



Connecting HDF5 to the Proactive Data Containers

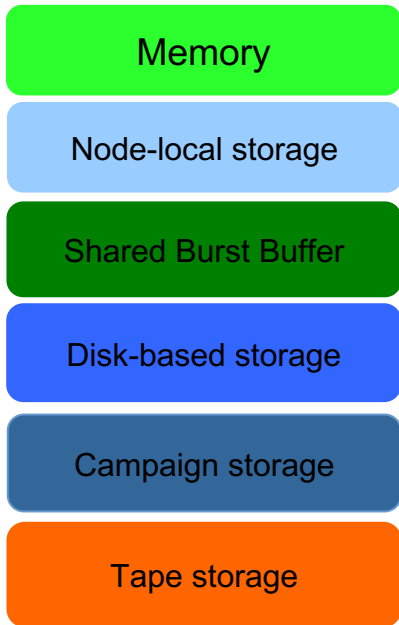
Houjun Tang

Lawrence Berkeley National Laboratory

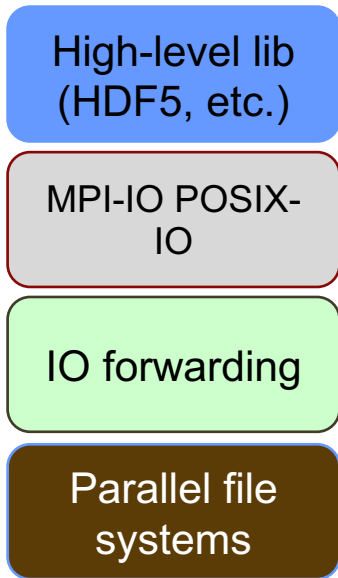


Storage Systems and I/O: Current status

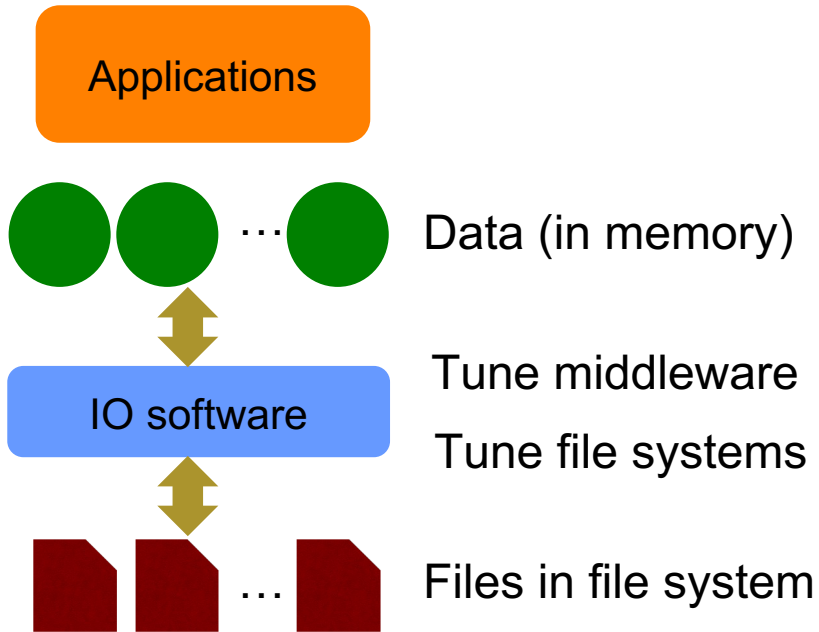
Hardware



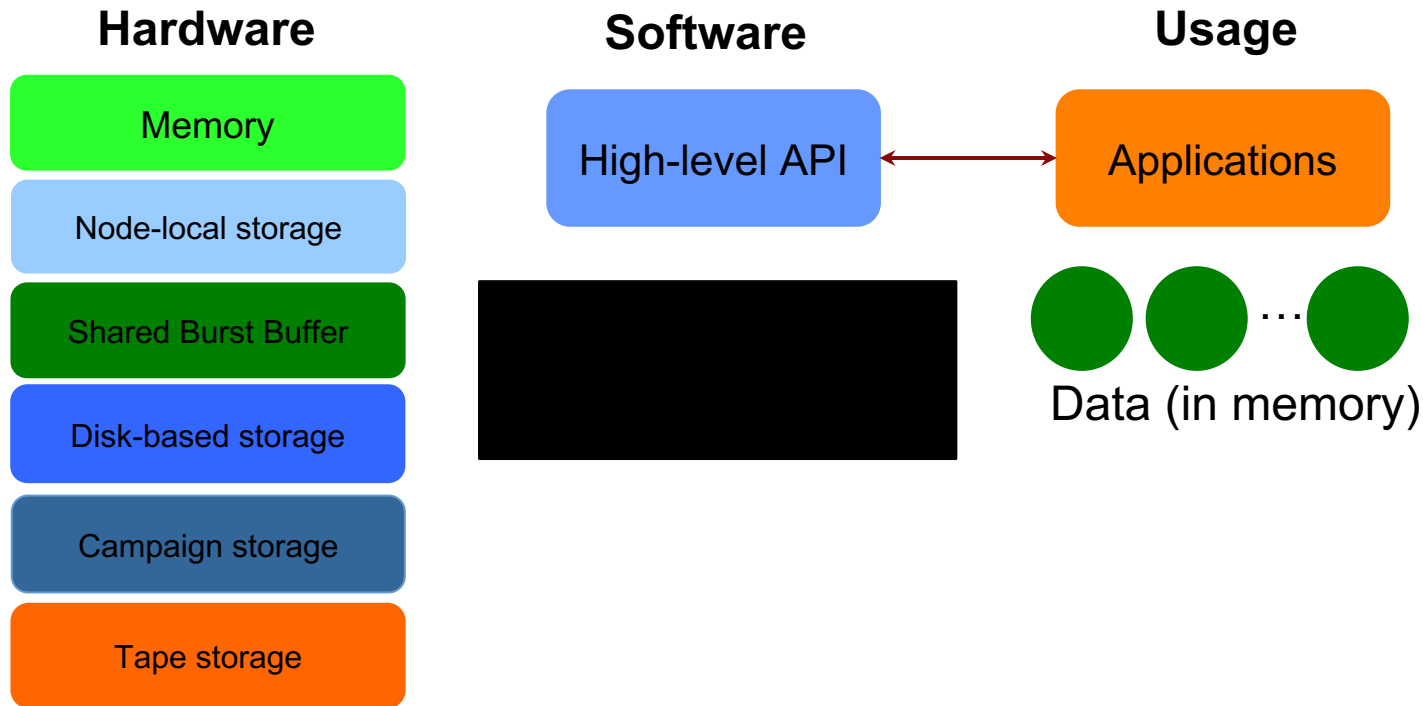
Software



Usage



Storage Systems and I/O: Next Generation





Storage Systems and I/O: Next Generation

- **Transparent, autonomous, and proactive data management**
