REX-IO 2023

HPC STORAGE: ADAPTING TO CHANGE

PHIL CARNS
Mathematics and Computer Science Division
Argonne National Laboratory

October 31, 2023
Santa Fe, New Mexico

EXASCALE SYSTEMS ARE HERE

What does this mean for HPC storage?

Aurora system @ the Argonne Leadership Computing Facility

Nearly all problem domains are data-intensive at this point, and unprecedented compute capabilities call for unprecedented storage capabilities.

It's not enough to just do the same things we've always done, but faster. Some issues to consider:

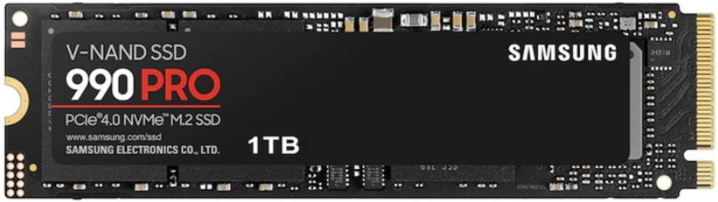
- Can existing storage system architectures take advantage of the potential of **new device technology**?
- What kinds of **novel data use cases** do we need to accommodate?
- How do we **embrace new users** from a broader collection of problem domains?

Let's look at some examples!

THE CHALLENGE OF NEW DEVICE TECHNOLOGY

DEVICE CAPABILITIES

Checking in on consumer NVMe specifications



990 PRO
$84.99 ~~$119.99~~
Save $35.00

990 PRO with Heatsink
$89.99 ~~$129.99~~
Save $40.00

Performance

Sequential Read Up to 7,450 MB/s * Performance may vary based on system hardware & configuration	**Sequential write** Up to 6,900 MB/s * Performance may vary based on system hardware & configuration	**Random Read (4KB ,QD32)** Up to 1,200,000 IOPS * Performance may vary based on system hardware & configuration
Random Write (4KB ,QD32) Up to 1,550,000 IOPS * Performance may vary based on system hardware & configuration	**Random Read (4KB ,QD1)** Up to 22,000 IOPS * Performance may vary based on system hardware & configuration	**Random Write (4KB ,QD1)** Up to 80,000 IOPS * Performance may vary based on system hardware & configuration

Screenshots from Samsung web site, retrieved October 2023

- October 2023: $85 buys an off-the-shelf storage device with:
  > 1 million IOPs and
  > 7 GB/s throughput.
  - No heat sink, though; that’s another $5.
- Devices with embedded compute features are not nearly as cheap or widely available, but they are coming.

WHY DEVICE CAPABILITIES MATTER

HPC storage is more than just checkpointing

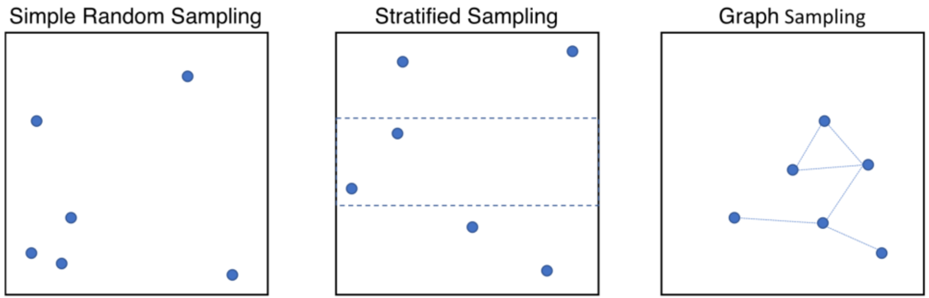


Figure credit: B. Settlemyer, G. Amvrosiadis, P. Carns and R. Ross, "It's Time to Talk About HPC Storage: Perspectives on the Past and Future," in *Computing in Science & Engineering*, vol. 23, no. 6, pp. 63-68, 1 Nov.-Dec. 2021

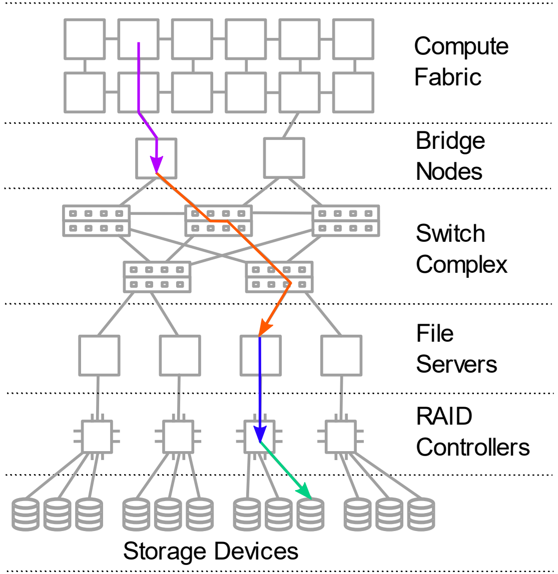
- For example: many data-intensive algorithms rely on statistical or AI methods that extract samples from immense data sets.
- The resulting storage access patterns *are often unpredictable to outside observers.*
- That's bad for general purpose caching and prefetching, but modern device characteristics should be well-equipped to deal with the workload directly.

