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

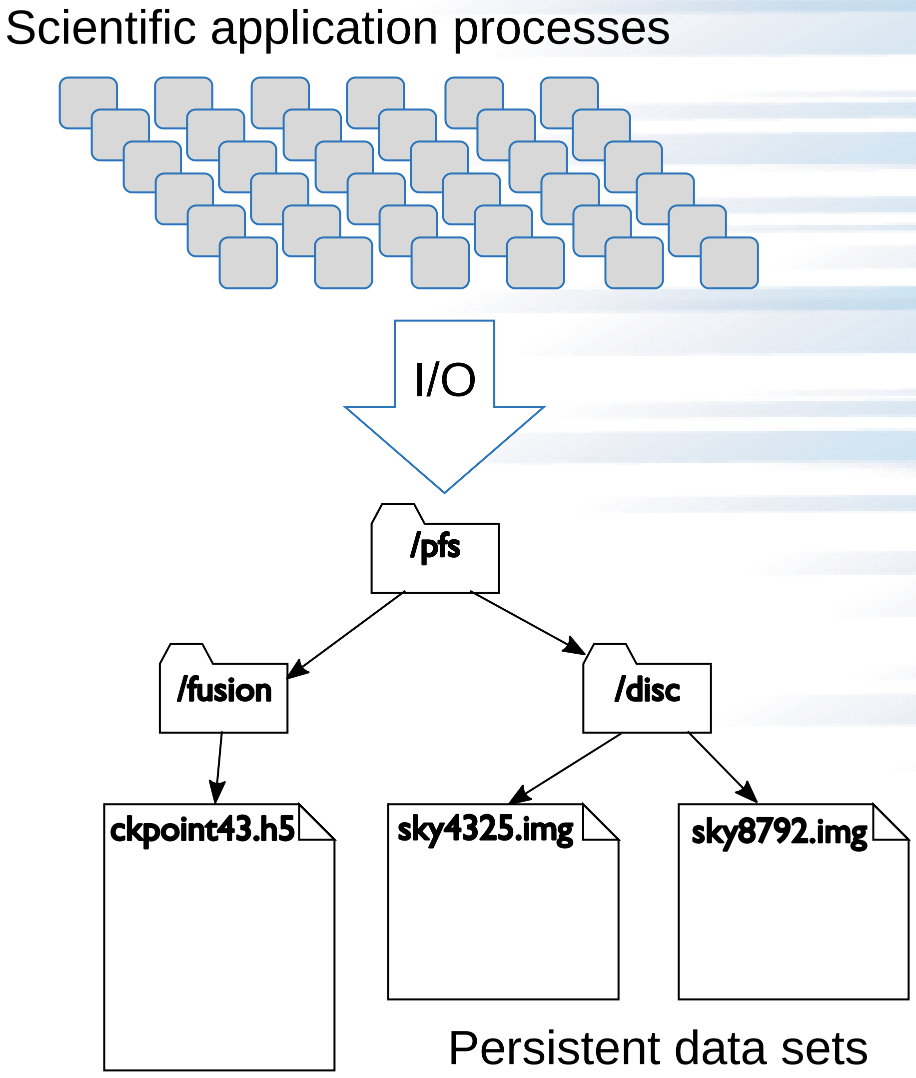
ATPESC 2021

Phil Carns
Mathematics and Computer Science Division
Argonne National Laboratory

August 6, 2021

Hands on exercises: https://github.com/radix-io/hands-on

What is HPC I/O?



- HPC I/O: storing and retrieving persistent scientific data on a high performance computing platform
  - Data is usually stored on a **parallel file system.**
  - Parallel file systems can quickly store and access enormous volumes of data.
  - They carefully orchestrate data movement between applications, system software, and storage hardware.
  - *It's an important job! Valuable CPU time is wasted if an application spends too long waiting for data.*
- Today's lectures are all about the proper care and feeding of exotic parallel file systems.

Parallel file systems

- A parallel file system *looks* just like the file system on your laptop:
    - directories and files, open/close/read/write.
- However, **parallel file systems do not behave like conventional file systems.**
- This presentation will highlight 5 crucial high-level differences.
- We'll revisit these general concepts throughout the day as we cover more specific optimization and usage tips.

What is unique about HPC I/O?
#1: You can select between several file systems on each platform

If file systems were vehicles, which one would you pick:

- To hold a *lot* of cargo
- To go as fast as possible
- To bring your friends with you
- To be as safe as possible
- To make quick, short trip

With vehicles, the choice is pretty intuitive.