 - Automatic object placement and data movement across storage layers.
 - Proactive data transformation and analysis.
- **Data management beyond POSIX restrictions**
 - Tunable consistency semantics.
 - Relaxed consistency provides more optimization opportunities.
- **Object-centric**
 - Direct data access without files.
 - Rich data and metadata accessible through queries.





Proactive Data Containers

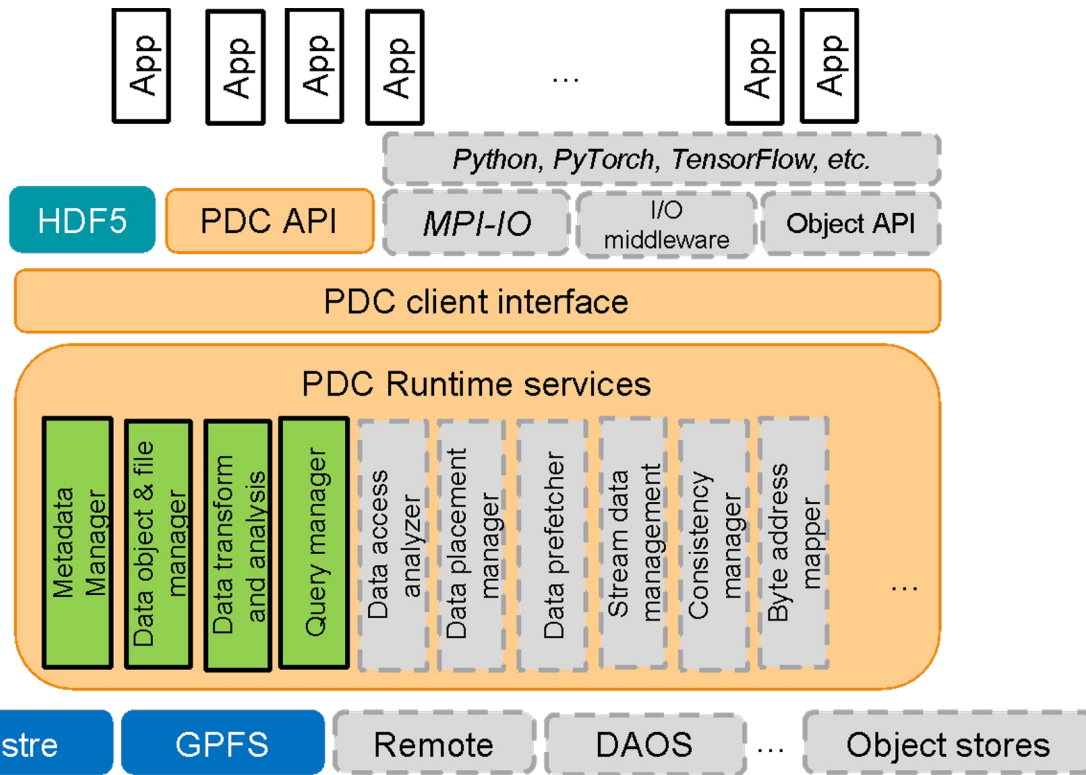
Data + Metadata + Data products +
Transformation and analysis operations



