Data Intensive Computing and I/O

ATPESC 2019

Rob Latham, Phil Carns, Quincey Koziol, and Glenn Lockwood

Q Center, St. Charles, IL (USA)
July 28 – August 9, 2019

Hands on exercises: https://xgitlab.cels.anl.gov/ATPESC-IO/hands-on

Welcome

Thank you for joining us for Track 3 of ATPESC 2019!

Data Intensive Computing and I/O:

How HPC storage systems work

Tools that simplify data management

How to access data more efficiently

Agenda (roughly)

- Morning:
  - Introductory concepts and tools
  - MPI-IO and PnetCDF
- Afternoon
  - HDF5
  - Architectures and tuning
- Evening
  - Hands-on exercises

Building up more detail as the day goes on

ATPESC attendees have access to a dedicated reservation on Theta throughout the day today. See the link at the top of each slide for details.

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.

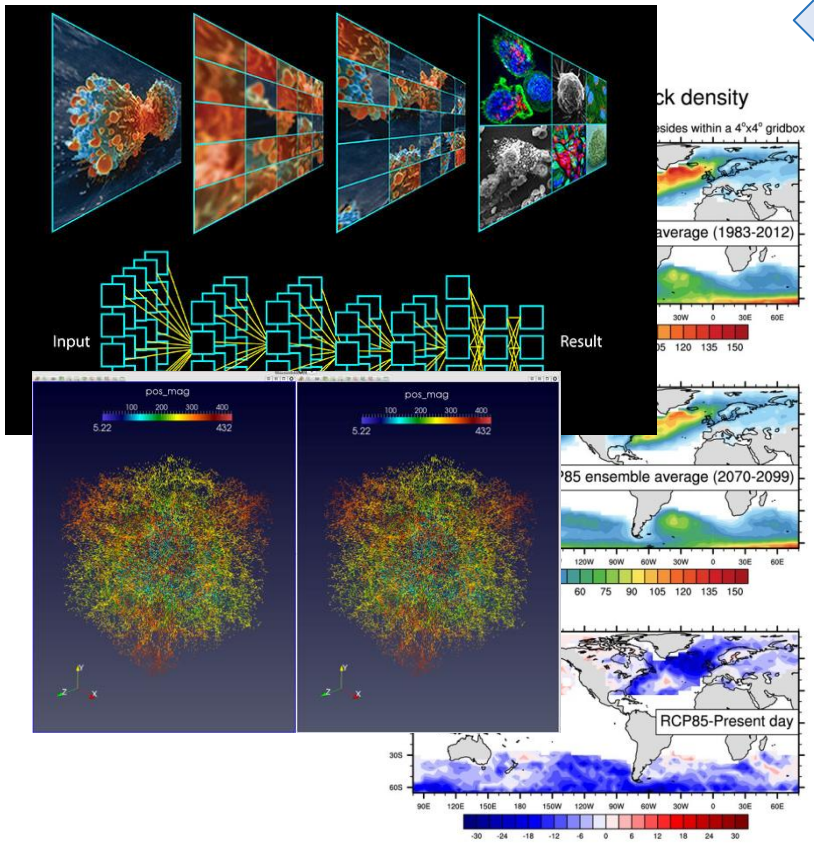
Quincey Koziol is a principal data architect at LBNL where he drives scientific data architecture discussions and participates in NERSC system design activities. He was the principal architect for the HDF5 project and a founding member of the HDF Group.

Glenn K. Lockwood is a storage architect at NERSC who specializes in I/O performance analysis, extreme-scale storage architectures, and emerging I/O technologies. He is an active maintainer of TOKIO, IOR, mdtest, and Darshan.

Phil Carns is a principal software development specialist at ANL who works on measurement, modeling, and development of data services. He has made key contributions to a variety of storage research projects, including Mochi, Darshan, CODES, and PVFS.

Bridging the gap between applications and storage systems

(your lecturers' day job)



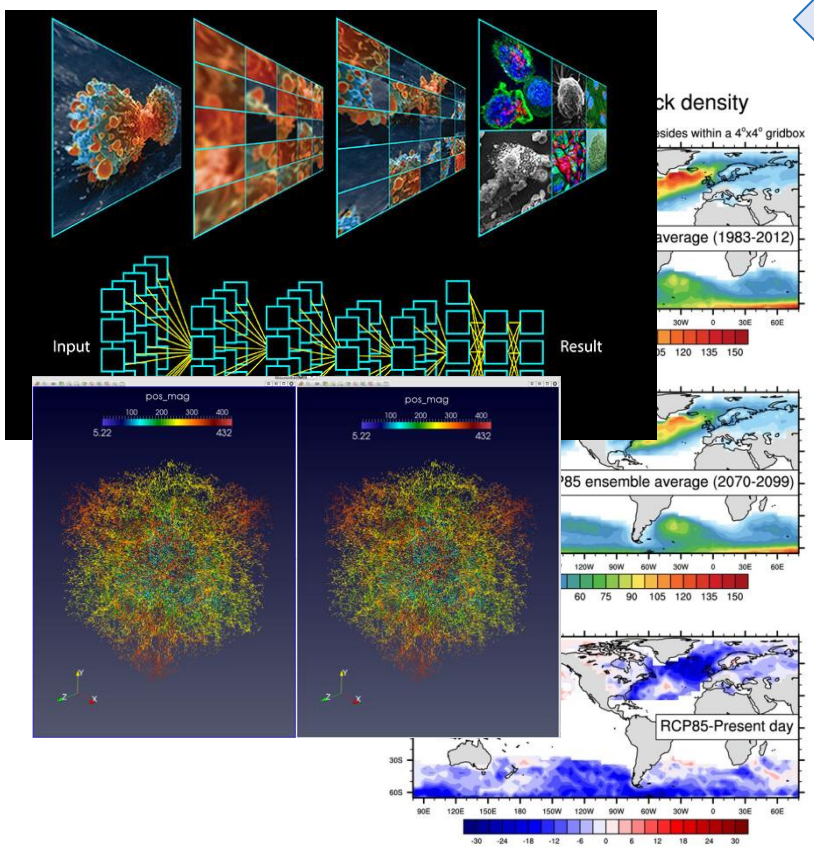
Techniques, algorithms, and software to bridge the "last mile" between scientific applications and storage systems.