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

/project

/global

/projects

/local

/home

$HOME

HPSS

/scratch

$DW_JOB_STRIPED

$SCRATCH

Now pick a storage system:

- To hold a lot of cargo
- To go as fast as possible
- To bring your friends with you
- To be as safe as possible
- To make a quick, short trip

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

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

/project

/global

/projects

/local

/home

$HOME

HPSS

/scratch

$SCRATCH

$DW_JOB_STRIPED

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

Neither are these.

Now pick a storage system:

- To hold a lot of cargo
- To go as fast as possible
- To bring your friends with you
- To be as safe as possible
- To make a quick, short trip

Use facility documentation!

https://www.alcf.anl.gov/support-center/theta/theta-file-systems
https://docs.nersc.gov/filesystems/
https://docs.olcf.ornl.gov/data/storage_overview.html

Argonne NATIONAL LABORATORY

EXASCALE COMPUTING PROJECT

How to *use* available vehicles

Can you tell what kind of vehicle you have by looking at it’s interface?

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()

Can you tell what kind of file system you have by looking at its interface?

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()

Can you tell what kind of file system you have by looking at its interface?

Not so much. This is good for portability, though!

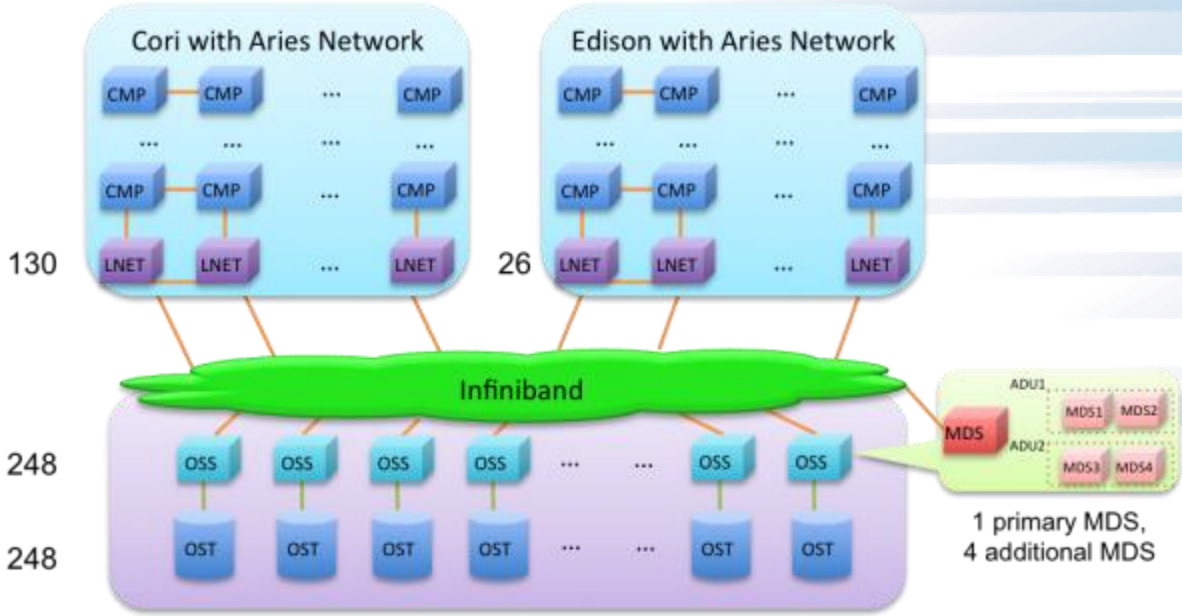
Be alert: an applications will work *correctly* on many different file systems, but they will work *best* on the one that is optimized for your goals.

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

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

- A large parallel file system looks like any other file system.
- But there are 10,000 or more disk drives!
- Specialized hardware and software is used to aggregate them into a coherent whole.
- Because of this internal difference, parallel file systems might not behave how you expect them to.

