SC19
Denver, CO | hpc is now.

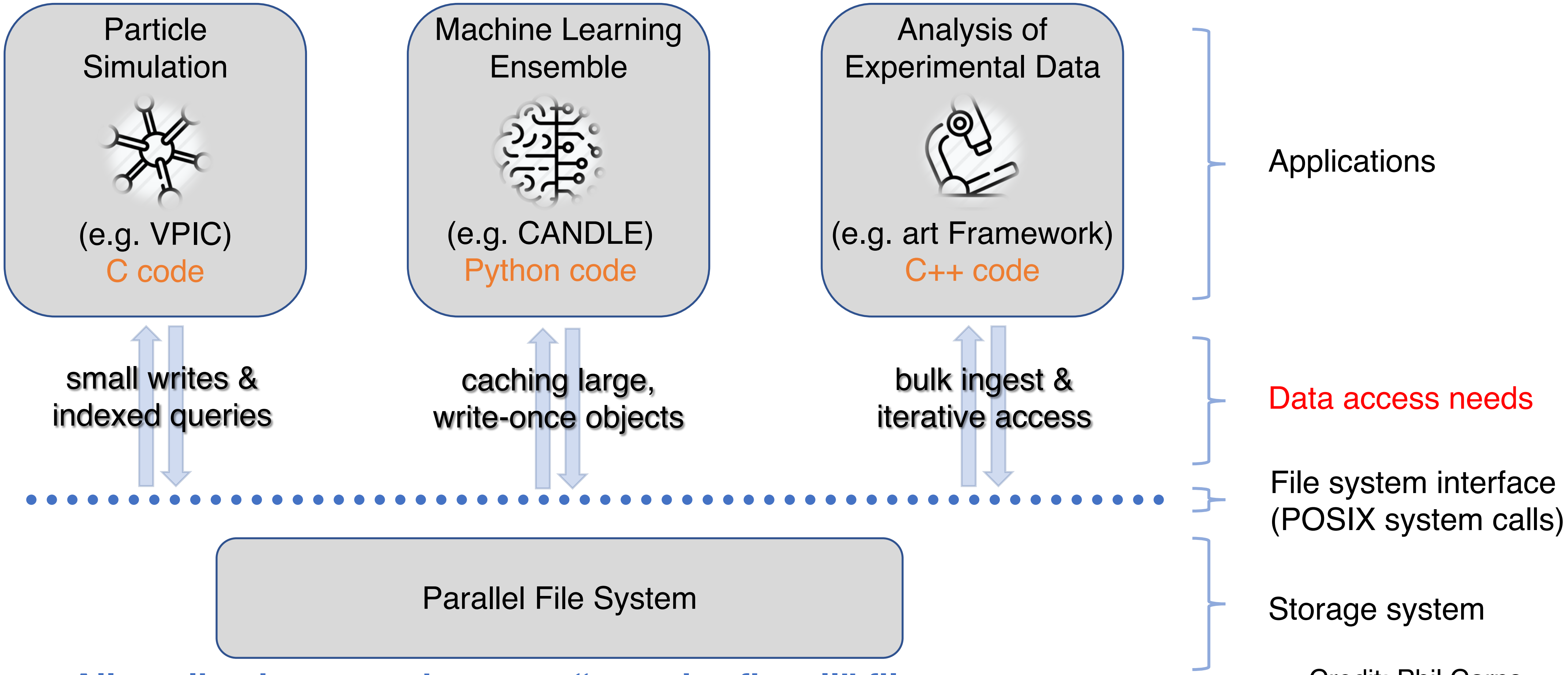
Enabling Data Services for HPC

Jerome Soumagne (The HDF Group)
Phil Carns (Argonne National Laboratory)
Mohamad Chaarawi (Intel Corporation)
Kevin Huck (University of Oregon)
Manish Parashar (Rutgers University)
Robert B. Ross (Argonne National Laboratory)

