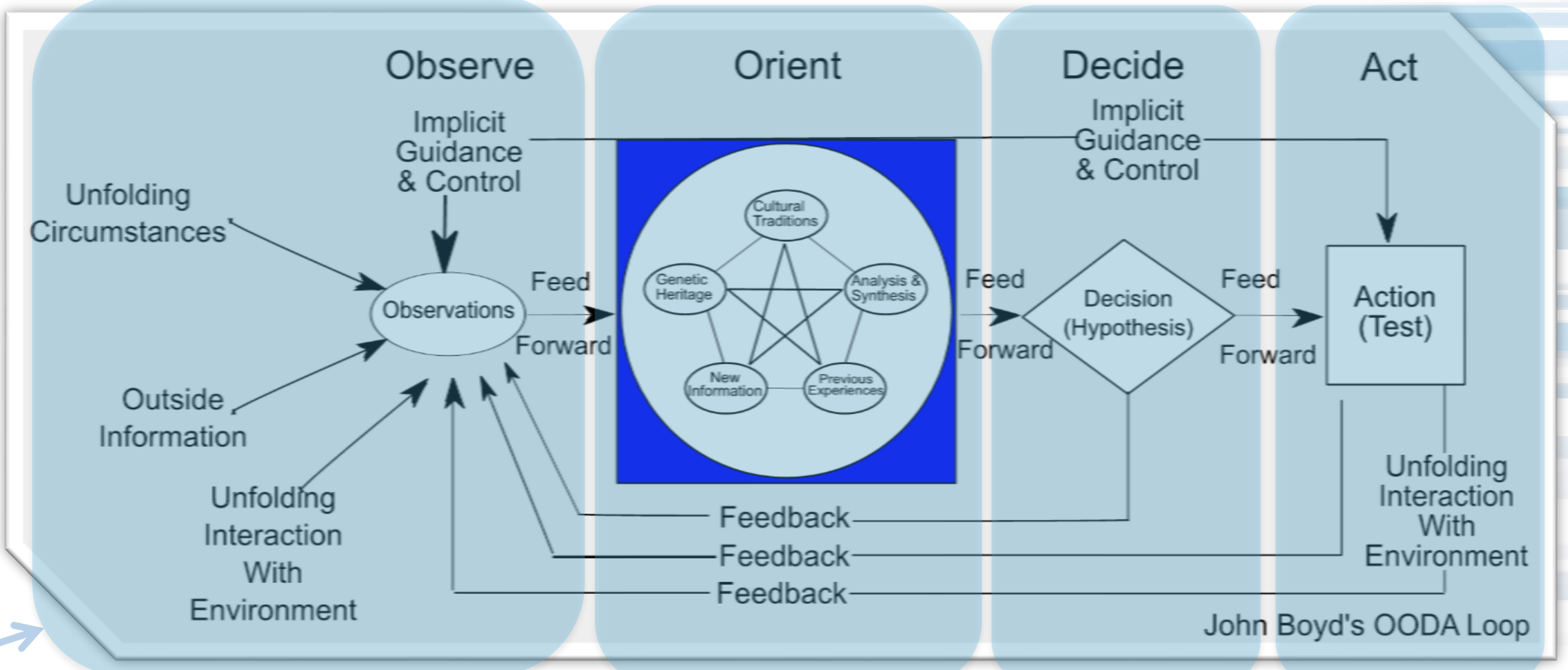


Figure by Patrick Edwin Moran

Performance characterization tools, like Darshan

Background knowledge that you'll learn today

Help from facility resources

Optimization techniques, tools, and libraries.

HOW IT WORKS: WHAT IS A REAL HPC STORAGE SYSTEM LIKE?