Cori scratch file system diagram
NERSC, 2017



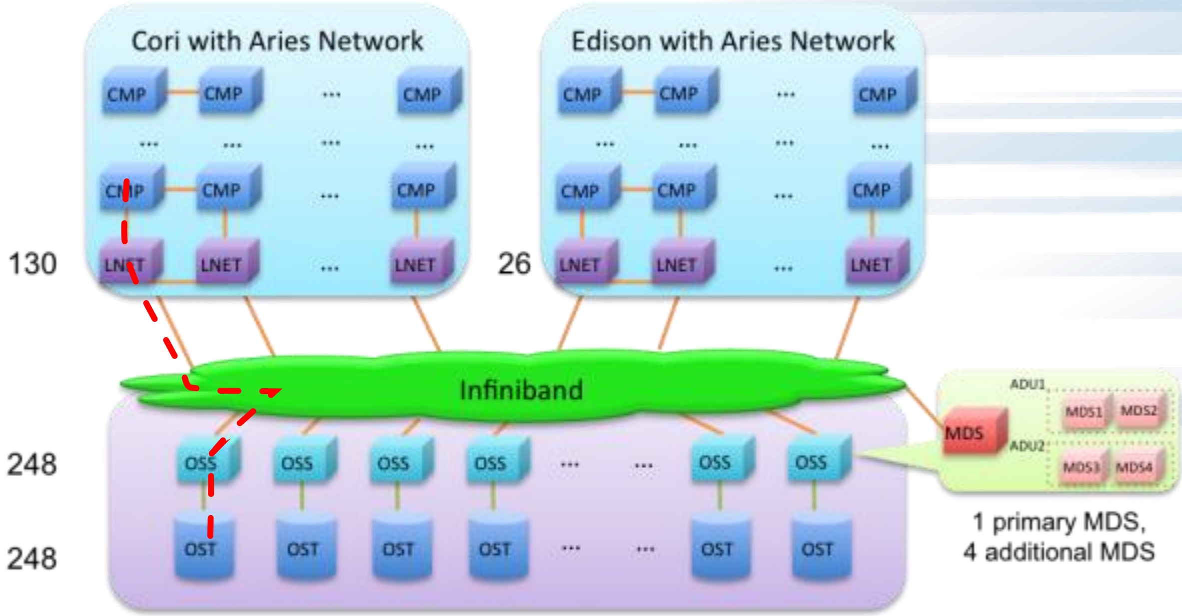
Each OSS controls one OST. The Infiniband connects the MDS, ADUs and OSSs to the LNET routers on the Cray XC System. The OSTs are configured with GridRAID, similar to RAID6, (8+2), but can restore failure 3.5 times faster than traditional RAID6. Each OST consists of 41 disks, and can deliver 240TB capacity.

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

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

- Moving data from a CPU to a disk drive requires several network "hops."
- Therefore, the ***latency***, or time to complete a single small operation, can actually be quite poor.
- This sounds like a bad thing (and to be honest, it is), but what's the silver lining?

Cori scratch file system diagram
NERSC, 2017

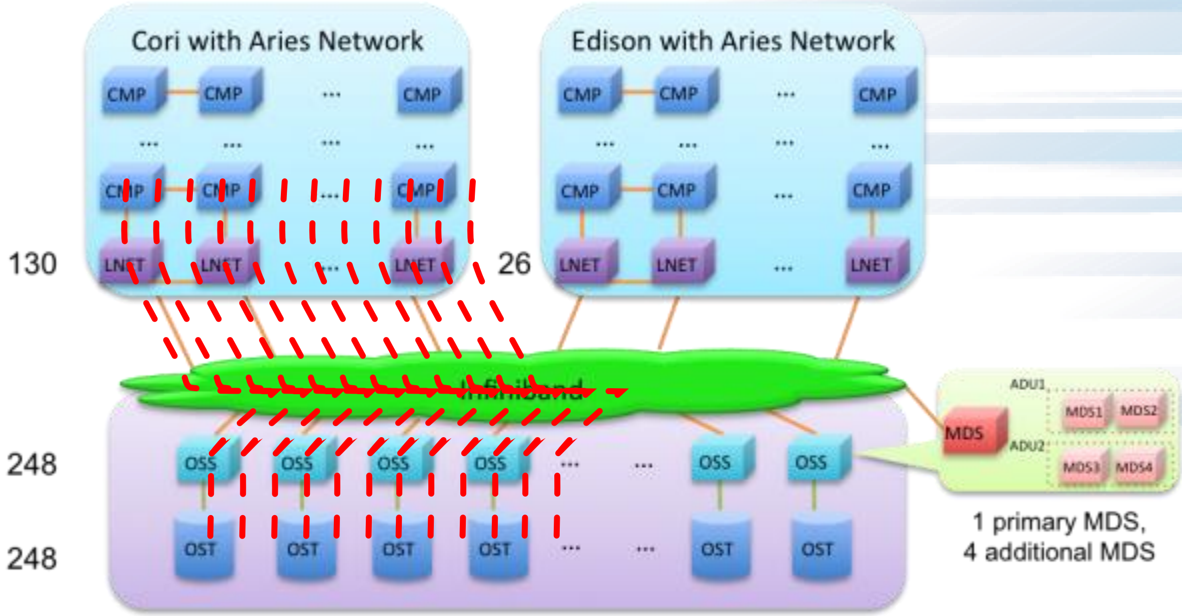


Each OSS controls one OST. The Infiniband connects the MDS, ADUs and OSSs to the LNET routers on the Cray XC System. The OSTs are configured with GridRAID, similar to RAID6, (8+2), but can restore failure 3.5 times faster than traditional RAID6. Each OST consists of 41 disks, and can deliver 240TB capacity.

