



CSE 5449: Intermediate Studies in Scientific Data Management

Lecture 20: Proactive Data Containers

Dr. Suren Byna

The Ohio State University

E-mail: byna.1@osu.edu

<https://sbyna.github.io>

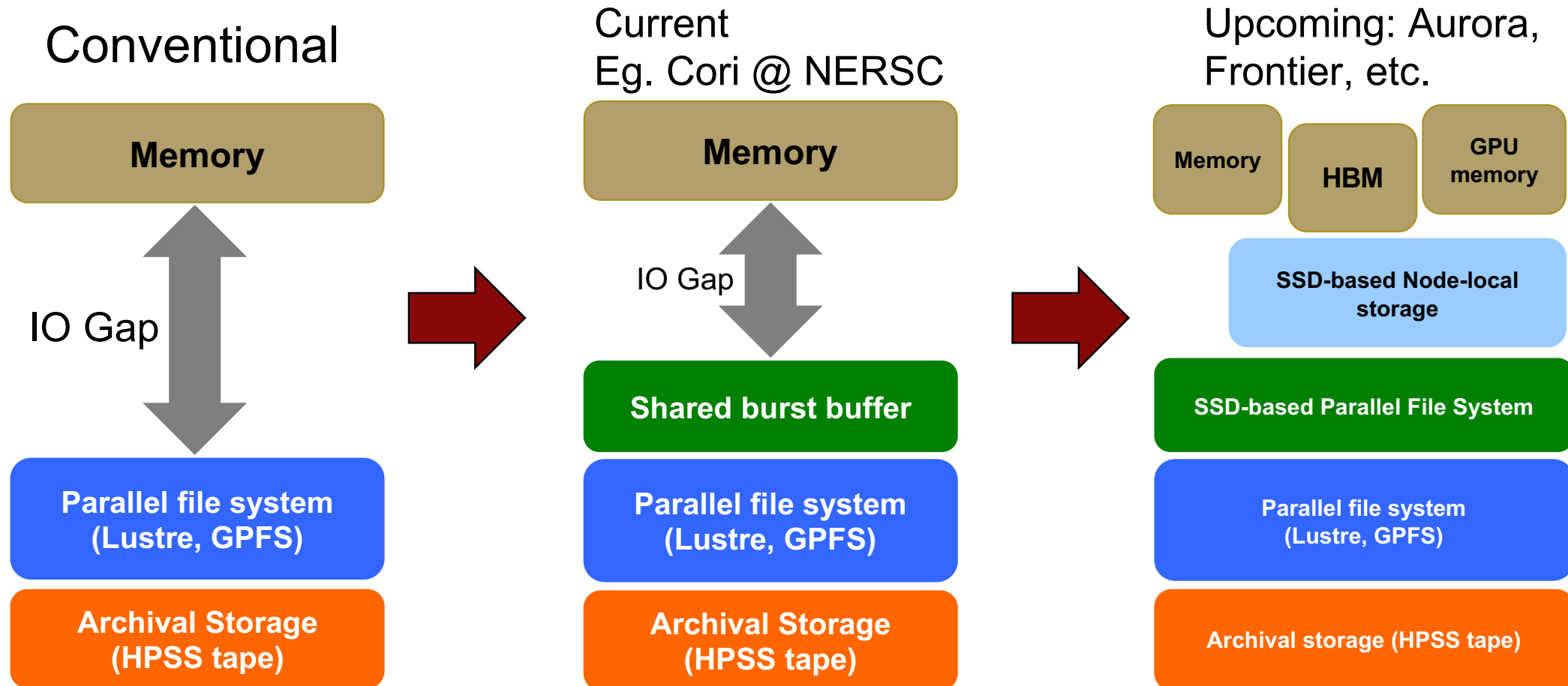
03/30/2023