MAKING USE OF DEVICE CAPABILITIES

A traditional HPC storage architecture

Figure credit: B. Settlemyer, G. Amvrosiadis, P. Carns and R. Ross, "It's Time to Talk About HPC Storage: Perspectives on the Past and Future," in *Computing in Science & Engineering*, vol. 23, no. 6, pp. 63-68, 1 Nov.-Dec. 2021

- The traditional HPC storage architecture is designed to maximize aggregate bandwidth in a disaggregated system.
- The resulting architectural model isn't great for response time, though.
  - There are many "hops", each with its own serialization, protocol, and buffering.
- How can you leverage the strengths of new storage devices in this environment?
  - This is a known problem, and a variety of potential solutions have been deployed.



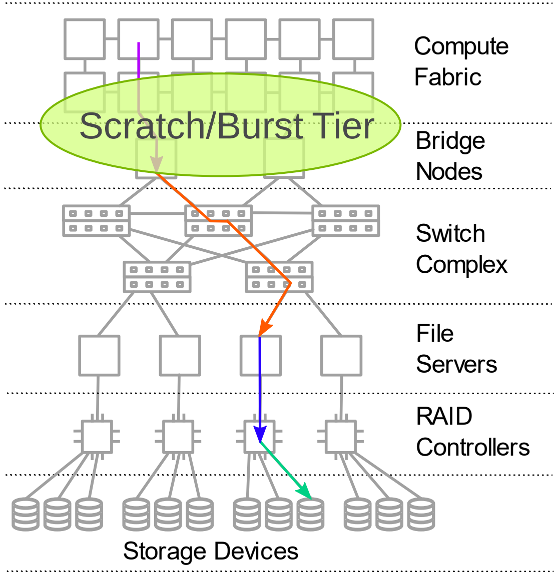
Systems designed to maximize aggregate throughput are poorly suited to servicing individual random reads.

MAKING USE OF DEVICE CAPABILITIES

Burst buffers and local devices

Figure credit: B. Settlemyer, G. Amvrosiadis, P. Carns and R. Ross, "It's Time to Talk About HPC Storage: Perspectives on the Past and Future," in *Computing in Science & Engineering*, vol. 23, no. 6, pp. 63-68, 1 Nov.-Dec. 2021

- One obvious solution is to (also) provide locally attached scratch devices or a burst buffer tier.
- How does this affect the user experience?
  - How do they stage data?
  - Can they still use shared data structures, or only local data structures?
  - Is the application still in charge of the mapping of data models to local POSIX files?
  - Where are the smart devices, and how do you use them while retaining portability?
  - **How portable is the overall data workflow across systems with different devices?**



Systems designed to maximize aggregate throughput are poorly suited to servicing individual random reads.

STORAGE ARCHITECTURES FOR MODERN DEVICES

- Conventional storage resources are often thought of as distinct silos with different properties (latency, bandwidth, sharing, etc.), but this can be problematic.
- Potential pitfalls:
  – Lack of integration: fragmented name spaces and policies
  – Over-generalization: least-common-denominator APIs cannot take advantage of device properties
  – Conservative hardware assumptions: scheduling, replication, and placement policies can be too pessimistic for modern devices
- Can we go beyond this model to deploy coherent on-demand services, with flexible APIs, adapted to available hardware resources… and still be portable?

IS IT PLAUSIBLE?

- Cloud services have been very successful offering a range of data service types and storage device properties.
- They are socializing the concept of choosing the solution for the task at hand.

Database type	AWS service
Relational	Amazon Aurora, Amazon RDS, Amazon Redshift
Key-value	Amazon DynamoDB
In-memory	Amazon ElastiCache, Amazon MemoryDB for Redis
Document	Amazon DocumentDB (with MongoDB compatibility)
Wide column	Amazon Keyspaces
Graph	Amazon Neptune
Time series	Amazon Timestream
Ledger	Amazon Ledger Database Services (QLDB)

Object, file, and block storage

Amazon Simple Storage Service (S3)

Object storage with industry-leading scalability, availability, and security for you to store and retrieve any amount of data from anywhere.

Amazon Elastic File System (EFS)

A simple, serverless, elastic, set-and-forget file system for you to share file data without managing storage.

Amazon Elastic Block Store (EBS)

Easy to use, high-performance block storage service for both throughput and transaction-intensive workloads at any scale.

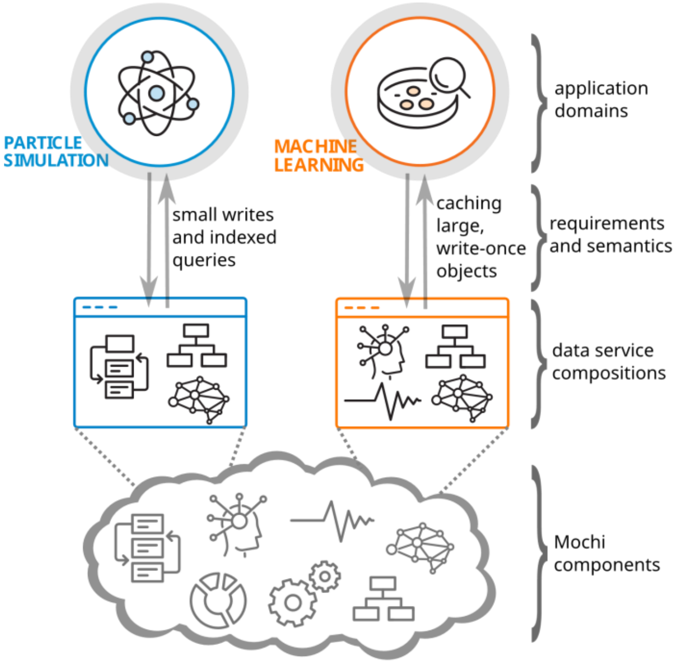
FSx — Amazon FSx

Fully managed, cost-effective file storage offering the capabilities and performance of popular commercial and open-source file systems.