Background and Objectives

- **Why do we need data services in HPC**
  - Hopefully this BoF will answer that!
  - Why can’t we simply port cloud services *as-is*
- **What are the current challenges**
- **Can we find solutions and take a common direction**
- **Terminology**
  - What is a data service
    - Component / set of components that provide a feature / set of features to the user in response to an application need
  - Monolithic data service ⇔ Micro-services

File system monoculture for data (dis)service



Particle Simulation

(e.g. VPIC)

C code

Machine Learning Ensemble

(e.g. CANDLE)

Python code

Analysis of Experimental Data

(e.g. art Framework)

C++ code

small writes & indexed queries

caching large, write-once objects

bulk ingest & iterative access

Parallel File System

Applications

Data access needs

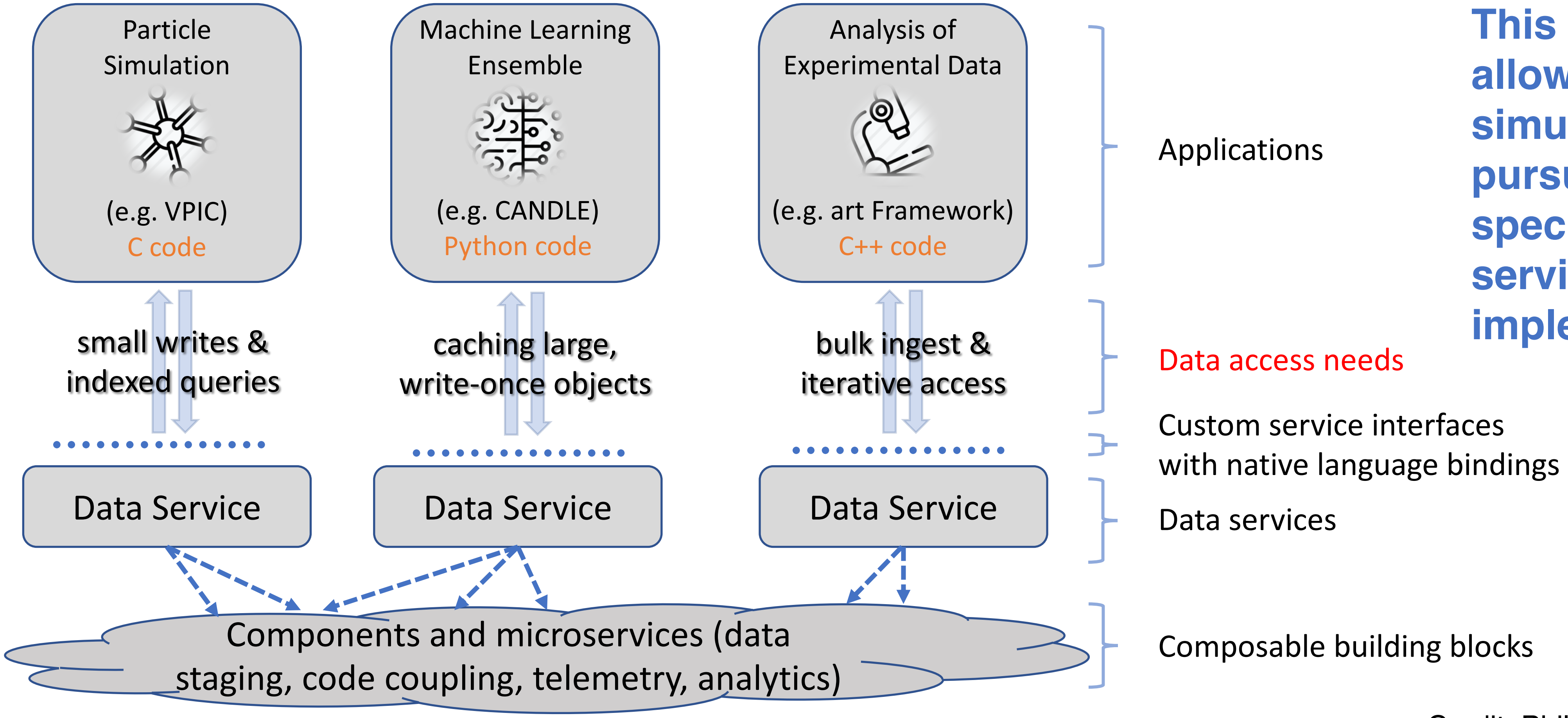
File system interface (POSIX system calls)

Storage system

Credit: Phil Carns

All applications use the same “one size fits all” file system interfaces, semantics, and policies for data access.