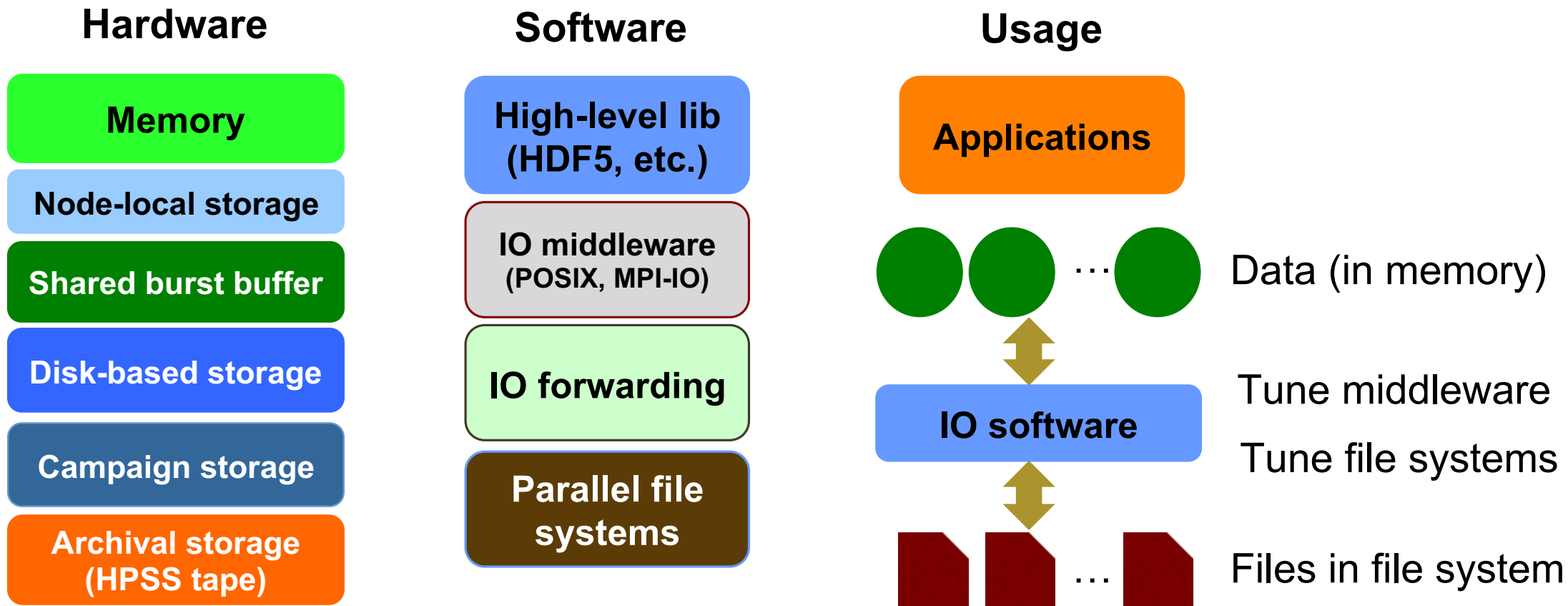
Today's class

- Any questions?
- Class presentation topic
- Today's class –
 - Introduction to Proactive Data Containers (PDC)

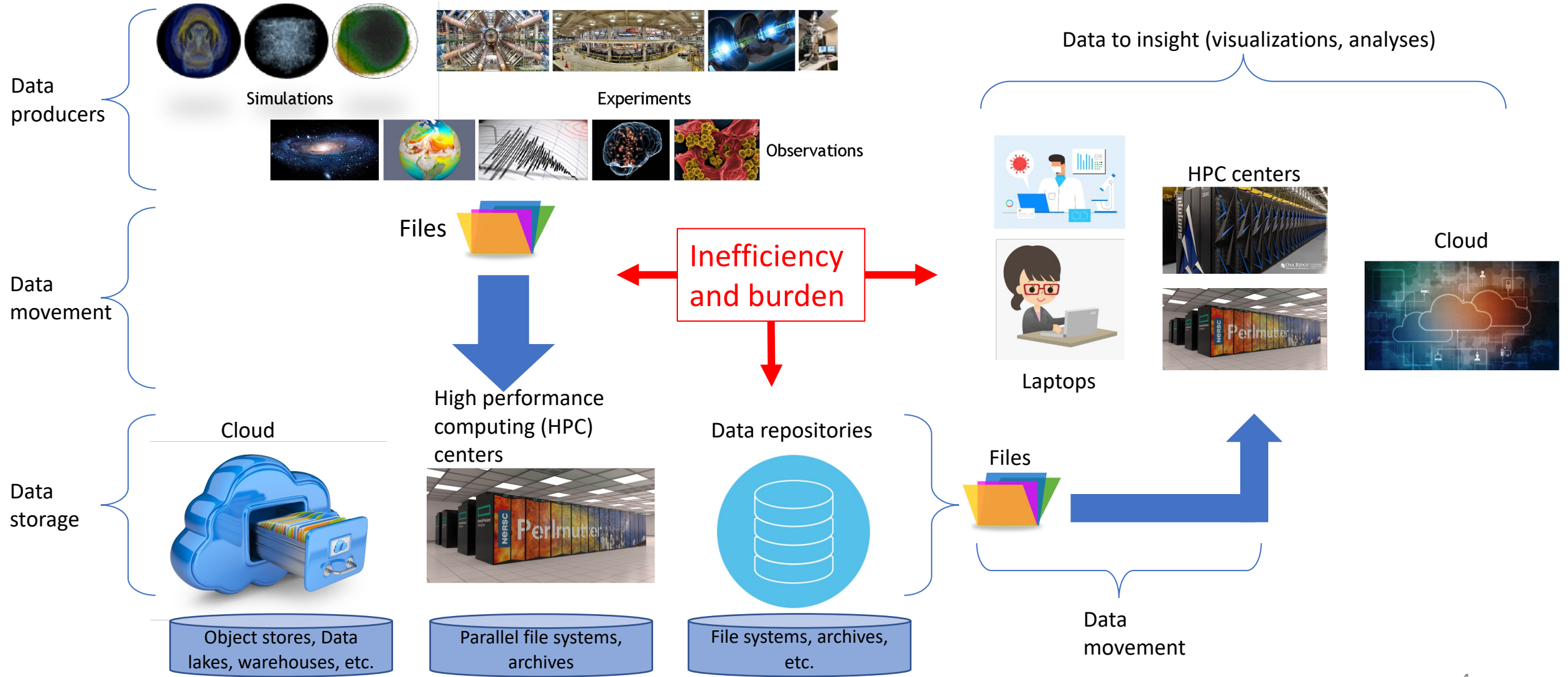
Storage systems in high performance compute systems