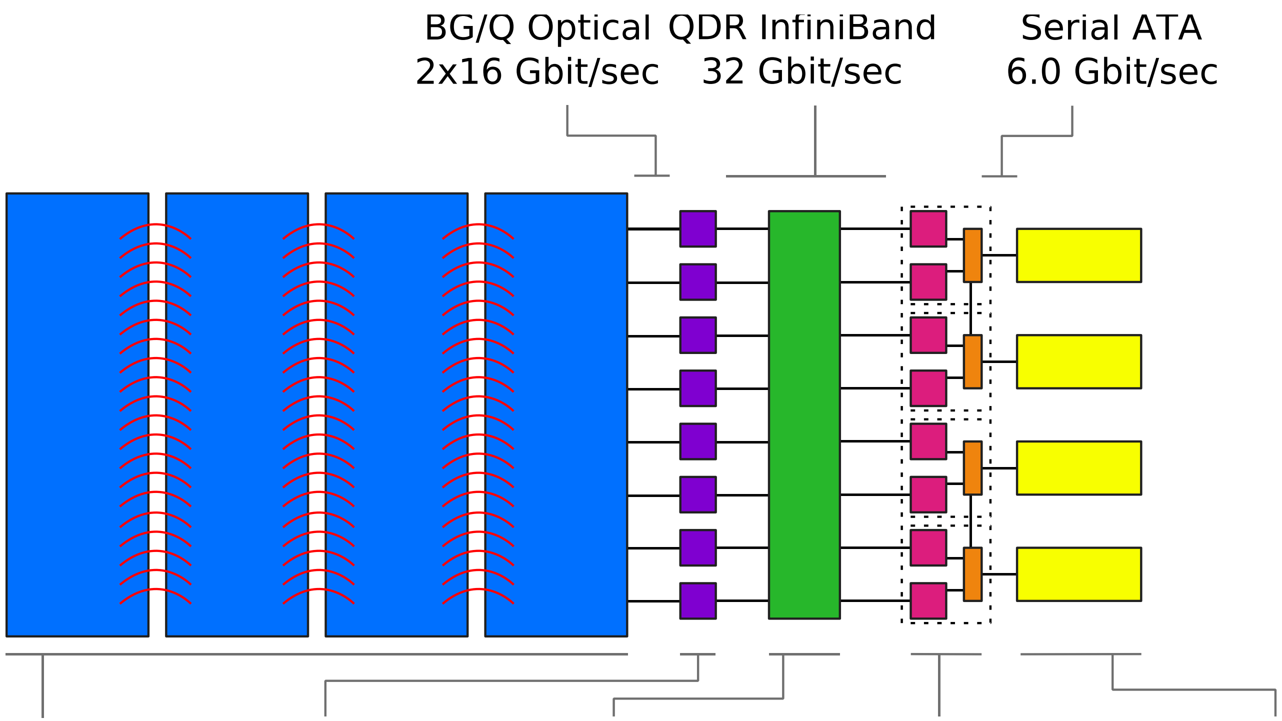
An example system: Mira (ALCF)

Mira is the flagship HPC system at Argonne National Laboratory

- 48 racks
- 786,432 processors
- 768 terabytes of memory

“Mira is 20 times faster than Intrepid, its IBM Blue Gene/P predecessor”

Mira storage hardware layout



Compute nodes run applications and some I/O middleware.

768K cores with 1 Gbyte of RAM each

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

384 16-core PowerPC A2 nodes with 16 Gbytes of RAM each

Commodity network primarily carries storage traffic.

QDR Infiniband Federated Switch (3024 ports)

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

SFA12KE hosts VM running GPFS servers

Enterprise storage controllers and large of disks are connecte InfiniBand.

32 DataDirect SFA12KE; 560 3 Tbyte drives + 32 200 GB SSD; 16 InfiniBand ports per pair

Largest file system (mira-fs0)

- 16 DDN storage systems
- 8,960 SATA disks
- 512 SSDs
- 12 PiB formatted storage
- 240 GiB/s performance

The Mira I/O stack

The “I/O stack” is the collection of software that transforms the application’s data model access into device operations. It has a few layers.

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.