Ecosystem of services co-existing and reusing functionality



Particle Simulation

(e.g. VPIC)

C code

Machine Learning Ensemble

(e.g. CANDLE)

Python code

Analysis of Experimental Data

(e.g. art Framework)

C++ code

small writes & indexed queries

caching large, write-once objects

bulk ingest & iterative access

Data Service

Data Service

Data Service

Components and microservices (data staging, code coupling, telemetry, analytics)

Applications

Data access needs

Custom service interfaces with native language bindings

Data services

Composable building blocks

This approach allows to simultaneously pursue multiple specialized service implementations

Credit: Phil Carns

Instead of "one size fits all", data services can present tailored interfaces, semantics, and policies for data access while still leveraging robust building blocks.

Advantages and Challenges

- Multiple services = Customization of environment
  - Add value to vendor-provided capabilities
- Services can allow for re-usability of functionality

- Complexity of deep layers complicates performance tuning
  - Tailoring to applications has performance wins, but diagnosing and tuning requires additional tools.
- Gaining the trust of users and facilities
  - Teams can be reticent of trusting new services with their data, especially when long-term sustainability of software can be uncertain.

Discussion Themes

- **Hardware and Facilities: trends and challenges**
  - Where do new technologies drive change?
- **Software: development, test, scaling, maintenance challenges**
  - How do we adapt distributed services to perform well at scale and in heterogeneous environments?
- **User/developer Adoption: barriers and challenges**
  - How do we help scientists manage and relate the different data used in their workflows?
- **Vision and long-term directions**
  - Where are we going?

Format

- **Panel with 5 representatives and different perspectives**
  - Application
  - Facility
  - Hardware
  - Research

- **Panelists**
  - André Brinkmann (Johannes Gutenberg - Universität Mainz)
  - Carlos Maltzahn (University of California, Santa Cruz)
  - Stéphane Ethier (Princeton Plasma Physics Laboratory)
  - Glenn Lockwood (National Energy Research Scientific Computing Center)
  - Paolo Faraboschi (Hewlett Packard Enterprise)

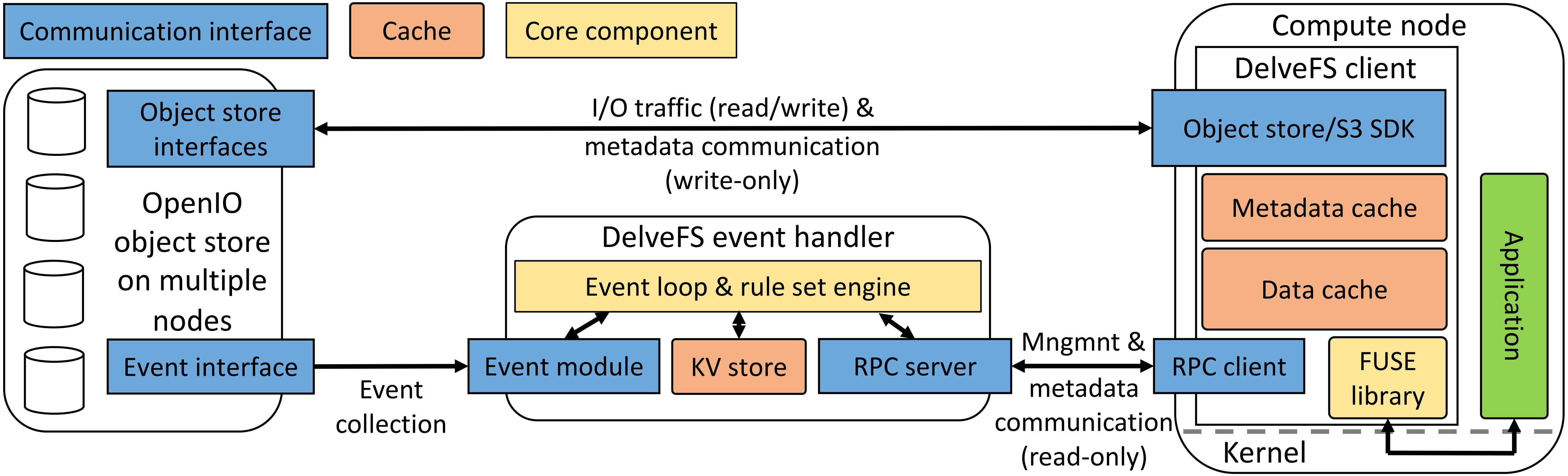
André Brinkmann (JGU)