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Bridging the gap between applications and storage systems

(your lecturers' day job)



This means:

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

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

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

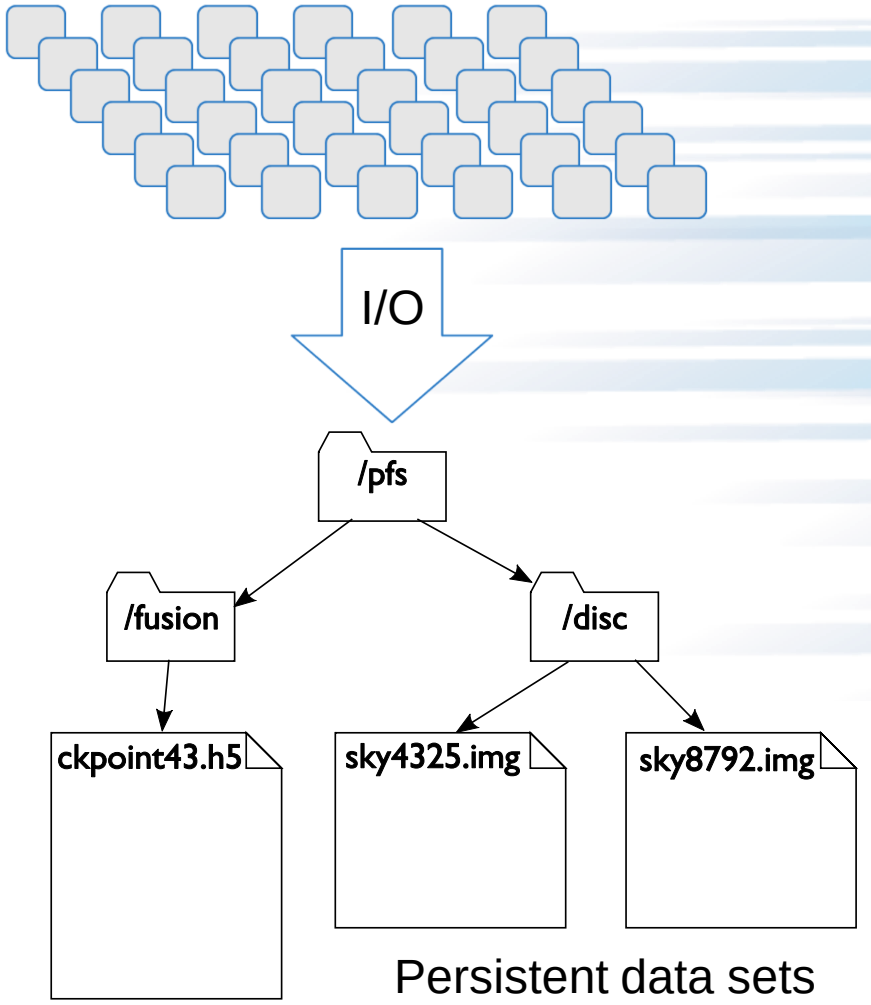
ATPESC 2019

Phil Carns
Mathematics and Computer Science Division
Argonne National Laboratory

Q Center, St. Charles, IL (USA)
July 28 – August 9, 2019

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.**
  - Parallel file systems can quickly store and access enormous volumes of data!
  - This requires coordination between applications, system software, and hardware components.
  - *Compute resources are wasted it takes too long to store and access 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
- But **a parallel file system does not behave like a conventional file system**
- We'll highlight 5 key, high-level differences in this presentation
- This is the background and motivation for more specific optimizations and usage tips that we will cover later in the day.

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 obvious which vehicle is best for each use case.

#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

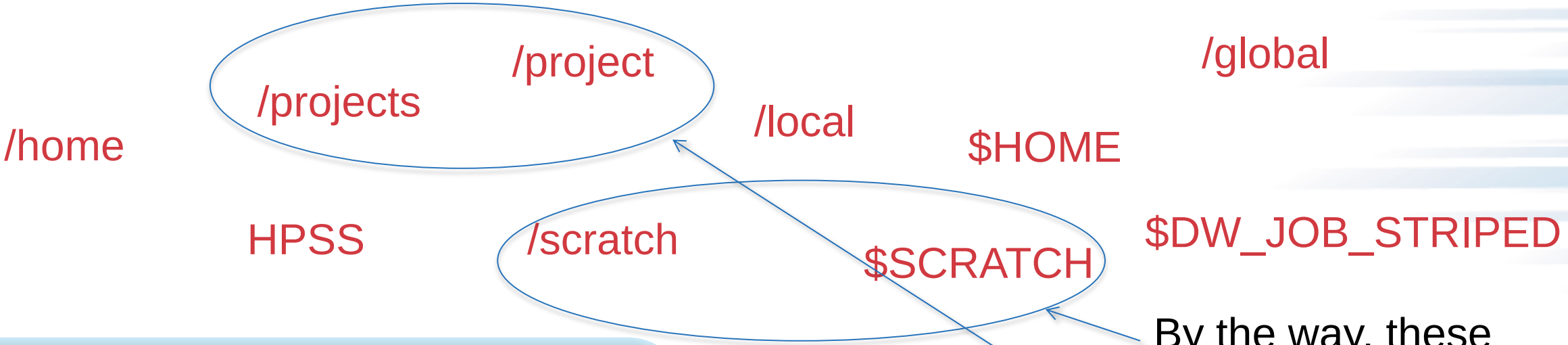
$DW_JOB_STRIPED

$SCRATCH

Now 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)