Storage Systems and I/O: Current status



Scientific data storage and access – File-based abstractions



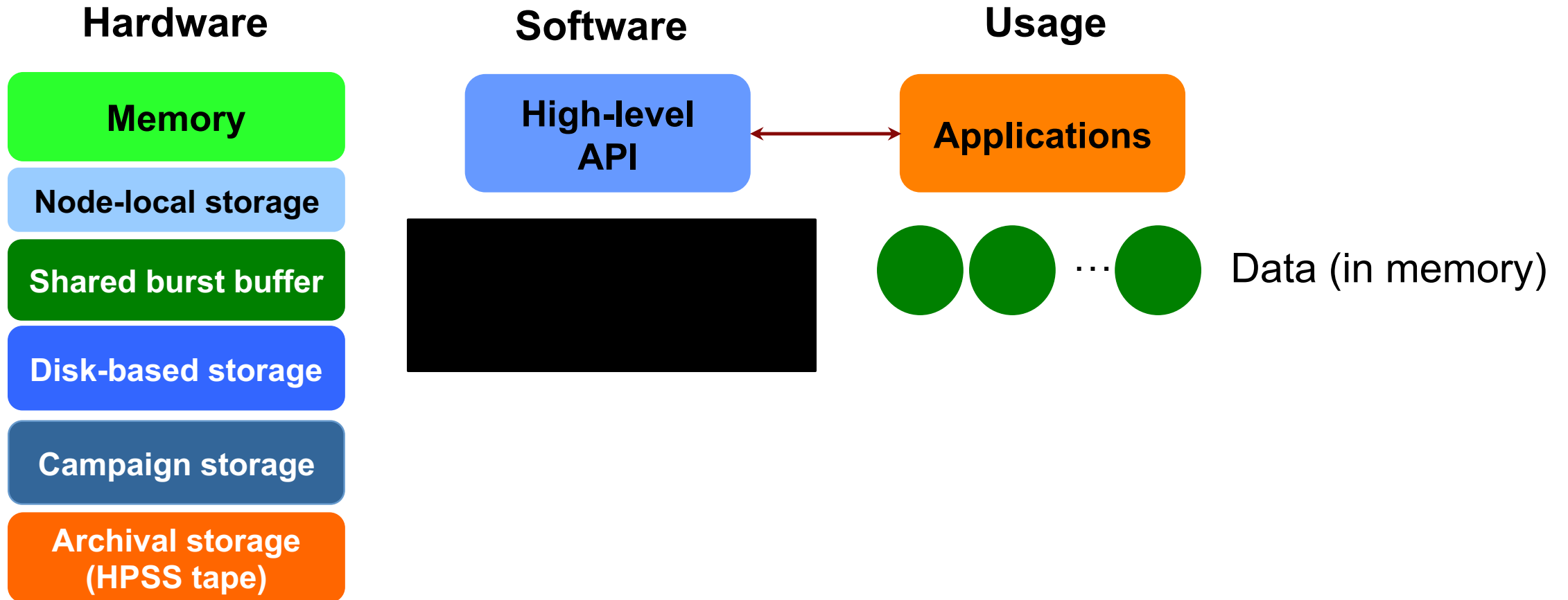
What do users want?