Screenshots from https://aws.amazon.com/products, April 2022

ONE APPROACH: THE MOCHI MODEL

Composable data services for varying use cases



The Mochi project provides a library of robust, reusable, modular, and connectable data management components and microservices along with a methodology for composing them into specialized distributed data services.

- This was originally envisioned to enable specialization for different application needs.
- Recent work is also pushing for greater architectural adaptability as well:
  - Using smart devices when available
  - Service elasticity to effectively use resources

Figure credit: P. Carns, M. Dorier, R. Latham, R. Ross, S. Snyder, J. Soumagne, "A Case Study in Translational Computer Science for HPC Data Storage," *in preparation*.

THE CHALLENGE OF NOVEL DATA USE CASES

DIVERSIFYING STORAGE USE CASES

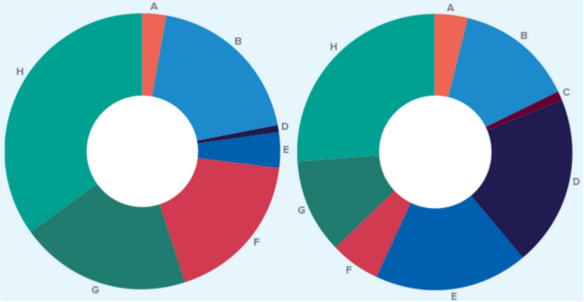
HPC is now accessible to more problem domains than ever before

Table and figure credit: 2022 ALCF Science report

2022 INCITE NODE HOURS	2022 ALCC NODE HOURS
17.8M	6.8M

2022 by Domain

Domain	INCITE	ALCC
A. Biological Sciences	3 %	4 %
B. Chemistry	9	14
C. Computer Science	—	1
D. Earth Science	1	20
E. Energy Technologies	4	18
F. Engineering	18	6
G. Materials Science	20	11
H. Physics	35	26



- ALCF example: there are over a thousand users and hundreds of projects.
- DOE Allocation programs are highly competitive and span diverse fields.
- This is one of the greatest triumphs of computer science in HPC: making cutting-edge compute resources accessible to all researchers.
- However, we (computer scientists) do not really know how all of these researchers are using storage, much less **how they would like to use the storage.**

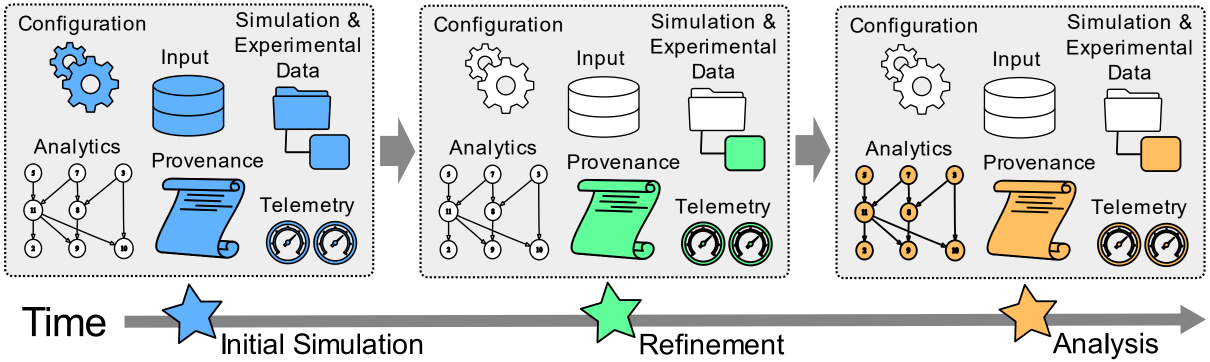
THE EVOLUTION OF COMPUTATIONAL SCIENCE

- You can get a feel for application trends by looking at user events at the ALCF, for example.
- It's not just Fortran linear algebra anymore (and hasn't been for a long time now); events focus on accelerators, machine learning, data, neuromorphic algorithms, neural networks, etc.
- Scientists are employing a variety of data models and programming models to reach their objectives.

Screenshot from https://www.alcf.anl.gov/, April 2022

EVERYTHING, EVERYWHERE, ALL AT ONCE

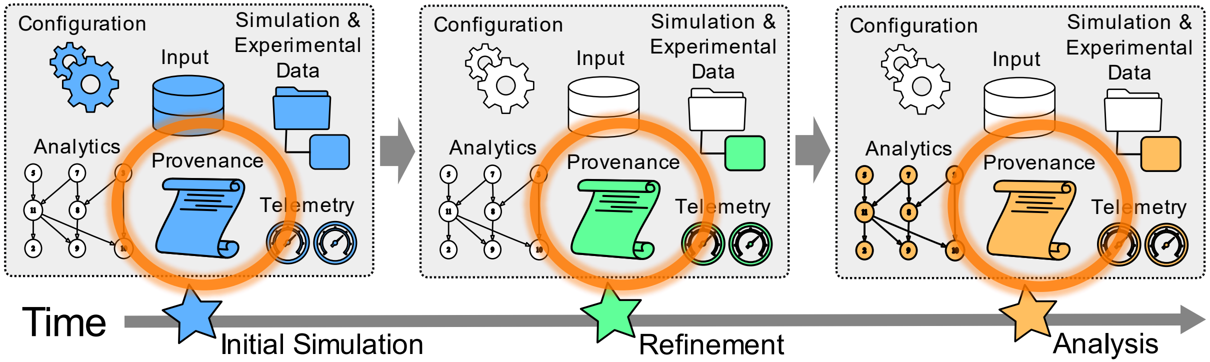
Using many computation methods in one workflow



- Not only does the modern scientific computing portfolio include observational data management, simulation, machine learning, analytics, and more…
- … but it increasingly combines several of those elements into a single workflow!
- A single workflow may similarly employ a wide range of data management methods.