IBM Spectrum Scale (GPFS)

- Application
- Data Model Support
- Transformations
- Parallel File System
- I/O Hardware

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

MPI-IO

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

The Mira I/O stack

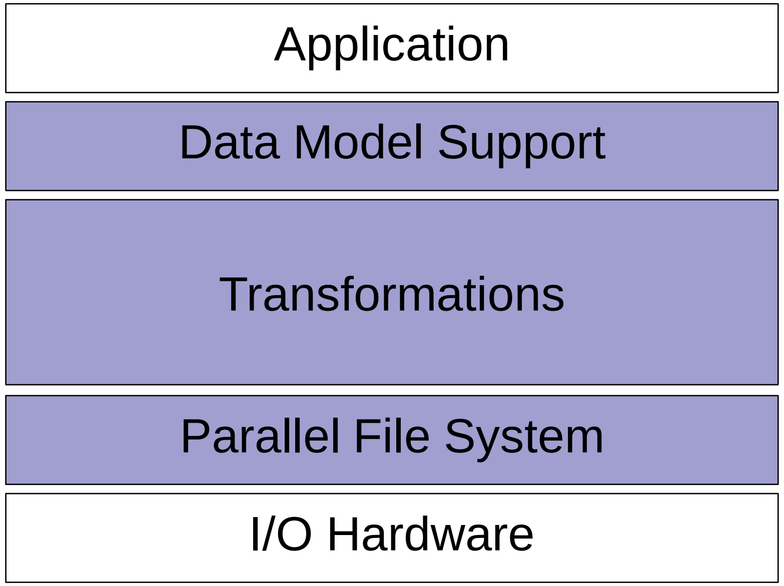
The I/O stack has a lot of software components (not to mention hardware), but data model libraries protect users from most of the complexity.

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.

GPFS



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

MPI-IO

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

What about Theta?

Key parts of the software and hardware stack are different

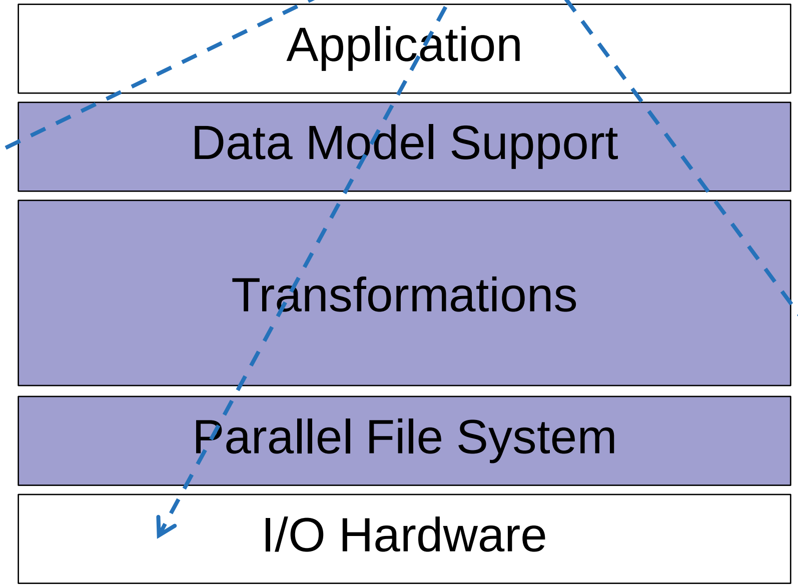
Different optimizations are needed to account for block sizes, storage device types, locking algorithms, etc.

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.

Lustre



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

MPI-IO

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.

Lnet routers

The high level library APIs used by applications are still the same, though!