Delve – Event-driven Workflows

- **What is a distributed data service?**
  - Data is consumed outside of an HPC job
  - Processing can be triggered by scheduler/job **or by the data itself**
- **Similarity to micro-service architectures**
  - Specific functionality is (only) launched to transform data
  - Transformation can trigger additional services (see AWS lambda)
- **Infrastructure for Distributed Data Services**
  - Data can be stored either in object store or file system
  - Relation between data and operations must be described
  - Database required to learn about data
  - Data storage must provide event interface

- **Provide infrastructure to connect to arbitrary data service workflows**

DelveFS – Bridging between objects and files



Carlos Maltzahn (UC Santa Cruz)

Declarative Data Services

SC19 Denver, CO | hpc is now.

- **Carlos Maltzahn, UC Santa Cruz**
  - Area of focus: Research in Programmable Storage Systems
    - Physical Design Management in Storage Systems
    - Eusocial Storage Devices
    - Reproducibiity-enabling infrastructures (see Maricq et al. OSDI'18)
  - Current hardware and facility needs
    - Shared Storage Testbeds spanning embedded, edge, cloud, and HPC environments
  - Current data management software needs
    - Access libraries witih plugin infrastructures (e.g. HDF5/VOL)
  - Vision / direction
    - Declarative configuration of data services such as physical design managemen
    - Production systems that support reproducibility
    - Production systems that enable deployment and testing of experimental storage systems

CROSS
CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE
cross.ucsc.edu

Baskin Engineering UC SANTA CRUZ

Successes / Challenges

- **Challenge: *access libraries***
  - Hard-wired assumptions about storage backends
    - Storage device performance characteristics: multi-tiered, heterogenous systems
    - Storage system functions: availability of filter, index, aggregation, and other data reduction operations
  - Data movement due to lack of context needed for computation
    - Data movement uses network resources *and* CPU resources
    - Semantic data partitioning to enable local (storage-side) computation
- **Challenge: *shared infrastructures for research***
  - Traces (ideally with datasets)
  - Statistical properties of infrastructure
  - Software-defined systems
  - Large variety of new storage devices

CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE
cross.ucsc.edu

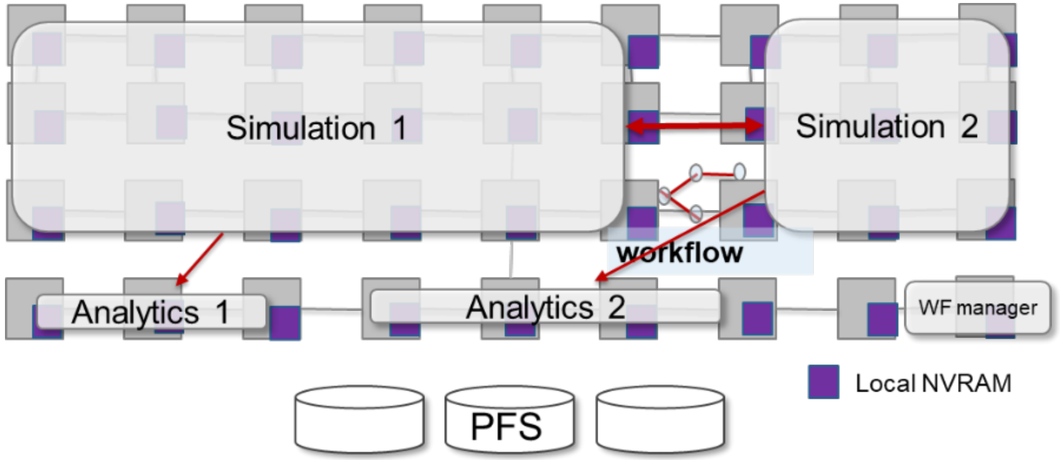
Stéphane Ethier (PPPL)