GOING META

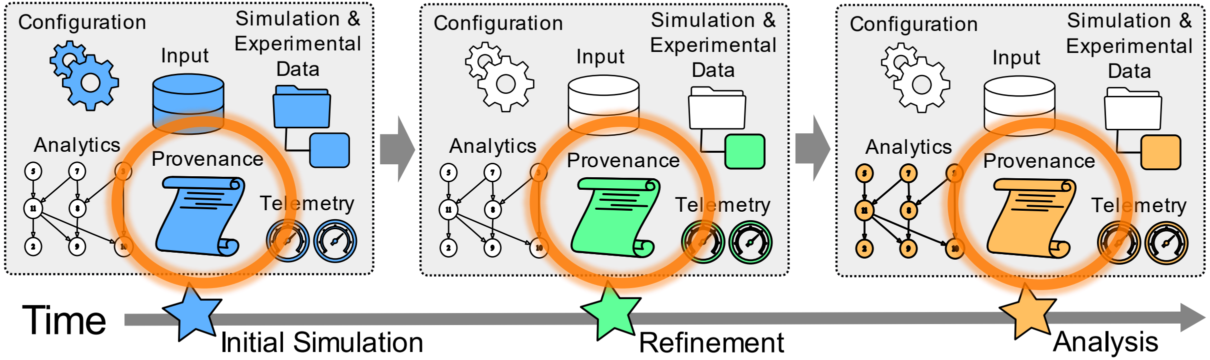
Using many computational methods in one workflow



- The workflow paradigm also introduces a new meta use case: storage and use of provenance data *about* the workflow.
- Why this is important:
  - How do you reproduce your results?
  - What do you do if results are not consistent?
  - What do you do if performance is not consistent?

GOING META

Using many computational methods in one workflow



- The workflow paradigm also introduces a new meta use case: storage and use of provenance data *about* the workflow.
- Why this is important:
  - How do you reproduce your results?
  - What do you do if results are not consistent?
  - What do you do if performance is not consistent?

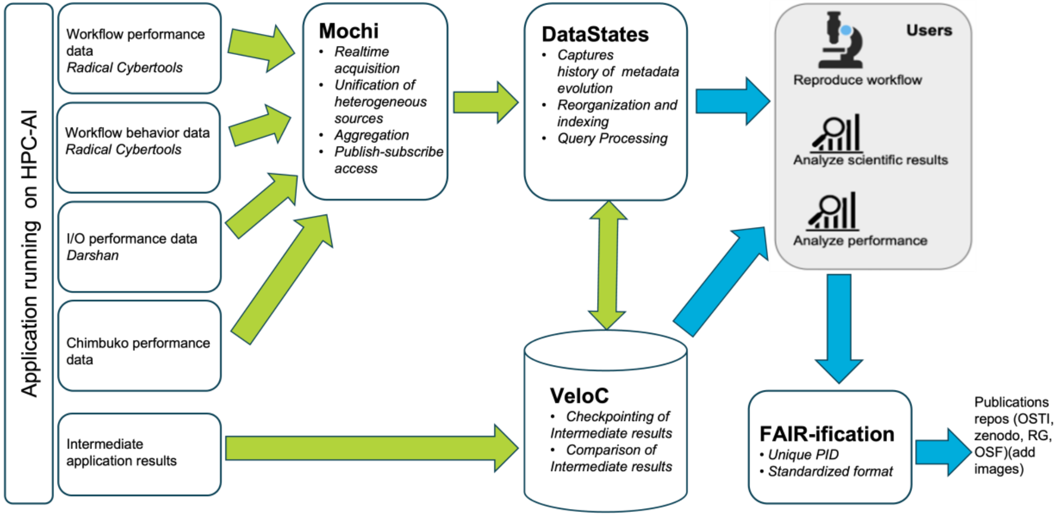
Findable Accessible Interoperable Reusable

FAIR logo: CC BY-SA 4.0 Sangya Pundir

Line Pouchard (BNL), PI
Bogdan Nicolae (ANL), Co-PI
Tanzima Islam (TXST), Co-PI

ONE APPROACH: RECUP

Scalable Metadata and Provenance Services for Reproducible Hybrid Workflows



Nicolae, Bogdan et al. "Building the I (Interoperability) of FAIR for Performance Reproducibility of Large-Scale Composable Workflows in RECUP." *2023 IEEE 19th International Conference on e-Science (e-Science)* (2023): 1-7.