Use case	Domain	Sim/EOD/analysis	Data size	I/O Requirements
FLASH	High-energy density physics	Simulation	~1PB	Data transformations, scalable I/O interfaces, correlation among
Easy <u>interfaces</u> to complex systems				
CMB / Planck	Cosmology	Simulation,	10PB	Automatic data movement
<u>Autonomous</u> data movement and performance tuning				
				transformations
<u>Information</u> capture, management, and search				
TECA	Climate	Analysis	~10PB	Data organization and efficient data movement
HipMer	Genomics	EOD/Analysis	~100TB	Scalable I/O interfaces, efficient and automatic data movement

Users:

- Scientists
- App developers
- Supercomputing facilities
- System designers

Storage Systems and I/O: Next generation

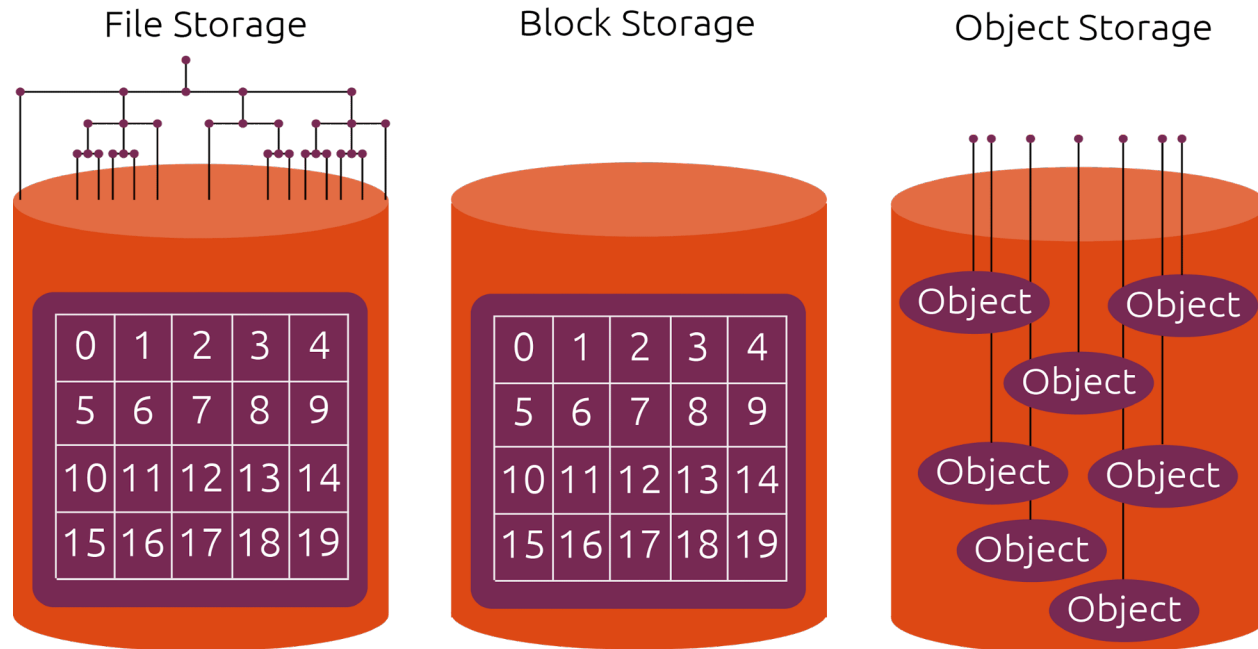




Next generation data management system - Requirements

- Transparent - Users are not required to perform explicit I/O or tune for performance
- Asynchronous I/O - Application is not blocked on data being moved to storage devices
- High performance and scalable
- Fault tolerant
- Autonomous - utilize memory and storage hierarchy, understand I/O usage, schedule I/O

Object-focused data management to the rescue?