ECP WDMApp (fusion app)

- **Stephane Ethier, principal Comp. Scientist, PPPL**
  - HPC and data analysis
    - Application developer focusing on upcoming hardware, new programming models, HPC optimization
    - Development of data analysis routines, workflow user.
  - Currently working on Summit and Cori
  - We are developing a “whole device model” that couples many independently-developed codes together
  - This WDM code needs to run at exascale and include as much physics as possible

Successes / Challenges

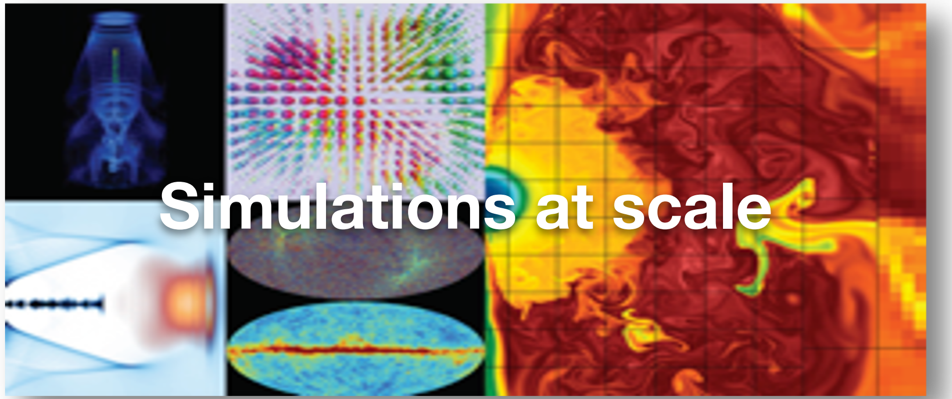
- **(Pick one or multiple of these to fill in and discuss)**
  - The current way of running several executables on the LCF systems (including NERSC) is to have each one on separate nodes, instead of sharing some of the nodes
  - This limits our flexibility to maximize resources and minimize communication costs
  - Why is it so hard to take a few threads on a node to run an analysis that runs as a separate executable?



Glenn Lockwood (NERSC)

National Energy Research Scientific Computing Center

- **NERSC is the mission HPC facility for the US DOE Office of Science**
  - Support workflows: traditional simulation, experimental data analysis, and/or artificial intelligence
  - 7,000 active users, 700 projects, 700 apps
  - Users from across almost all science domains
  - 2,500 publications in 2018
  - 5.8 billion CPU hours (25% on capability jobs) in 2018
  - > 1.0 exabyte of I/O in 2018
- **Glenn is a storage architect**
  - Define, design, procure, deploy, operate all storage tiers
  - Determine strategic directions, investments, technologies related to I/O