EXASCALE COMPUTING PROJECT

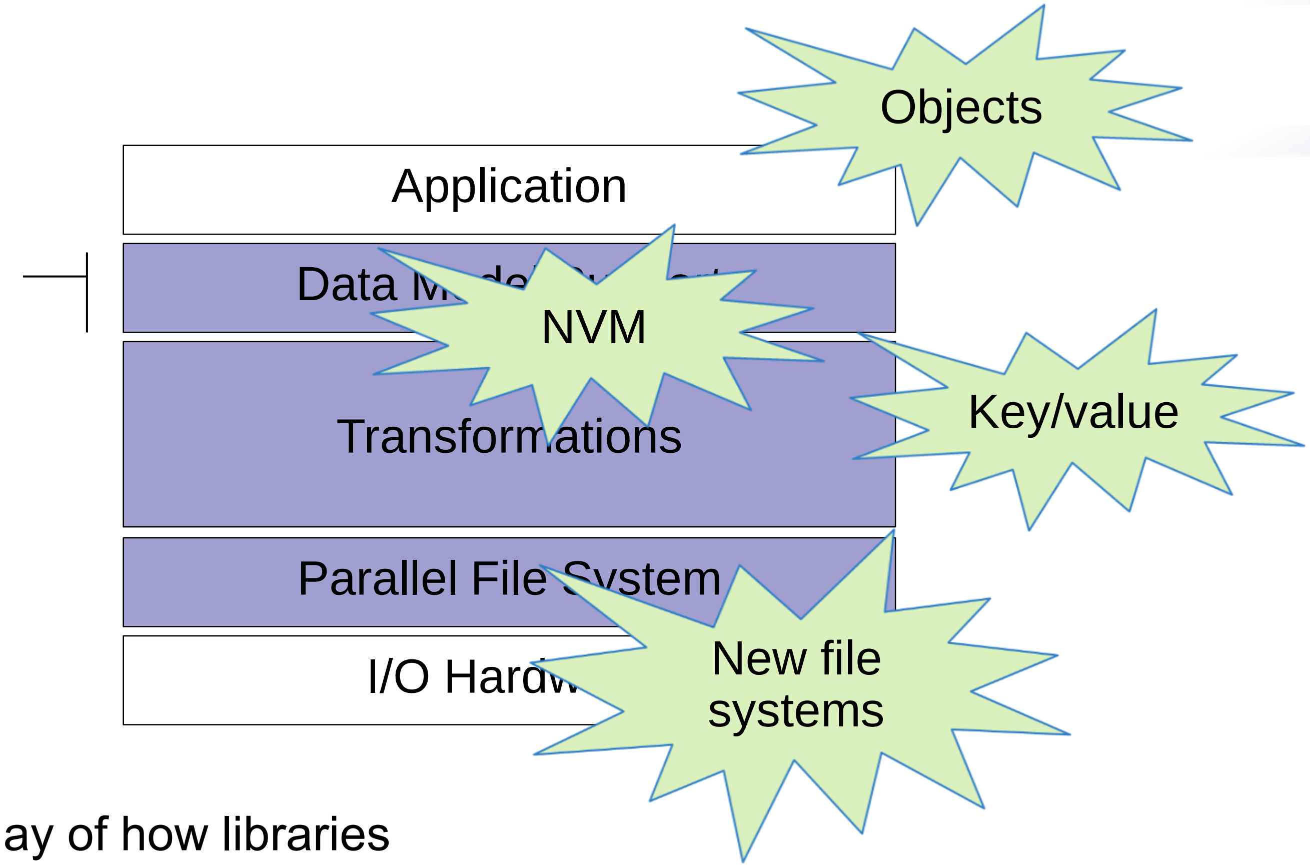
31

What about the future?

Choosing the right libraries and interfaces for your application isn't just about fitting your data model, but also future-proofing your application.

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

HDF5, Parallel netCDF, ADIOS



We'll see examples later in the day of how libraries are adapting to storage technology.

EXASCALE COMPUTING PROJECT

Next up!

The next presentation by Jialin Liu will cover **I/O Topologies.**

<u>System reservations for use throughout the day</u>

Cori.nersc.gov

- 9am – 1pm, atpesc18-haswell queue: 40 Haswell nodes
- 9am – 1pm, atpesc18-knl queue: 40 KNL nodes

Theta.alcf.anl.gov

- 11:15am – 5:30pm, training queue: 77 nodes
- 6:30pm – 9:30pm, training queue: 152 nodes

You can also submit jobs to the general or debugging queues at any time; these are just reserved nodes with faster turnaround.

Thank you!