Object storage

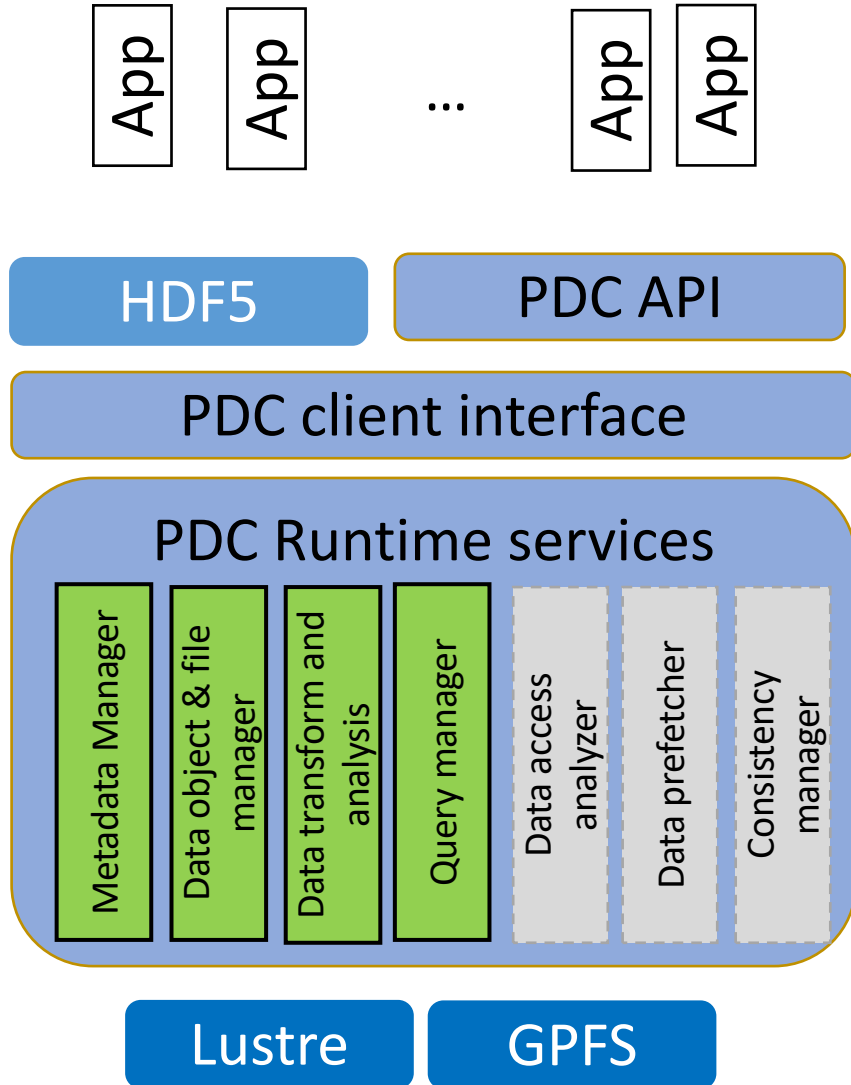
- Designed for unstructured data such as media, documents, logs, backups, application binaries and VM images
- Data objects are associated with metadata descriptions
- Common API: REST
- Examples: Ceph, Swift, Amazon S3, etc.



Objects - An overloaded term

Object management system	What does object mean?
Parallel file systems (Lustre, etc.)	Chunks of a file (block storage)
Cloud object storage (S3, etc.) OpenStack Swift, MarFS, Ceph, etc.	Files (images, videos, etc.) + metadata
HDF5 datasets, netCDF variables	Multi-dimensional arrays, images, any type of data associated with metadata – self-describing data
DAOS	Multi-dimensional arrays, files