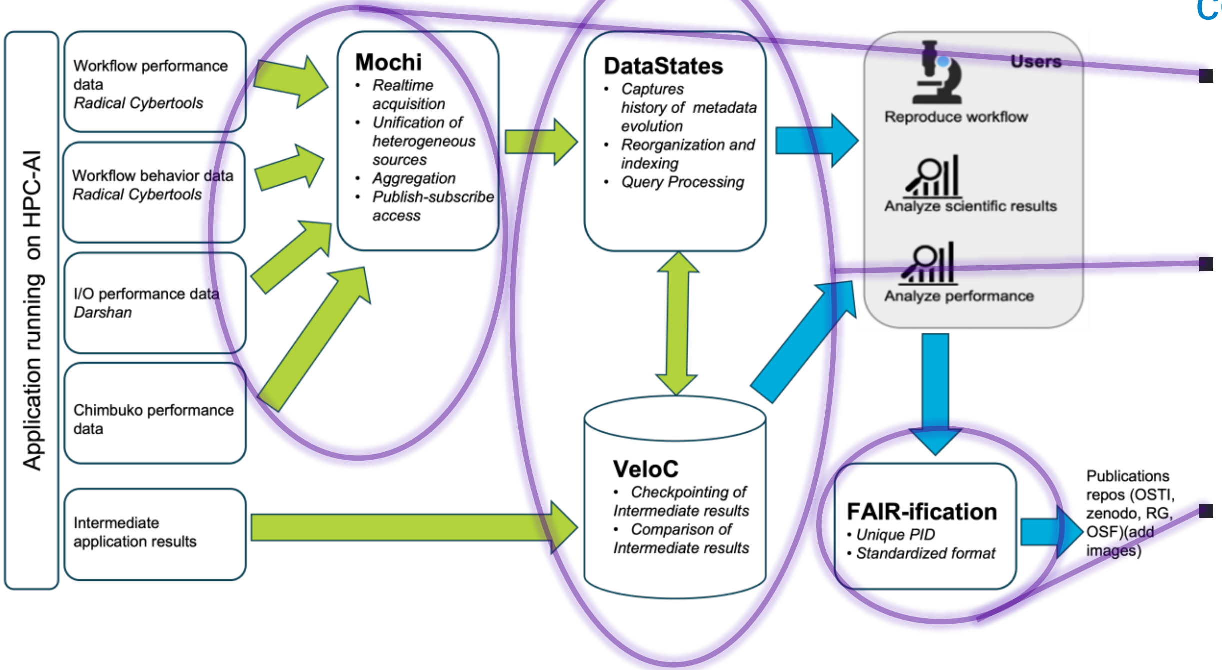
Hybrid workflows: addressing workflows that include data-intensive tasks and numerical calculations

Performance reproducibility: minimal run-to-run variation using a consistent set of configurations

Result reproducibility: the statistical reproducibility of results within certain error bounds

ONE APPROACH: RECUP

Scalable Metadata and Provenance Services for Reproducible Hybrid Workflows



This means addressing new data use case challenges that aren't well-served by conventional file systems:

- Aggregating diverse, telemetry, at scale, in a coherent manner
- Absorbing not just raw data, but also its lineage and evolution for comparative analysis
- Making data Findable, Accessible, Interoperable, and Reusable