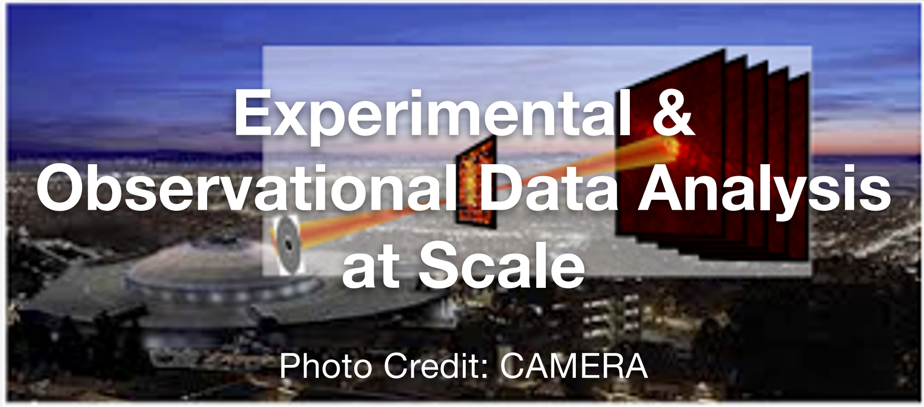
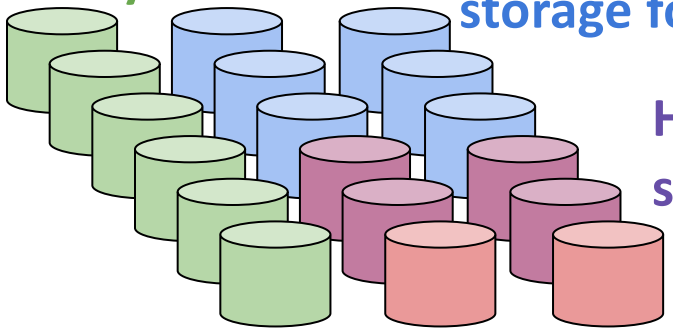


Photo Credit: CAMERA

Chemical Science
computing
NERSC
THE JOURNAL OF PHYSICAL CHEMISTRY B
Water Resources Research
THE JOURNAL OF PHYSICAL CHEMISTRY C
jmr Journal of MATERIALS RESEARCH
pss b

Storage in the age of complex workflows

- **All-flash Lustre in five years would have 2x "performance," 3x capacity**
  - Users will face same problems
  - "I/O performance" != "peak IOR bandwidth"
- **Challenges: contention, scalability, responsiveness**
- **Solutions exist at cost of peak bandwidth**
  - storage QOS
  - latency-optimized data paths
- **Smart storage: common hardware, reconfigurable software**
  - performance balances for different usage patterns
  - optimize storage on demand



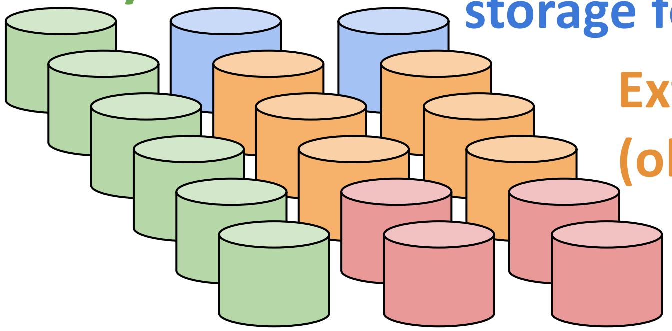
Global scratch for everyone

"Node-local" storage for job 1

High IOPS (3x copy) storage for job 2

Unreserved

Later that day...



Global scratch for everyone

"Node-local" storage for job 3

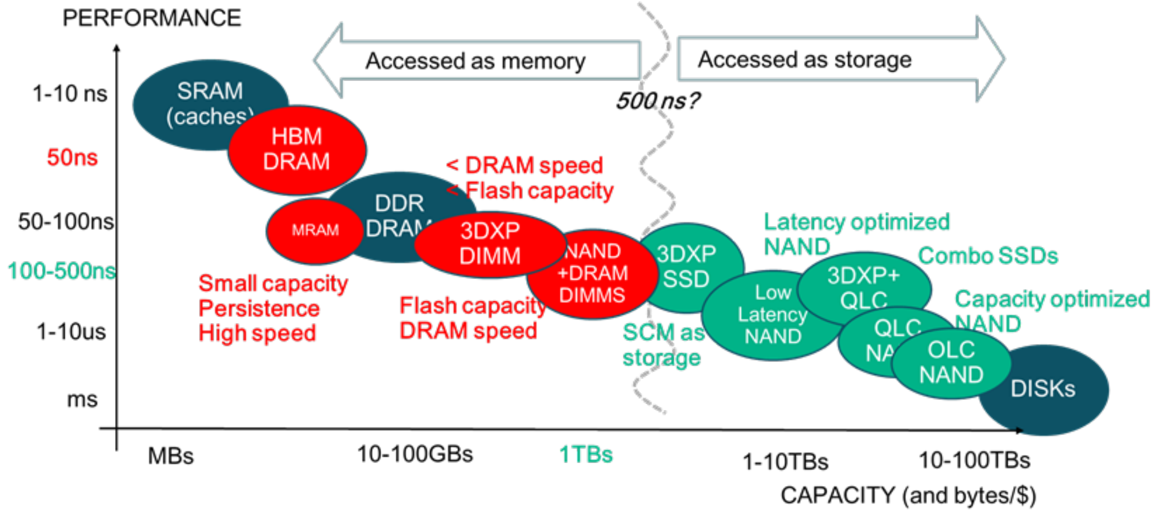
Extreme performance (object store) for job 4