Now 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/

Hands on exercises: https://xgitlab.cels.anl.gov/ATPESC-IO/hands-on

How to *use* available file systems

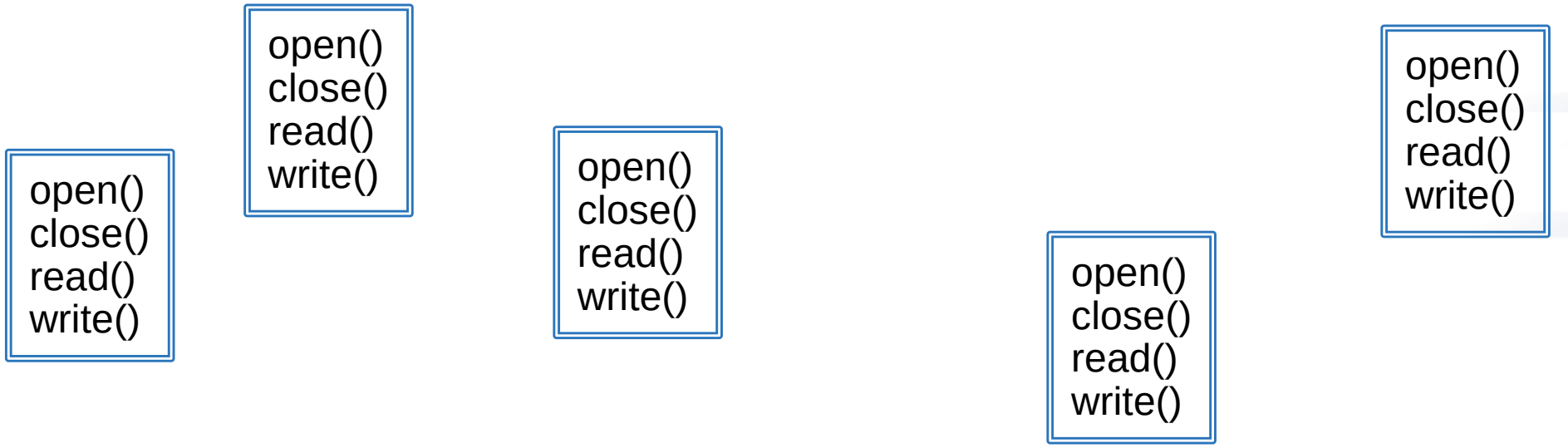
Can you tell what kind of vehicle you have by looking at it's interface?

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

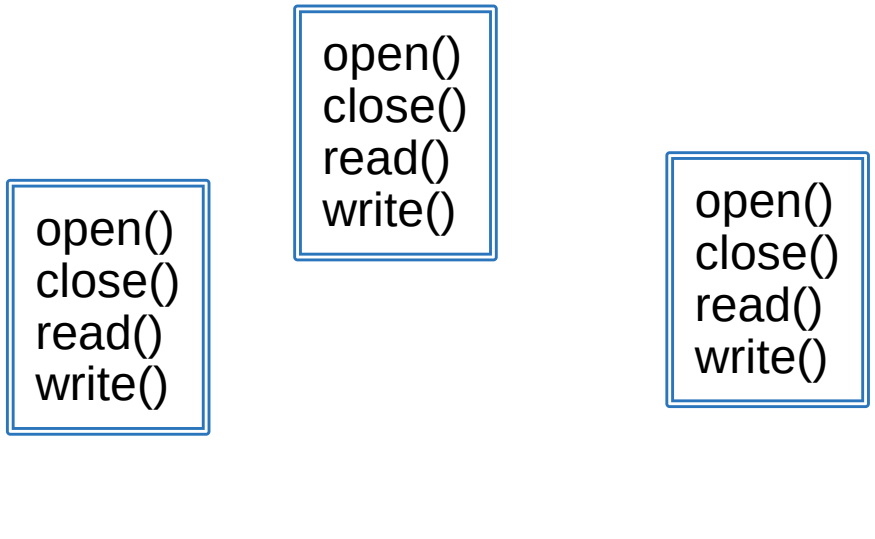
Hands on exercises: https://xgitlab.cels.anl.gov/ATPESC-IO/hands-on

How to *use* available file systems



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

How to *use* available file systems



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

Not so much. This is good for portability though!