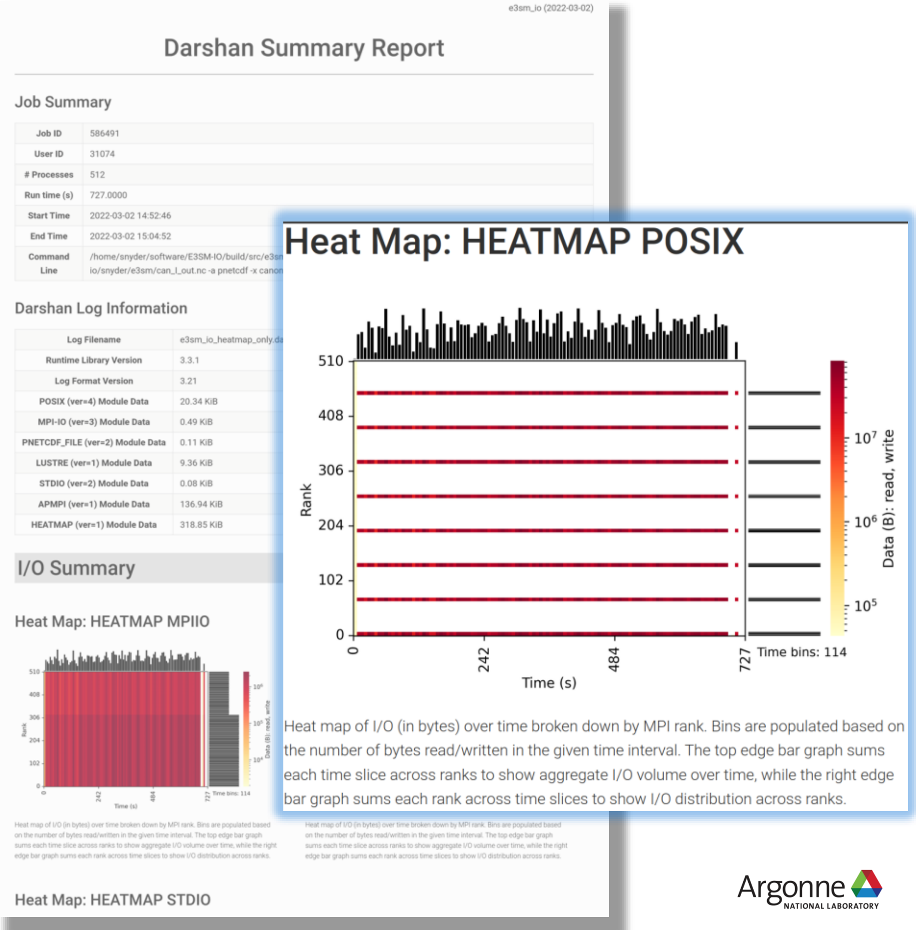
IMPLICATIONS OF NOVEL DATA USE CASES

- FAIR / metadata / provenance / reproducibility is just one example of new use cases pushing the envelope of today's HPC storage.
- There are many other use cases emerging.
- We need to do more work to take the needs of the science community into account.
- Don't forget that computer science is real science too!
- Faster and larger capacity storage is great, but this isn't just business as usual.

THE CHALLENGE OF EMBRACING NEW USERS

UNDERSTANDING IO PERFORMANCE

- *“8 GiB/s sounds good.”*
- *“No, wait; I just Googled the facility documentation. I should be getting many hundreds of GiB/s!”*
- “Stunt mode” platform benchmarks are misleading (at best) for real users.
- Tools like Darshan can provide more meaningful insight into application I/O behavior.
- The goal: provide easy-to-interpret metrics to users that are relevant to their objectives and scale.



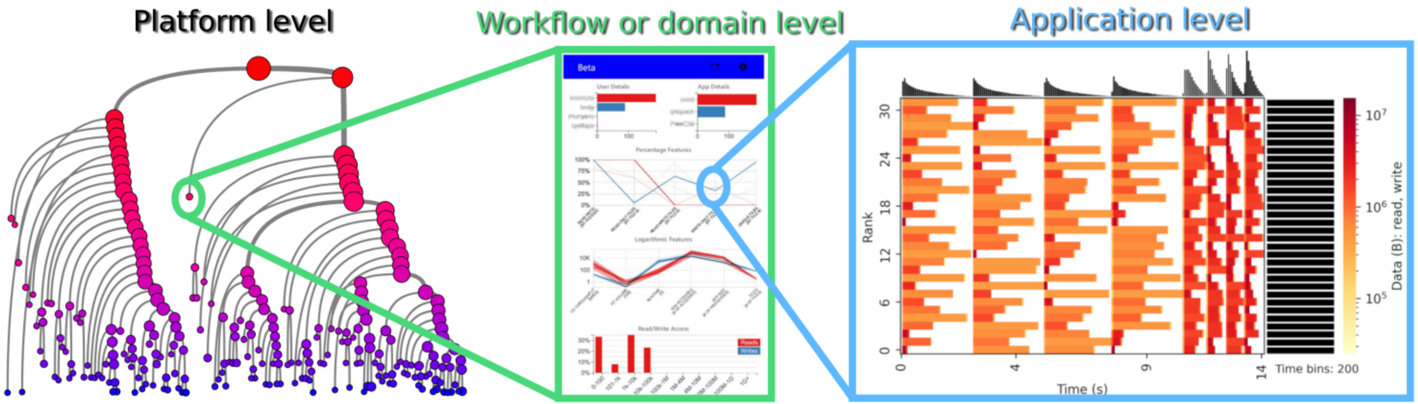
THE NEXT FRONTIER: INTERPRETATION

- Context is a crucial to interpretation:
  - **Spatial:** How does this application relate to similar or concurrent applications?
  - **Temporal:** How has this workload performed in the past?
  - **Science objectives:** Did good storage throughput actually contribute to productivity?

- Context is crucial… if the data can be interpreted at all. Scientists are passionate about their chosen field, not parallel file system arcana. We need to bridge this gap.

- It's also important to recognize that the user community keeps scaling, but the I/O expert community does not.

IO PERFORMANCE IN CONTEXT: SPATIAL



Hierarchical clustering of jobs on a system in terms of I/O workload characteristics

Salient features of applications that constitute a cluster

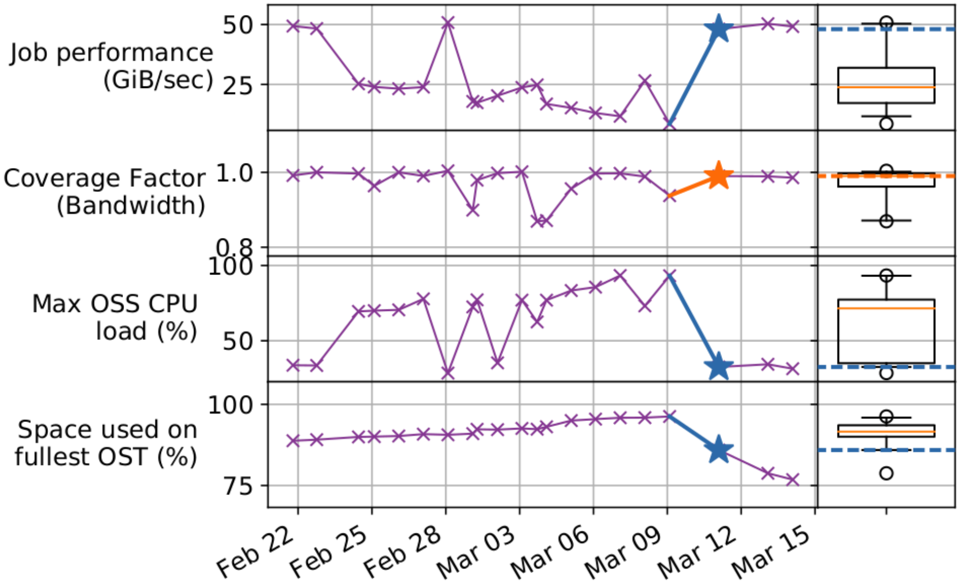
Workload intensity across processes (vertical) and time (horizontal) for a single application run

- Platform: what workloads are present, and can you group them to gain insight?
- Workflow or domain: how do similar jobs behave and why are some faster?
- Application: how is the I/O distributed within the job?