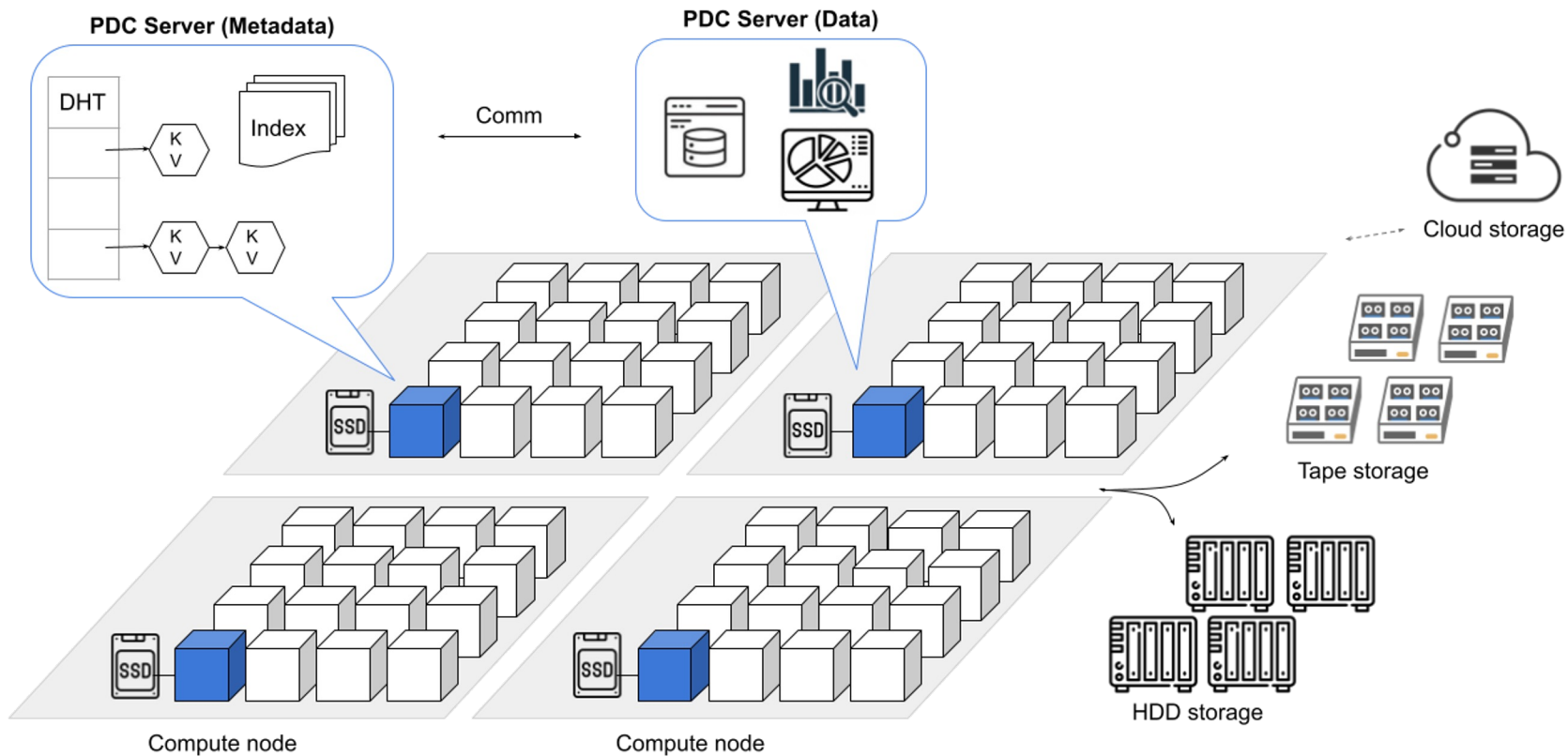
Proactive Data Containers (PDC)