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

- The network is fast, and you can perform 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 for individual small operations

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.

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

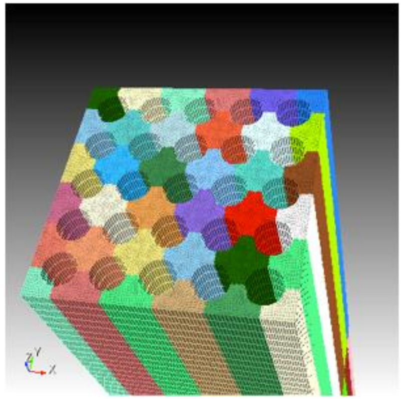
What is unique about HPC I/O?
#3 sophisticated application data models

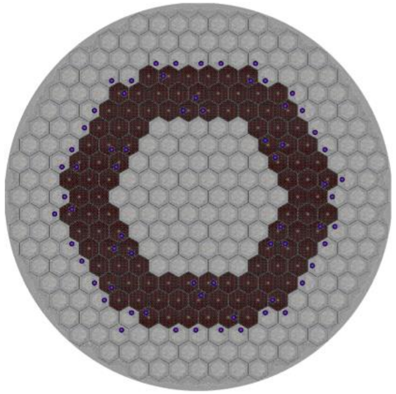
Hands on exercises: https://github.com/radix-io/hands-on

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

- Applications use advanced data models according to their scientific objectives
  - The data itself: Multidimensional typed arrays, images composed of scan lines, etc.
  - Descriptions of data (metadata): Headers, attributes, time stamps, etc.
- In contrast, parallel file systems present a very simple data model:
  - Tree-based hierarchy of containers
  - Containers with streams of bytes (files)
  - Containers listing other containers (directories)

You could map between these two models yourself: "The frequency attribute is an 8-byte float in GHz, stored at offset 4096."





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.

Argonne NATIONAL LABORATORY ECP EXASCALE COMPUTING PROJECT

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

Hands on exercises: https://github.com/radix-io/hands-on

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

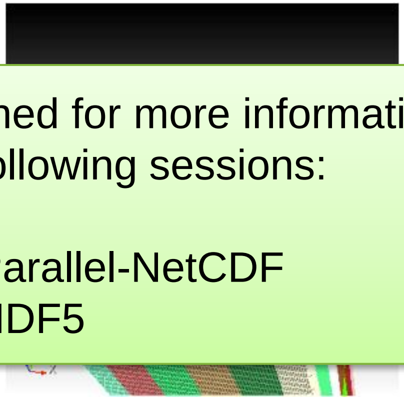
Data libraries (like HDF5, PnetCDF, and ADIOS) help to bridge this gap between application data models and file system interfaces.

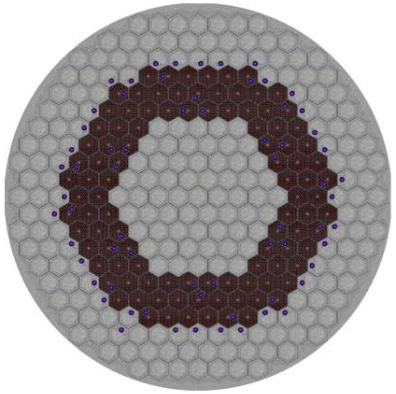
Why use a high level data library?

- More expressive interfaces for scientific data
  - e.g., multidimensional variables and their descriptions
- Interoperability
  - e.g., enables collaborators to share data in known formats
- Performance
  - e.g., high level libraries hide the details of platform-specific optimizations
- Future proofing
  - e.g., interfaces and data formats that outlive specific storage technologies

Stay tuned for more information in the following sessions:

11:30 Parallel-NetCDF
12:15 HDF5





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.

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

CRAY DATAWARP™

IBM Spectrum Scale

panasas®

lustre®

… and more

- HPC systems are custom-built by a handful of specialized vendors.
- Their storage systems are custom-built as well
  - Different hardware
  - Different software
  - → **Different performance characteristics**
- Use portable tools and libraries (see previous slide) to handle platform-specific optimizations.
- Learn performance debugging principles that can be applied anywhere.