Image credits: Isakov 2020, Del Rosario 2020, Awtrey 2021

IO PERFORMANCE IN CONTEXT: TEMPORAL

- Previous executions (of your own application or similar applications) can also serve as a reference point.
- Understand if current performance is normal or anomalous.
- Gain insight into variability and correlated system factors that impact performance.
- Sometimes performance changes for reasons beyond the user's control.



Lockwood et al. "UMAMI: a recipe for generating meaningful metrics through holistic I/O performance analysis" in PDSW 2017

IO PERFORMANCE IN CONTEXT: SCIENCE OBJECTIVES

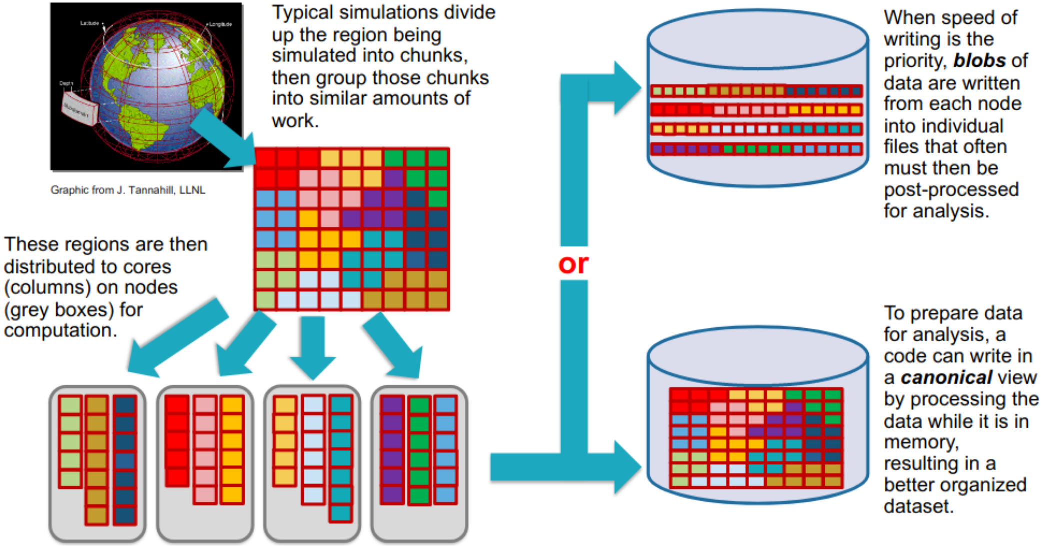
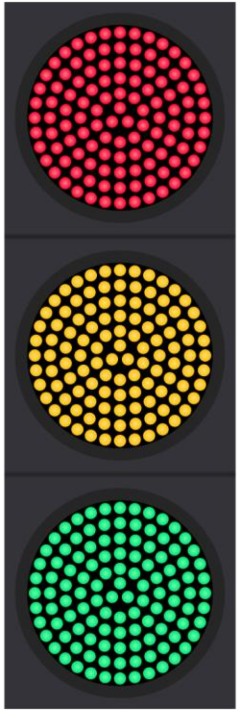


Image credit:
Rob Ross, ANL

- The I/O technique that yields the shortest simulation runtime doesn't necessarily yield the highest science productivity.
- Users must consider their data management and analysis needs as well.
- How does the data management strategy impact the overall workflow?

ENGAGING AND EMPOWERING STAKEHOLDERS



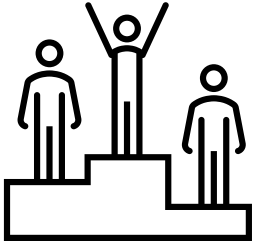
How do we turn this wealth of contextual information into something easily usable?

There are many publications about I/O tuning, but we as a community haven’t distilled it and transferred it well to stakeholders.

What if we could automatically identify concise, salient features that would give users the best “bang for the buck”?

- “The workload for this file would perform better on /mnt/foo”
- “This file is not striped well; set hint “abracadabra””
- “Writes are interleaved and unaligned; try a collective write.”

Can we go even further and quantify potential gains and costs to help users game the system?