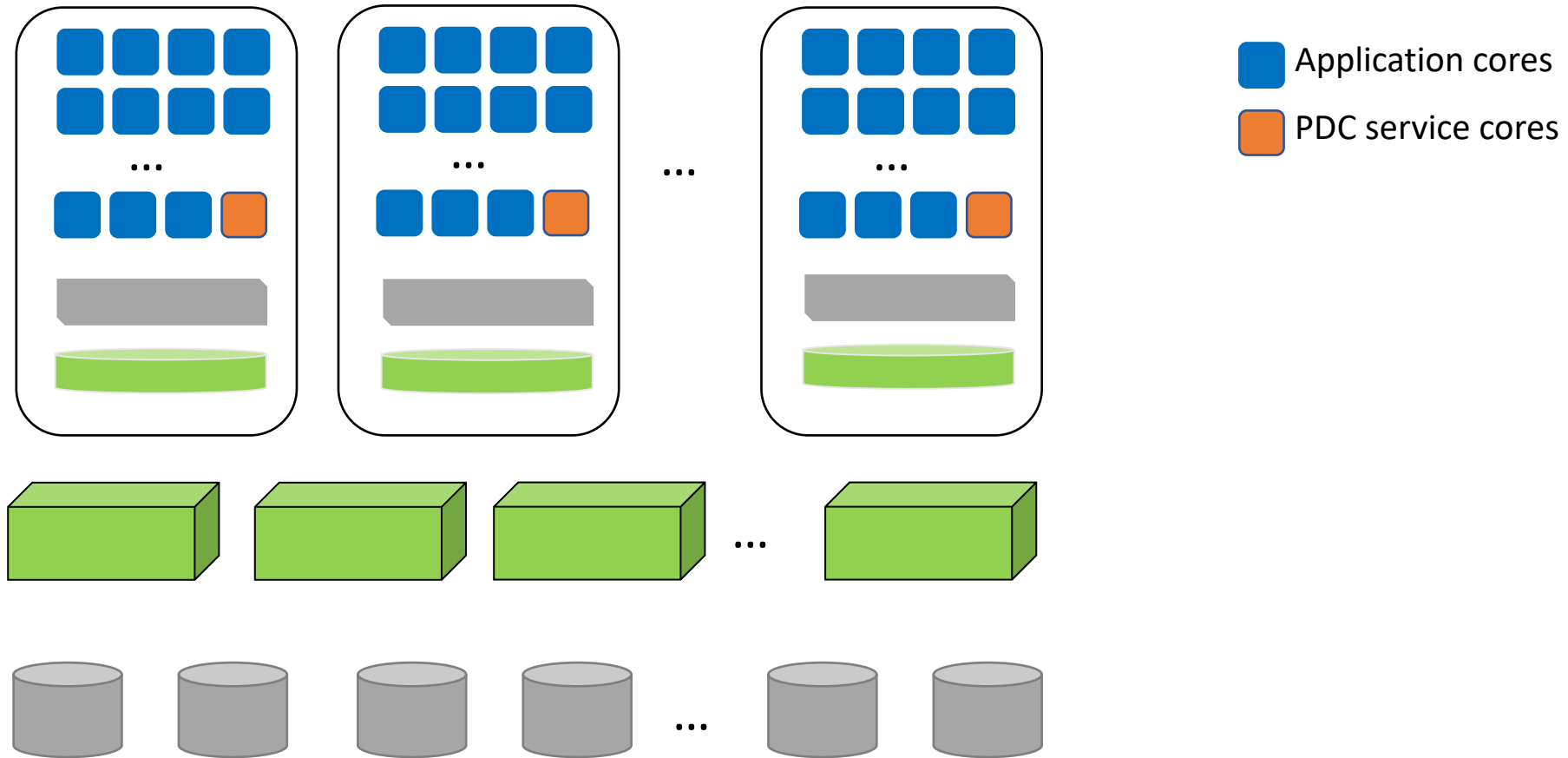
Proactive Data Containers (PDC): An autonomous object-centric data management services framework