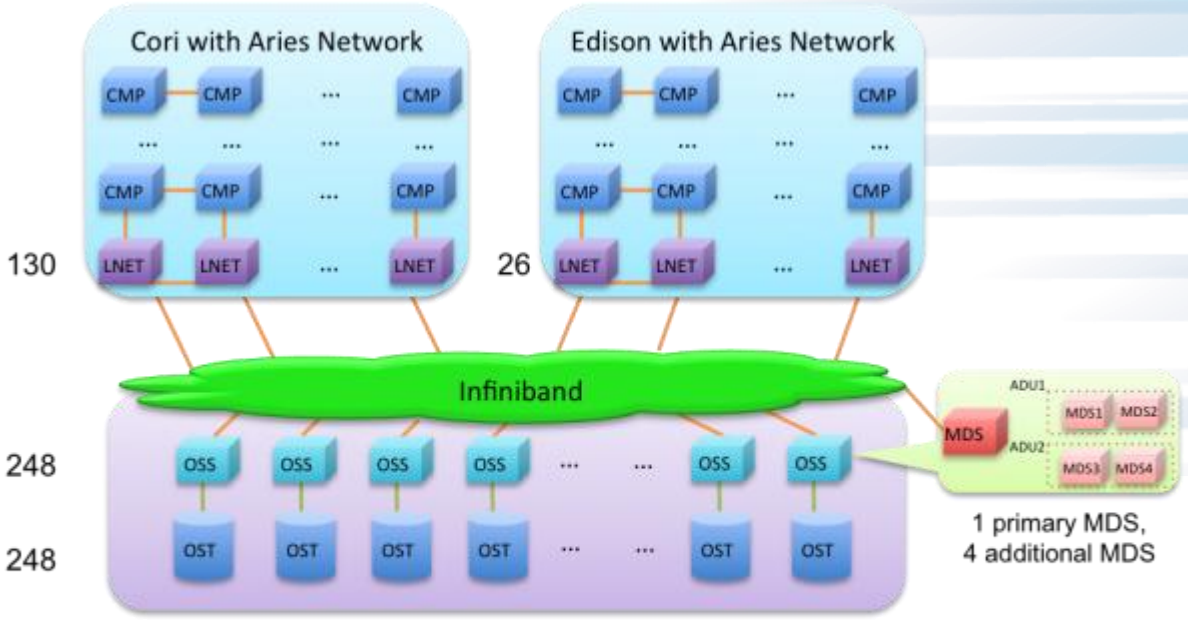
Be alert: applications will work correctly on many file systems, but the performance and capability differences are crucial.

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

- It looks like any other file system.
- But there are 10,000 or more disk drives!
- It takes a special hardware arrangement to handle thousands of disk drives.
- As a result, parallel file systems might not behave how you expect them to.

Cori scratch file system diagram
NERSC, 2017

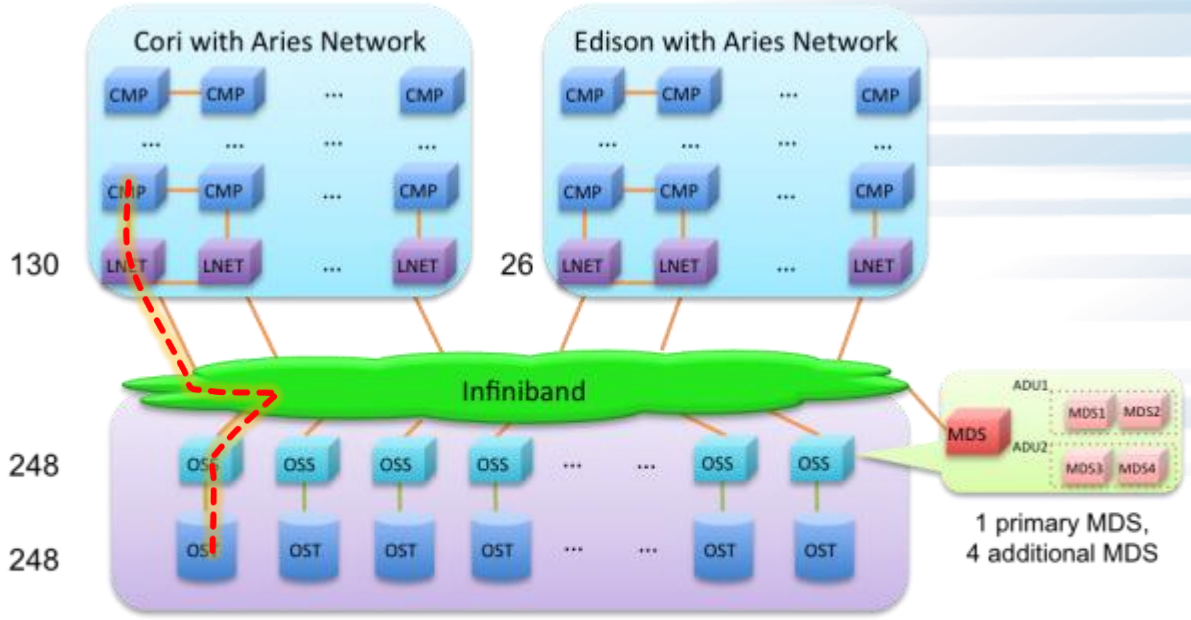


Each OSS controls one OST. The Infiniband connects the MDS, ADUs and OSSs to the LNET routers on the Cray XC System. The OSTs are configured with GridRAID, similar to RAID6, (8+2), but can restore failure 3.5 times faster than traditional RAID6. Each OST consists of 41 disks, and can deliver 240TB capacity.

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.
- This sounds like a bad thing (and frankly, it is), but what’s the silver lining?