Each HPC facility is different: Mira / ALCF example (previous gen)

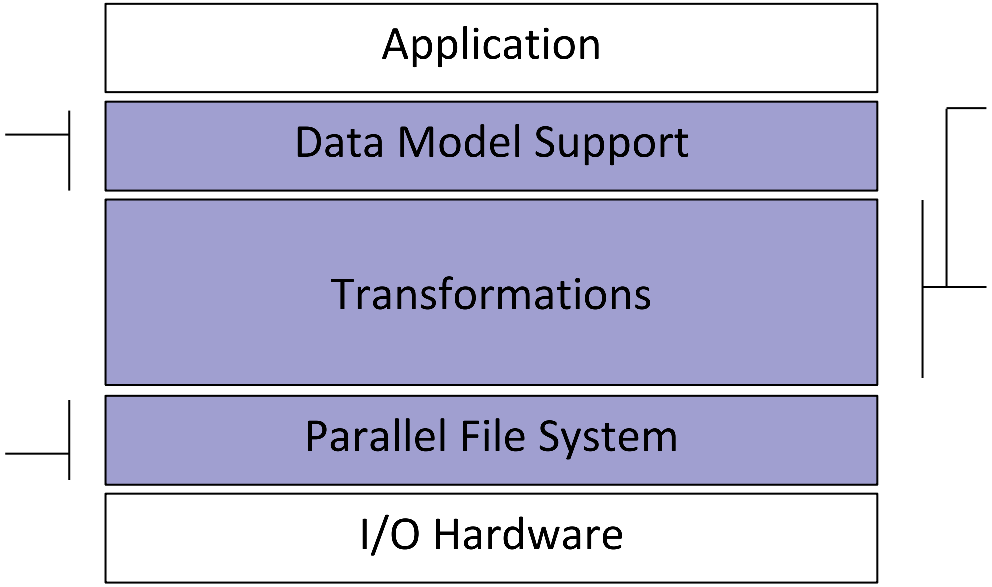
The “I/O stack” is the collection of software that translates application data access into storage system operations. It has a few layers.

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)



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.

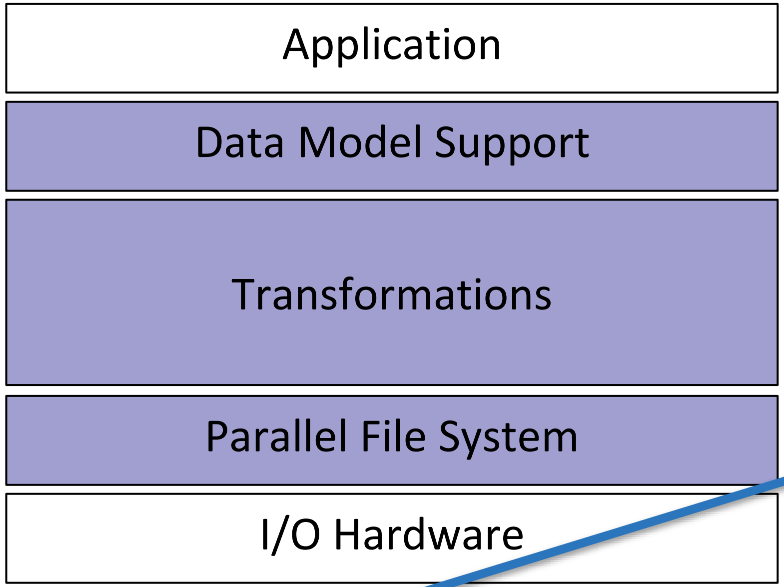
IBM ciod

Each HPC facility is different: Theta / ALCF example (current gen)

The application interface doesn't change.

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

HDF5, Parallel netCDF, ADIOS



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

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

Lustre

… even though some system details are very different!