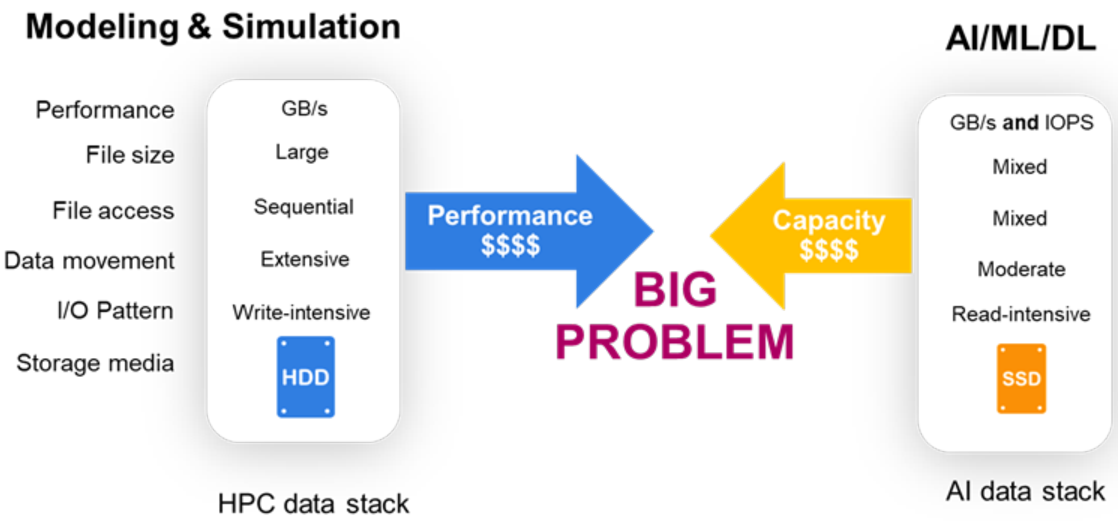
Unreserved

Paolo Faraboschi (HPE)

The HPC+AI Data Continuum

- **Paolo Faraboschi**
  Fellow and VP, Hewlett Packard Labs
  - System architecture research for HPC + AI
  - Hardware trends
    - Extreme heterogeneity in media and access protocols / interconnects
  - Data management trends
    - Need for a converged data stack; reconcile different drivers and requirements