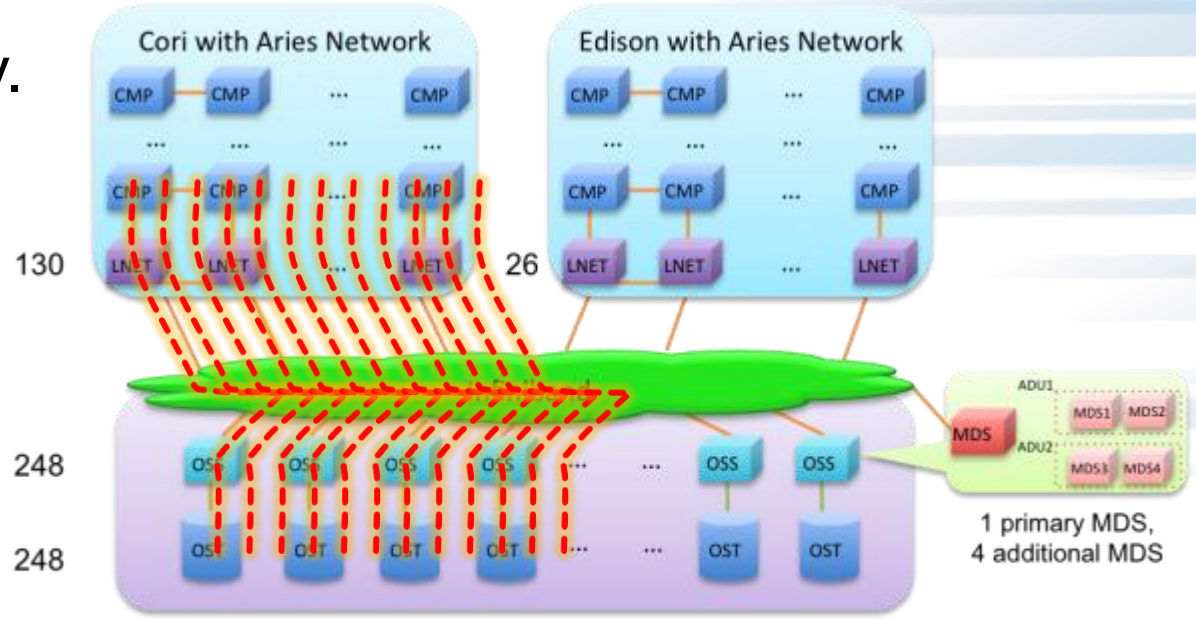
Cori scratch file system diagram
NERSC, 2017



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

- The network is very 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 for sequential small operations

Cori scratch file system diagram
NERSC, 2017



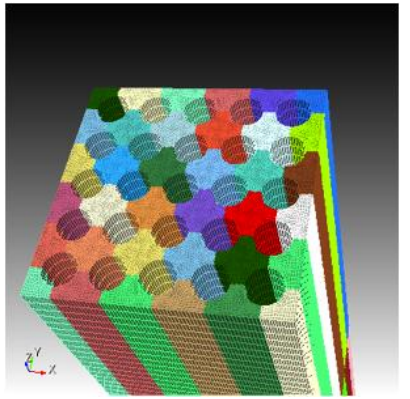
Each OSS controls one OST. The Infiniband connects the MDS, ADUs and OSSs to the LNET routers on the Cray XC System. The OSTs are configured with GridRAID, similar to RAID6, (8+2), but can restore failure 3.5 times faster than traditional RAID6. Each OST consists of 41 disks, and can deliver 240TB capacity.

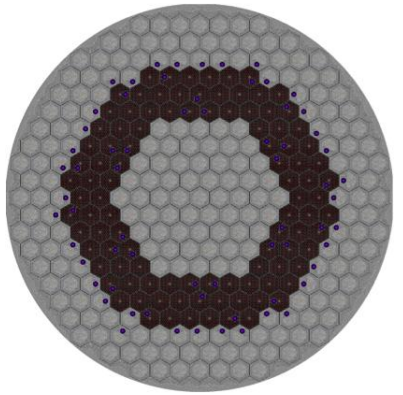
What is unique about HPC I/O?
#3 sophisticated application data models

Hands on exercises: https://xgitlab.cels.anl.gov/ATPESC-IO/hands-on

- 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)





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.

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

Argonne NATIONAL LABORATORY

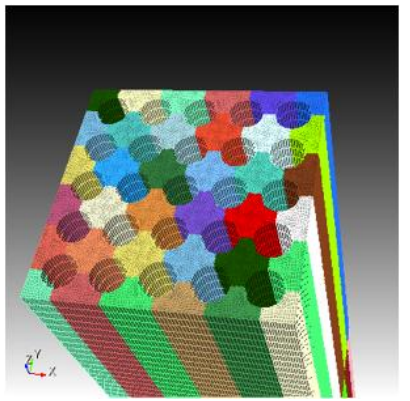
ECP EXASCALE COMPUTING PROJECT

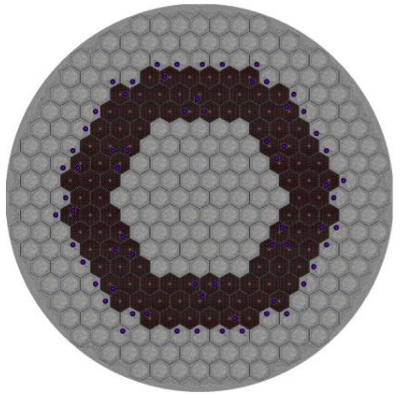
What is unique about HPC I/O?
#3 sophisticated application data models

Hands on exercises: https://xgitlab.cels.anl.gov/ATPESC-IO/hands-on

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 during the HDF5 and PNetCDF presentations.





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.

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

Hands on exercises: https://xgitlab.cels.anl.gov/ATPESC-IO/hands-on

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

IBM Spectrum Scale

panasas®

lustre®

… 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).