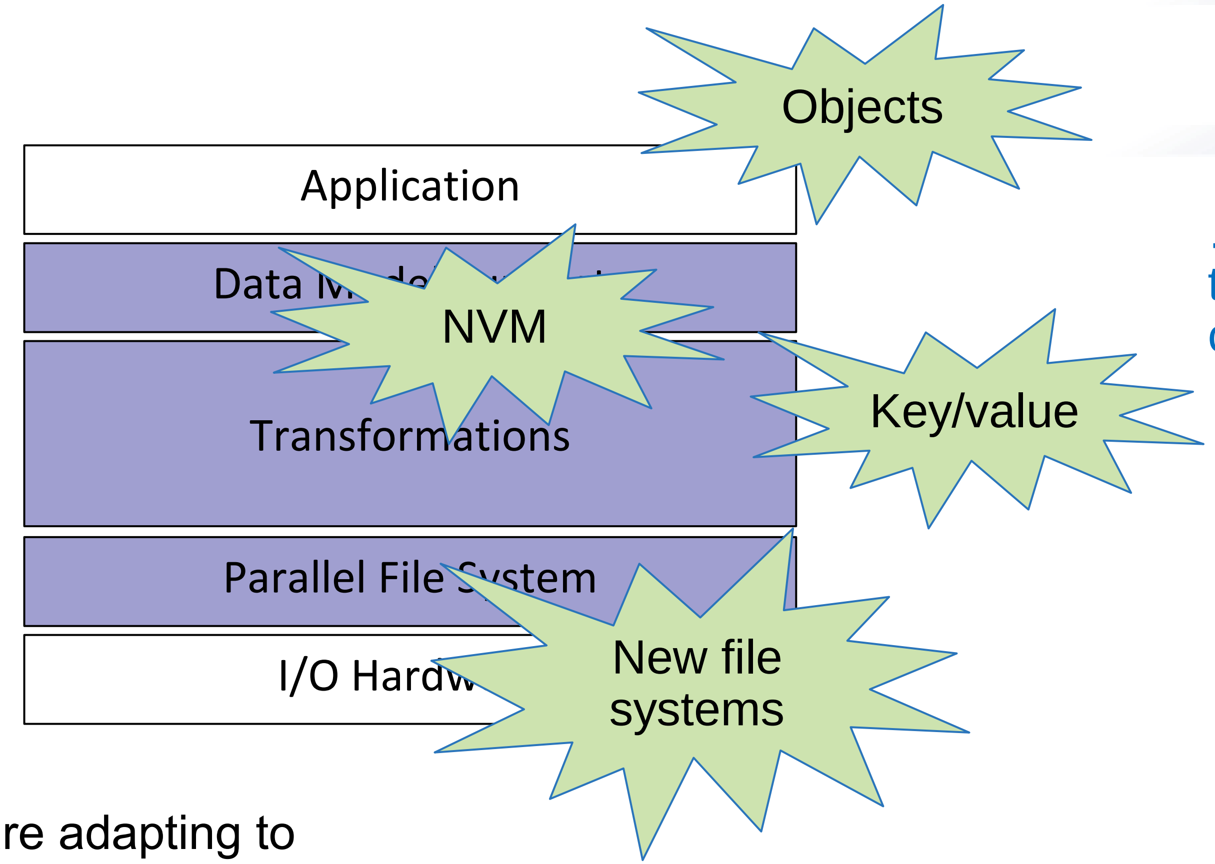
Each HPC facility is different: future machine example (next gen)

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

The interface is still familiar.