ONE APPROACH: DRISHTI

J. L. Bez, H. Ather and S. Byna, "Drishti: Guiding End-Users in the I/O Optimization Journey," *2022 IEEE/ACM International Parallel Data Systems Workshop (PDSW)*, Dallas, TX, USA, 2022, pp. 1-6

```
Drishti

DRISHTIv.0.3

JOB:            1190243
EXECUTABLE:     bin/8_benchmark_parallel
DARSHAN:        jlbez_8_benchmark_parallel_id1190243_7-23-45631-11755726114084236527_1.darshan
EXECUTION DATE: 2021-07-23 16:40:31+00:00 to 2021-07-23 16:40:32+00:00 (0.00 hours)
FILES:          6 files (1 use STDIO, 2 use POSIX, 1 use MPI-IO)
PROCESSES       64
HINTS:          romio_no_indep_rw=true cb_nodes=4

1 critical issues, 5 warnings, and 5 recommendations

METADATA

▶ Application is read operation intensive (6.34% writes vs. 93.66% reads)
 ▶ Application might have redundant read traffic (more data was read than the highest read offset)
 ▶ Application might have redundant write traffic (more data was written than the highest write offset)

OPERATIONS

 ▶ Application issues a high number (285) of small read requests (i.e., < 1MB) which represents 37.11% of all
read/write requests
   ↪ 284 (36.98%) small read requests are to "benchmark.h5"
 ▶ Application mostly uses consecutive (2.73%) and sequential (90.62%) read requests
 ▶ Application mostly uses consecutive (19.23%) and sequential (76.92%) write requests
 ▶ Application uses MPI-IO and read data using 640 (83.55%) collective operations
 ▶ Application uses MPI-IO and write data using 768 (100.00%) collective operations
 ▶ Application could benefit from non-blocking (asynchronous) reads
 ▶ Application could benefit from non-blocking (asynchronous) writes
 ▶ Application is using inter-node aggregators (which require network communication)

2022 |  LBL|  Drishti report generated at 2022-08-05 13:19:59.787458 in 0.955 seconds
```

Drishti (by Jean Luca Bez, LBNL) is an example of taking the next step in interpretation:

Giving users actionable recommendations in addition to characterization and analysis.

This example shows a human-readable interpretation of I/O issues within an application.

ONE APPROACH: DRISHTI

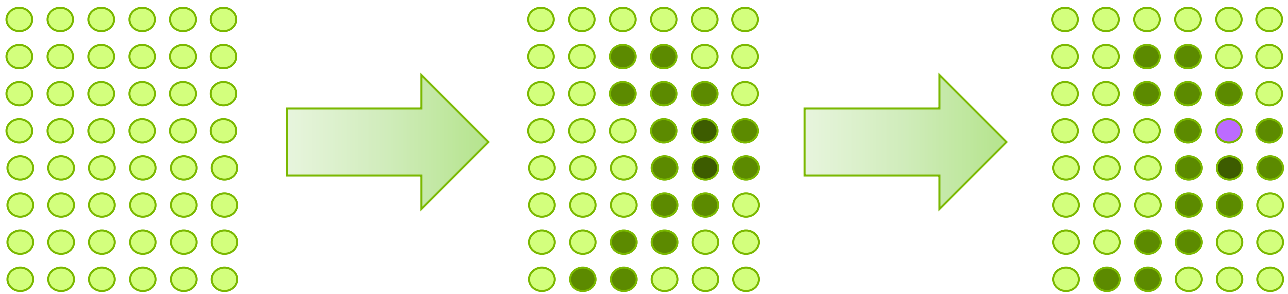
J. L. Bez, H. Ather and S. Byna, "Drishti: Guiding End-Users in the I/O Optimization Journey," *2022 IEEE/ACM International Parallel Data Systems Workshop (PDSW)*, Dallas, TX, USA, 2022, pp. 1-6

```
OPERATIONS

 ▶ Application issues a high number (285) of small read requests (i.e., < 1MB) which represents 37.11% of all
read/write requests
   ↪ 284 (36.98%) small read requests are to "benchmark.h5"
   ↪ Recommendations:
     ↪ Consider buffering read operations into larger more contiguous ones
     ↪ Since the appplication already uses MPI-IO, consider using collective I/O calls (e.g. MPI_File_read_all() or
MPI_File_read_at_all()) to aggregate requests into larger ones

        Solution Example Snippet

          MPI_File_open(MPI_COMM_WORLD, "output-example.txt", MPI_MODE_CREATE|MPI_MODE_RDONLY, MPI_INFO_NULL,
          ...
          MPI_File_read_all(fh, &buffer, size, MPI_INT, &s);
```

Drishti can also recommend remedies; in this case with a code snippet.

FUTURE OPPORTUNITIES IN AI FOR SERVICES

- What if you need to tune not just an application, but an entire on-demand storage architecture?
- Imagine this: “Act like an HPC I/O expert. The following is a description of my workload and composition. Write a good starting configuration.”
- LLMs might not be ideal for something as esoteric as HPC storage configurations, but ML methods *can* produce surrogate models that help us explore the parameter space to find “good” solutions more quickly, repeatably, and predictably.



RECAP

- Storage device technology, data use cases, and the HPC user community are all rapidly evolving. These are great opportunities!
- The HPC storage community must embrace this evolution and strike a balance between research and practice to maximize impact.
- Look to conceptual frameworks like “translational computer science[1]” for inspiration on how to turn research into practice.

Driftwood Beach, Jekyll Island, Georgia, USA. It’s not actually “driftwood” at all; the beach is an oak forest that failed to adapt to change in the form of saltwater encroachment. Very pretty, though!

[1]D. Abramson and M. Parashar, "Translational Research in Computer Science," in *Computer*, vol. 52, no. 9, pp. 16-23, Sept. 2019

THANK YOU!

THIS WORK WAS SUPPORTED BY THE U.S. DEPARTMENT OF ENERGY, OFFICE OF SCIENCE, ADVANCED SCIENTIFIC COMPUTING RESEARCH, UNDER CONTRACT DE-AC02-06CH11357.