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Each HPC facility is different: I/O stack on Mira / ALCF

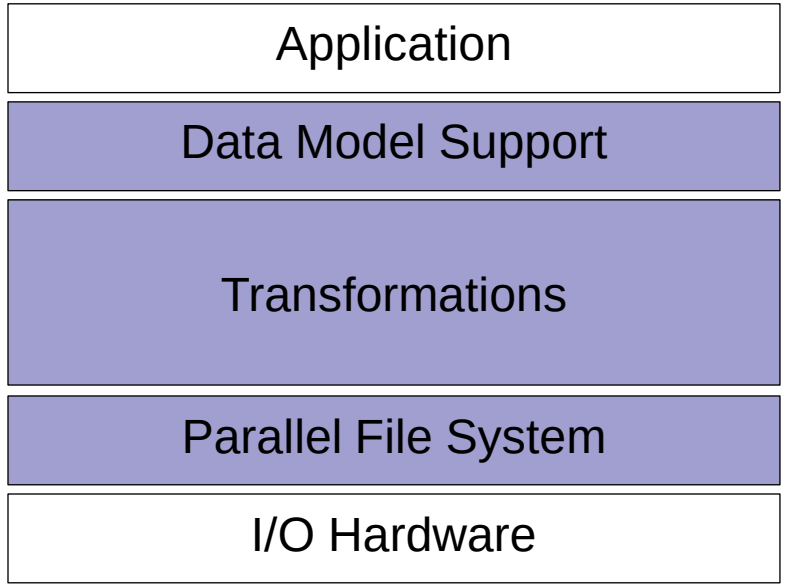
The "I/O stack" is the collection of software that translates application data access into storage system operations. It has a few layers.

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.

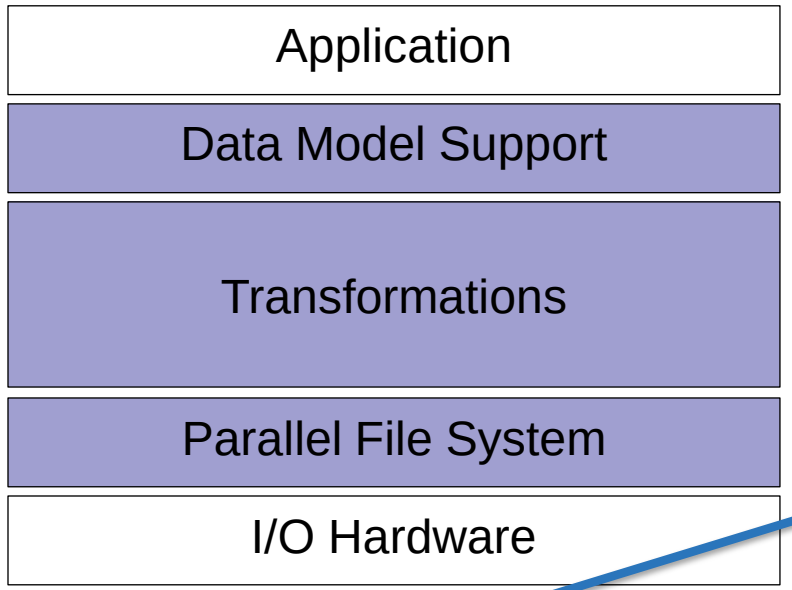
IBM ciod

Each HPC facility is different: I/O stack on Theta / ALCF

The application interface doesn't have to 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.

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

Lustre

Lnet routers

… even though some system details are very different!

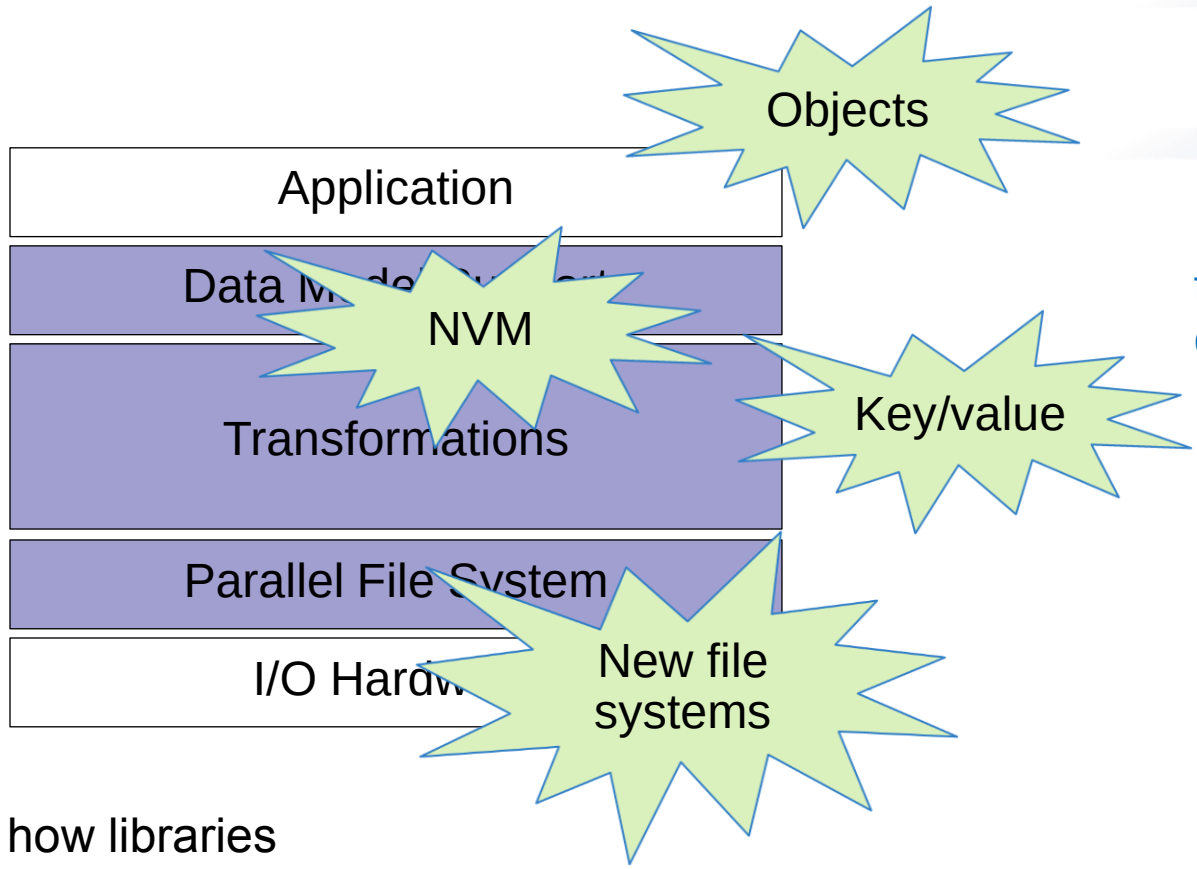
Each HPC facility is different: I/O stack on future machines