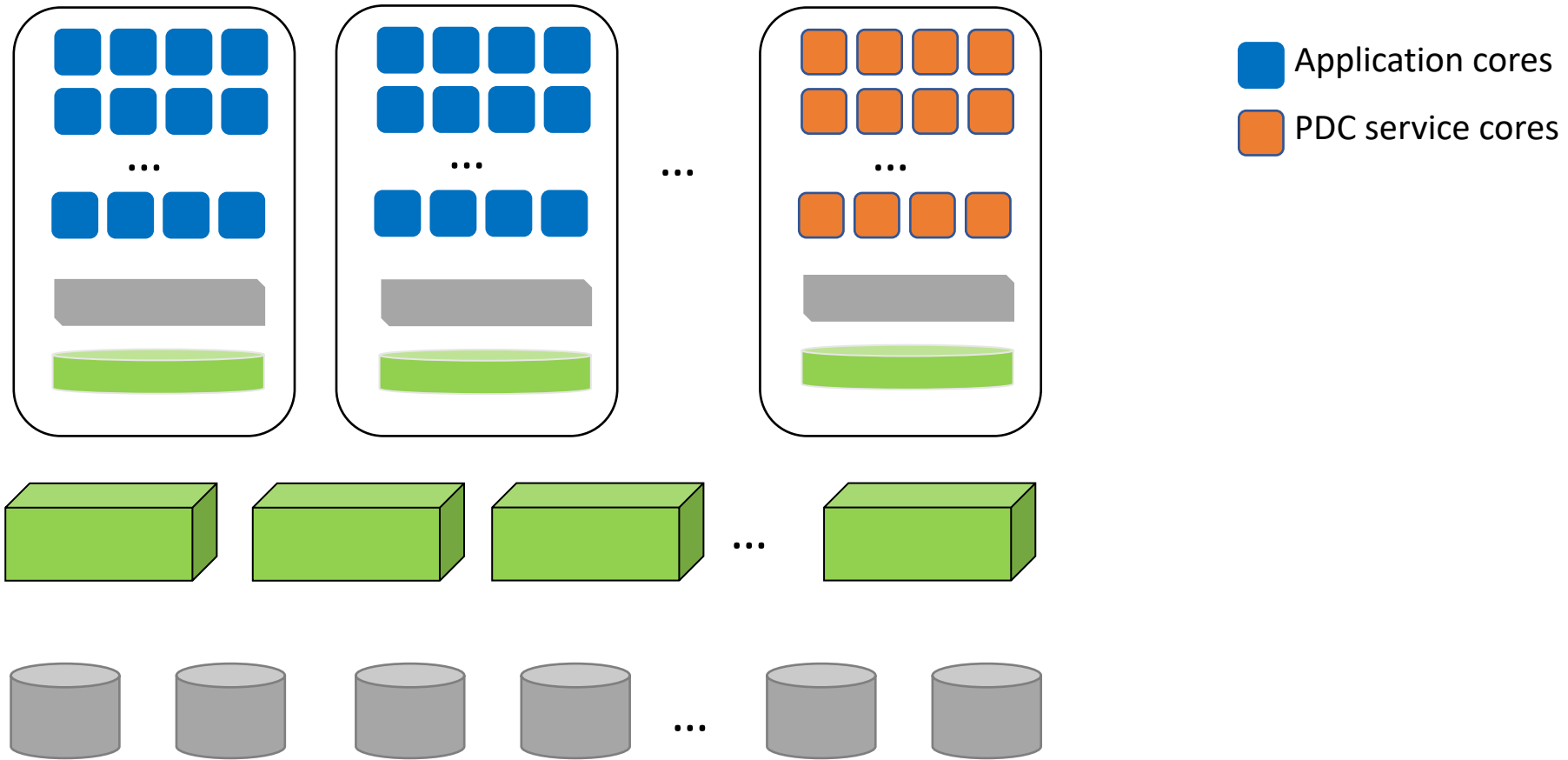
- Advantages of PDC

- Application-level object abstractions - Freedom from file management
- Transparent utilization of storage hierarchy and data movement
- Superior and scalable performance
- Live system for data management services
 - Metadata management, analysis, indexing and querying services, *consistency, data placement, etc.*