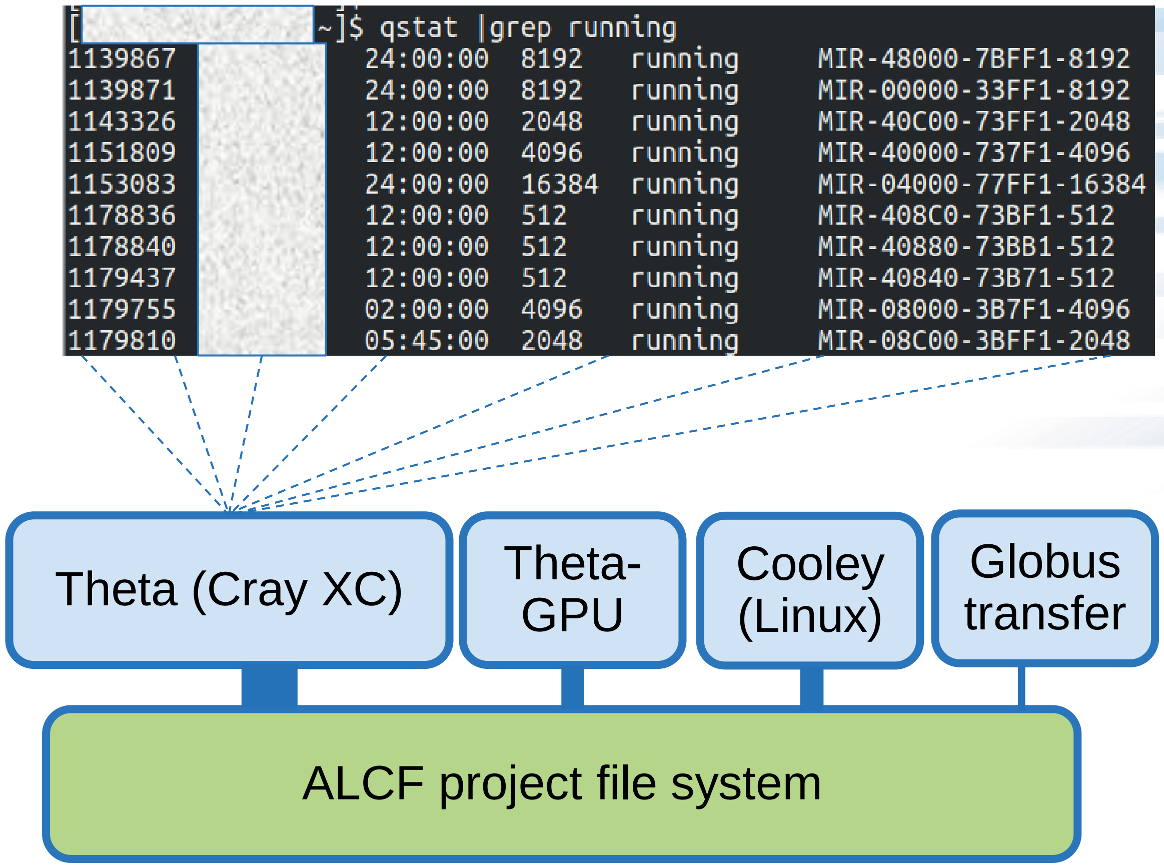


… even as technology changes.

We'll see examples of how libraries are adapting to storage technology later today.

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

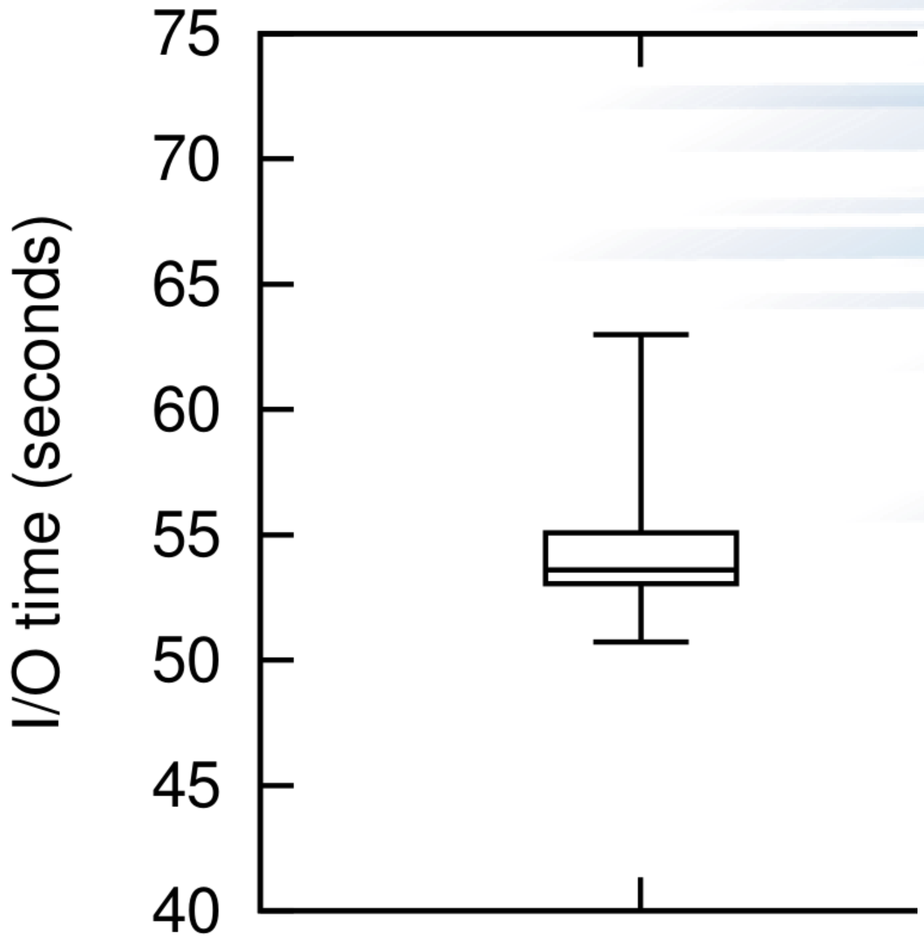
- Why:
  - Thousands of hard drives will *never* perform perfectly at the same time.
  - You are sharing storage with many other users across multiple HPC systems.
  - You are also sharing storage with remote transfers, tape archives, and other data management tasks.
- Compute nodes belong exclusively to you during a job allocation, but the storage system does not.
- Performance variance is normal.