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.



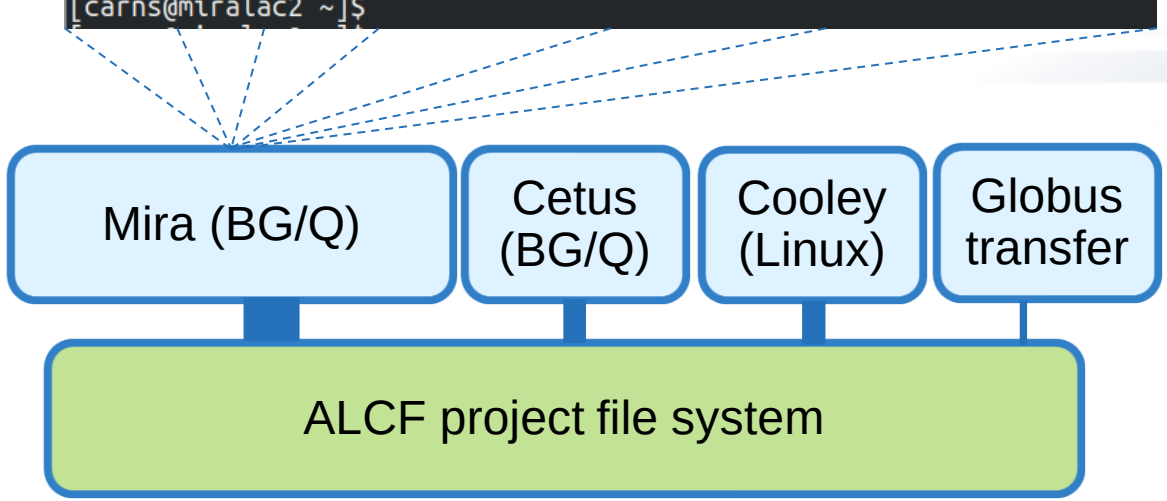
… even as technology changes.

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

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

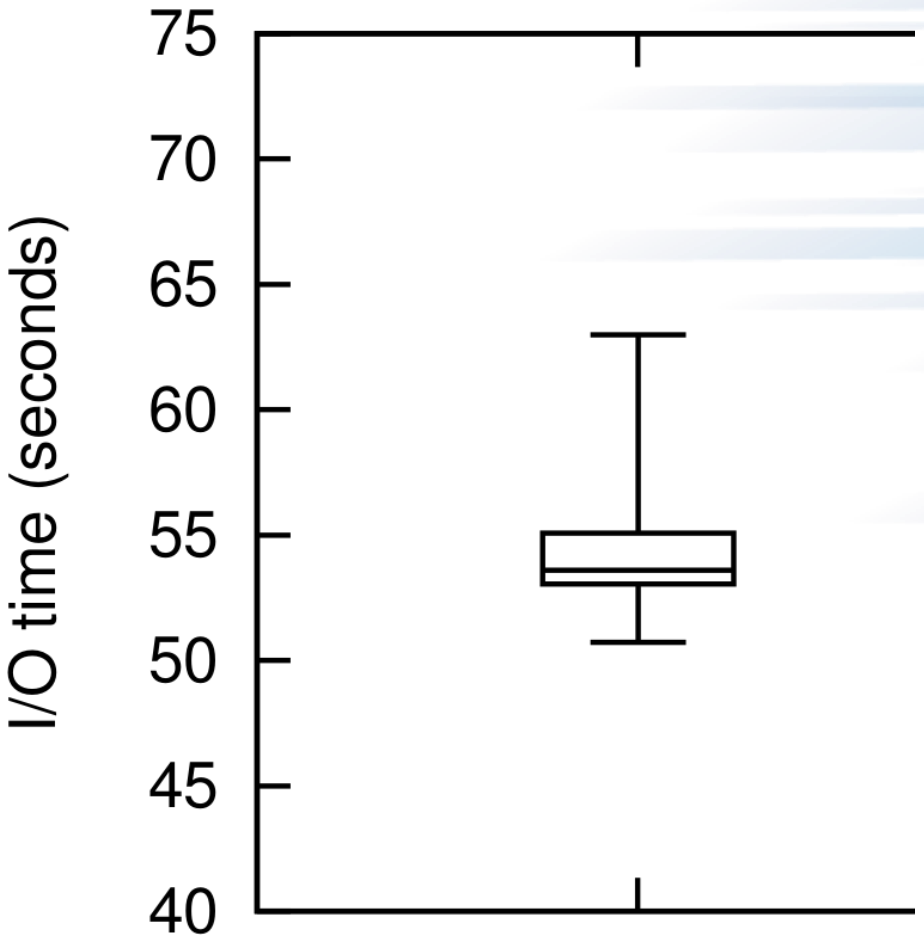
- Why:
  - Thousands of hard drives will *never* perform perfectly at the same time.
  - You are sharing storage with many other users.
  - You are also sharing storage with remote transfers, tape archives, and other data management tasks.
  - You are also sharing storage across multiple HPC systems.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

```
[carns@miralac2 ~]$ qstat |grep running
1139867          24:00:00  8192   running     MIR-48000-7BFF1-8192
1139871          24:00:00  8192   running     MIR-00000-33FF1-8192
1143326          12:00:00  2048   running     MIR-40C00-73FF1-2048
1151809          12:00:00  4096   running     MIR-40000-737F1-4096
1153083          24:00:00  16384  running     MIR-04000-77FF1-16384
1178836          12:00:00  512    running     MIR-408C0-73BF1-512
1178840          12:00:00  512    running     MIR-40880-73BB1-512
1179437          12:00:00  512    running     MIR-40840-73B71-512
1179755          02:00:00  4096   running     MIR-08000-3B7F1-4096
1179810          05:45:00  2048   running     MIR-08C00-3BFF1-2048
[carns@miralac2 ~]$
```