PDC modes of operation – Shared mode



PDC modes of operation – Dedicated mode



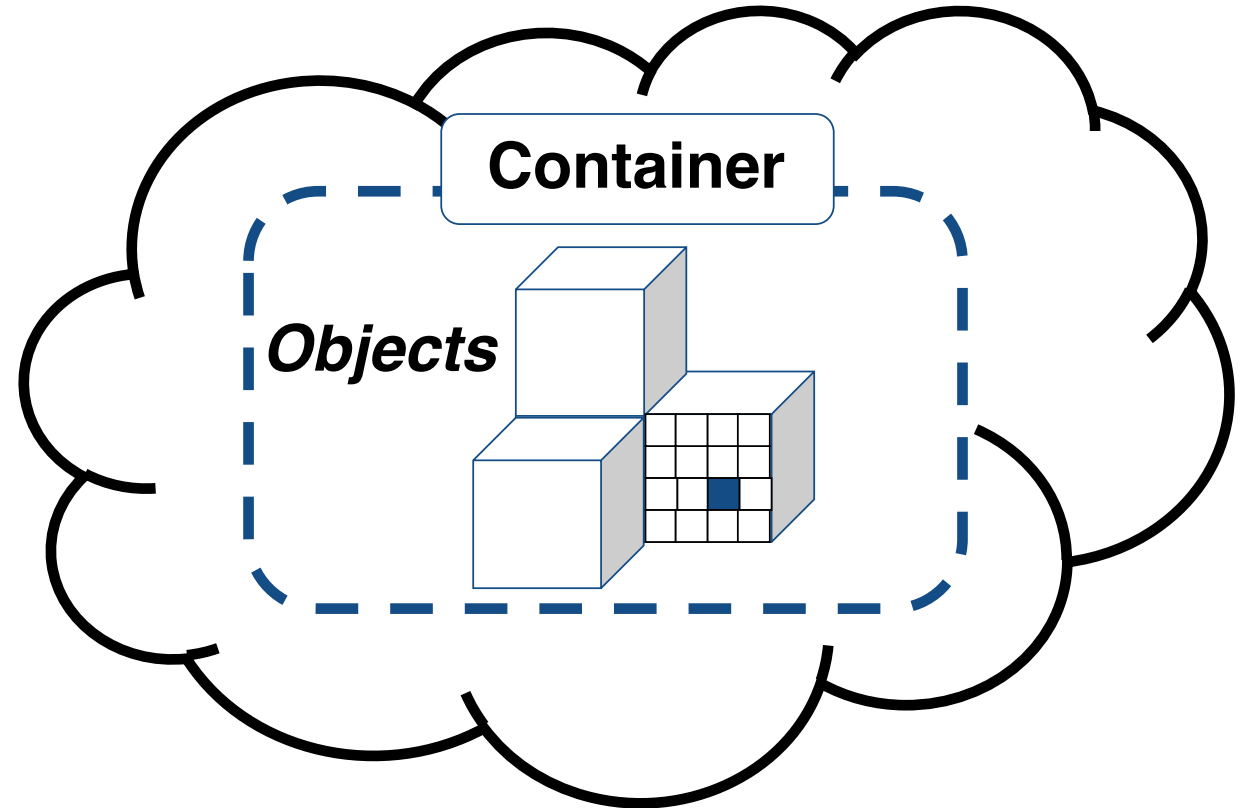


PDC Services

- **Object-centric abstractions**
- **Metadata management**
- **Data management**
- Querying
- Analysis in data path
- Bridge to HDF5 API

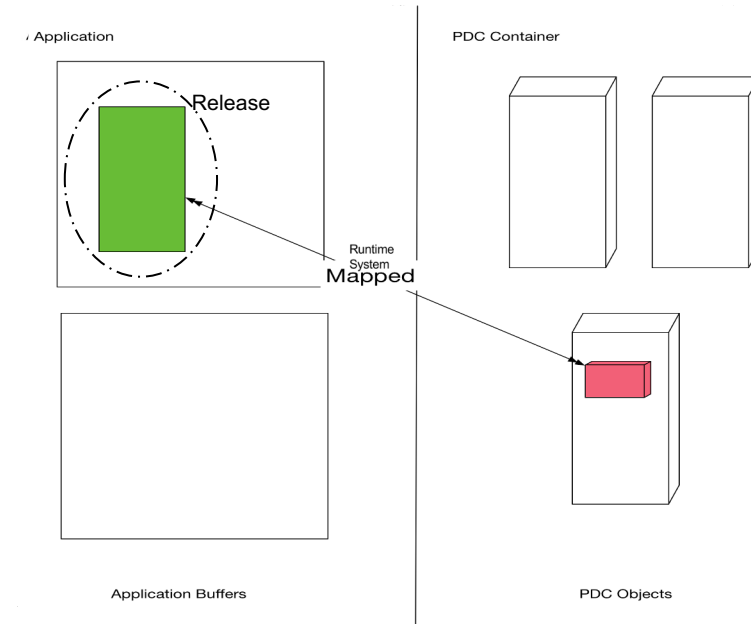
Object-centric abstractions

- Object abstractions
 - Container – special case of object, metadata only
 - Object - metadata and data payload
 - Arrays
 - *KV stores*
 - Regions
 - A region in a multi-dimensional array



PDC runtime system for asynchronous data movement

- User does not manage files; only creates and maps objects and regions
- Container
 - create container
 - delete container
 - add / delete objects
- Objects & Regions
 - create object
 - add metadata
 - create regions
 - map objects / regions from source to destination
 - Source and destinations can be memory or PDC spaces
 - lock when updating an in-memory object
 - release informs PDC runtime for implicit data movement
 - find object (followed by “map” for reading)
 - Explicit put and get object functions are also available
- Extensions for explicit asynchronous data movement
 - start data transfer / wait
- Allow diverse consistency modes
 - Eventual (PDC default), Session, Commit, POSIX



<https://github.com/hpc-io/pdc>



Summary of today's class

- Today's class: Proactive Data Containers
 - <https://github.com/hpc-io/pdc>
 - <https://pdc.readthedocs.io/en/latest/>
 - <https://sdm.lbl.gov/pdc/pubs.html>
- Next Class – Proactive Data Containers (PDC) – Metadata service
- Class project –
 - Status update on Apr 4th
 - Final presentation on Apr 20th
- Final exam on Apr 25th