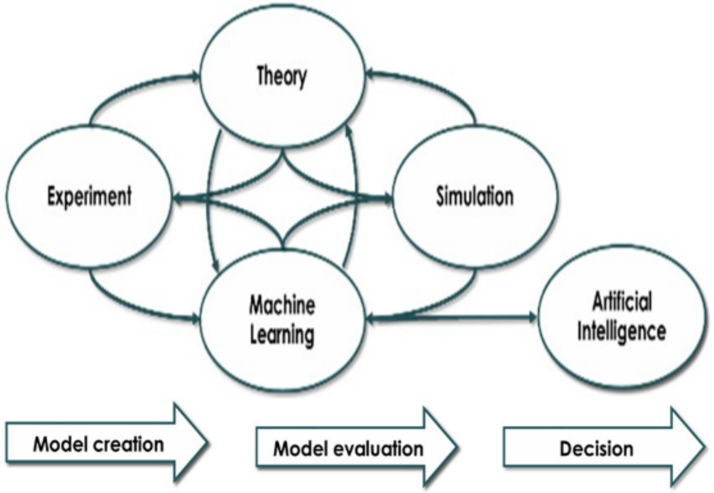


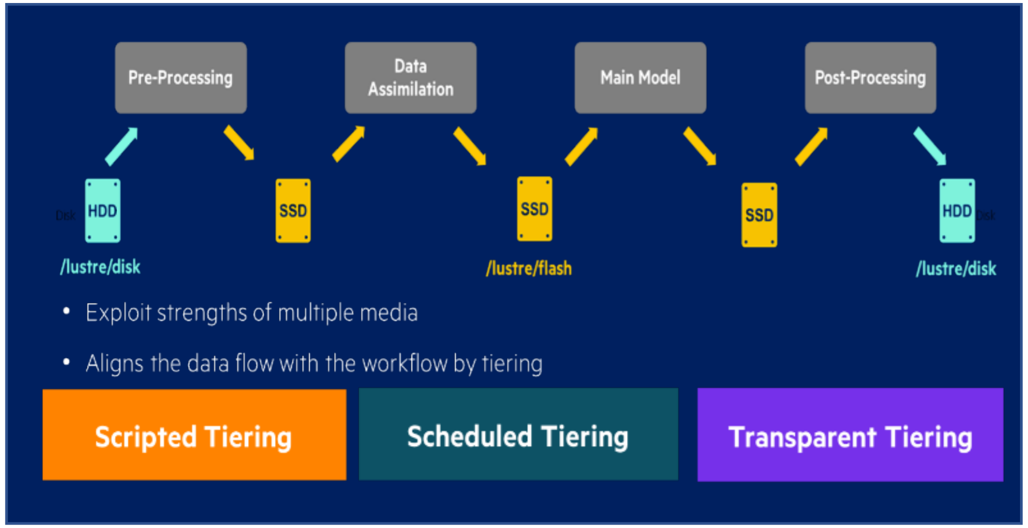
Modeling & Simulation	HPC data stack	AI data stack
Performance	GB/s	GB/s and IOPS
File size	Large	Mixed
File access	Sequential	Mixed
Data movement	Extensive	Moderate
I/O Pattern	Write-intensive	Read-intensive
Storage media	HDD	SSD

AI/ML/DL

Challenges and Opportunities

- ***"I've been surprised by"* the complexity of converged HPC+AI workloads**
  - AI straddles across scientists and experimentalists, edge & datacenter
- ***"I wish there was a way to"* automate data orchestration to match the AI+HPC workflows**
  - Indexing, search engine, policy management, tiering engine, transparent data movers, container orchestration, etc.
  - Embrace heterogeneity, align data flows & tiering to workflow





- Exploit strengths of multiple media
- Aligns the data flow with the workflow by tiering

Q&A

Hardware / Facilities

- Where do new technologies drive change?

- **Can hardware and facilities support multiple services running on a system?**
  - More concurrency, current system limits?
  - Are future systems ready?
  - Scheduler issues?
- **Difficulty in running anything that is not MPI-based or not directly supported by vendor**
- **Can we allow for any type of data service?**
  - Which rules to follow? Guidelines for adding/developing new services?

Software

- **How do we adapt distributed services to perform well at scale and in heterogeneous environments?**

- **Communication and Deployment challenges**
  - Persistent or transient?
  - Running in user-space? Cross job coupling?
- **Resiliency and data recovery**
  - Response to service failure? Can we survive system restart?
- **Security**
  - How does this impact deployment?
- **Are we reducing or increasing software maintenance cost?**

User and developer adoption

- **How do we help scientists manage and relate the different data used in their workflows?**

- **Best way to advertise services to users?**
  - Which one fits their needs, etc.
- **Risk of user overwhelmed by collection of services?**
  - How to facilitate access to services to new users?
- **How to help developers leverage existing services?**

Vision and long-term direction

- **Where are we going?**

- **Need for adaptivity to control application resource consumption?**
  - Dynamic provisioning of resources?