Argonne NATIONAL LABORATORY
EXASCALE COMPUTING PROJECT

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

- Take multiple samples when measuring I/O performance.
- This figure shows 15 samples of I/O time from a 6,000 process benchmark on the (now retired) Edison system.
- How do you assess if a change in your application helped or hurt performance? You have to consider natural variability as well.
- We will have a hands-on exercise later in the day that you can use to investigate this phenomenon first hand.

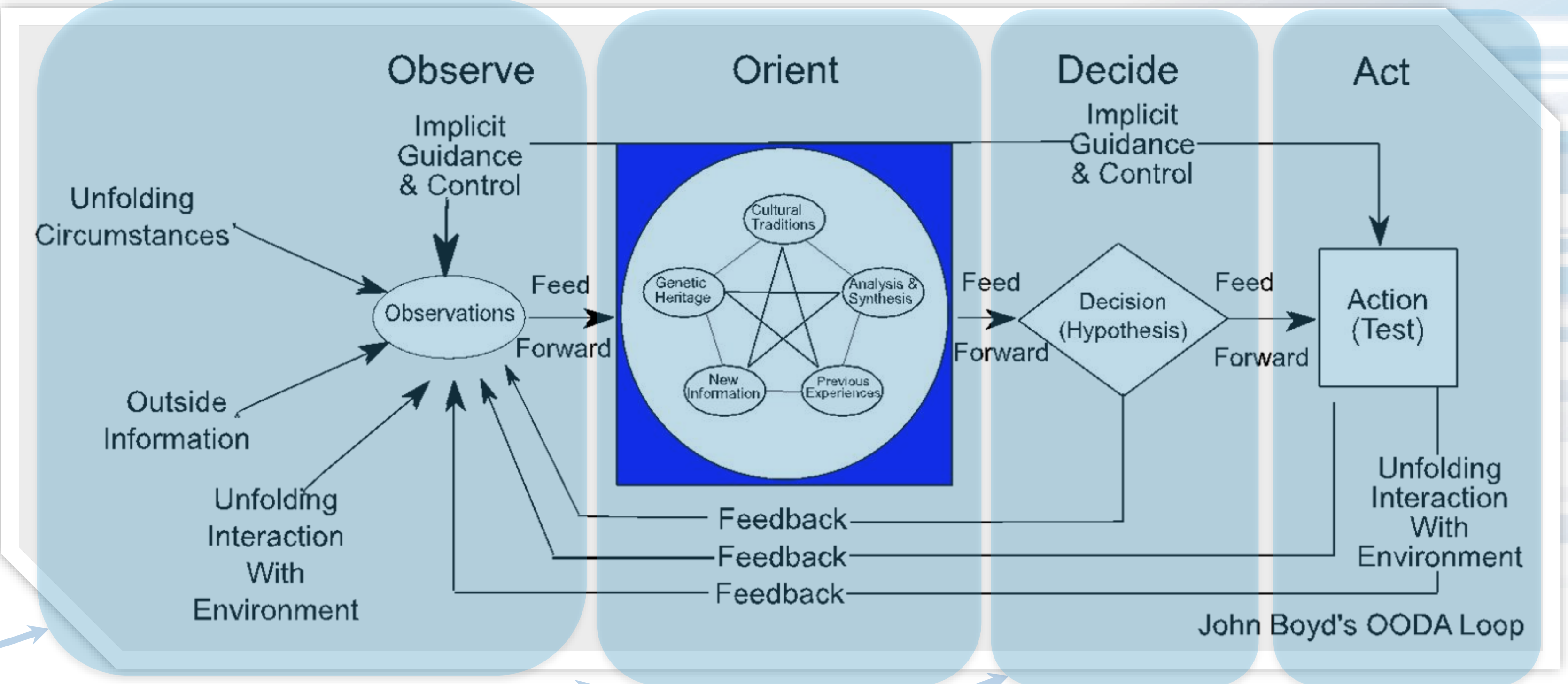


Putting it all together for a happy and healthy HPC I/O experience:

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 high level libraries for data management when possible.
4. Learn about performance debugging tools and techniques that you can reuse across systems.
5. Be aware that I/O performance naturally fluctuates over time.

One more thing: Improving I/O performance is an ongoing process

Figure by Patrick Edwin Moran



Applications are updated, systems change, and new allocations are granted.

We want to "teach a man to fish" by equipping you with the tools you need to monitor and improve your I/O performance.

Performance characterization tools (e.g., Darshan)

Background knowledge about how storage systems work (e.g., this presentation)

Facility resources (e.g., ALCF, OLCF, and NERSC staff and documentation)

Optimization techniques, tools, and libraries (e.g., later presentations today)

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Thank you!