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

- When measuring I/O performance, take multiple samples.
- This figure shows 15 samples of I/O time from a 6,000 process benchmark on Edison system.
- Think about how you would accurately assess if a new data strategy performed better or worse. 2 measurements probably aren't enough.
- 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 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 about 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.

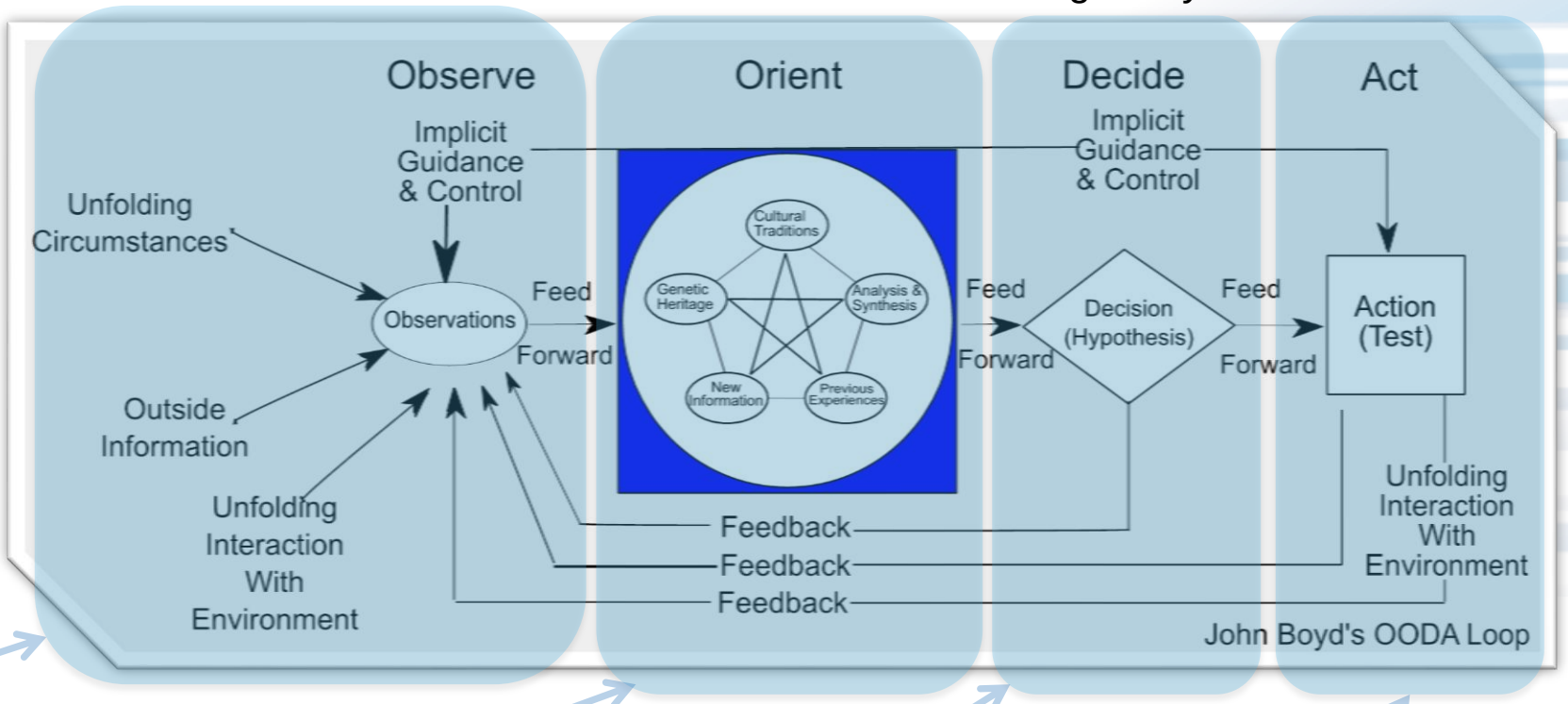
Hands on exercises: https://xgitlab.cels.anl.gov/ATPESC-IO/hands-on

Last but not least: Improving I/O performance is an ongoing process

Figure by Patrick Edwin Moran

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

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



Performance characterization tools, like Darshan

Background knowledge about how storage systems work

Help from facility resources

Optimization techniques, tools, and libraries.

Thank you!