PDC Components



PDC Runtime System



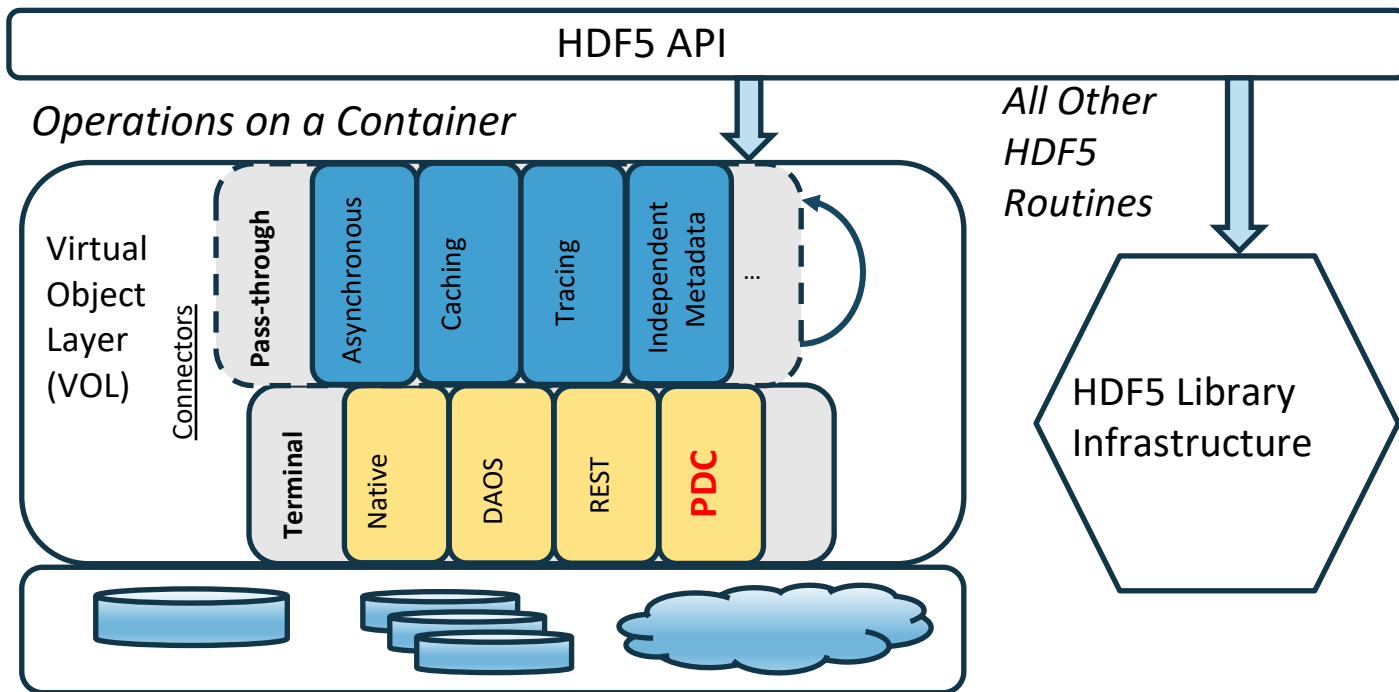


PDC Interfaces

- HDF5 PDC VOL Connector
 - Allows HDF5 applications to utilize PDC as a storage backend without code changes.
 - <https://github.com/hpc-io/vol-pdc>
- PDCpy provides a Python interface to utilize PDC
 - Uses NumPy array syntax that can be directly integrated with existing Python applications.
 - Currently support all PDC core functions, such as container, object, region, kvtag operations.
 - <https://github.com/hpc-io/PDCpy>
- PDC command line tools
 - pdc_ls displays all objects and their metadata in JSON format, with filter capability.
 - pdc_import/export imports and exports existing HDF5 files into the PDC system.
 - <https://github.com/hpc-io/pdc/tree/stable/tools>



HDF5 PDC VOL Connector





Mapping HDF5 to PDC

HDF5	PDC
File	- (in metadata)
Group	Container
Dataset	Object
Attribute	Metadata
Data (selection, chunk)	Data (region)
Property	Property





HDF5 PDC VOL Connector Implementation

```
H5Fcreate()
H5Gcreate(...)      cont_id = PDCcont_create(...)
H5Dcreate(...)      obj_id = PDCobj_create(cont_id, ...)
H5Screate_simple(...) local_reg = PDCreg_create(local_offset, local_count)
H5Screate_simple(...) remote_reg = PDCreg_create(remote_offset, remote_count)

H5Dwrite()          xfer_id = PDCtransfer_request_create(buf, acc_type, obj_id, local_reg, remote_reg);

                    PDCregion_transfer_start(xfer_id);
                    PDCregion_transfer_wait(xfer_id);
                    PDCregion_transfer_close(xfer_id);

                    H5Sclose(...)      PDCreg_close(local_reg);
                    H5Sclose(...)      PDCreg_close(remote_reg);
                    H5Dclose(...)      PDCobj_close(obj_id);
                    H5Gclose(...)      PDCcont_close(cont_id);
                    H5Fclose()
```

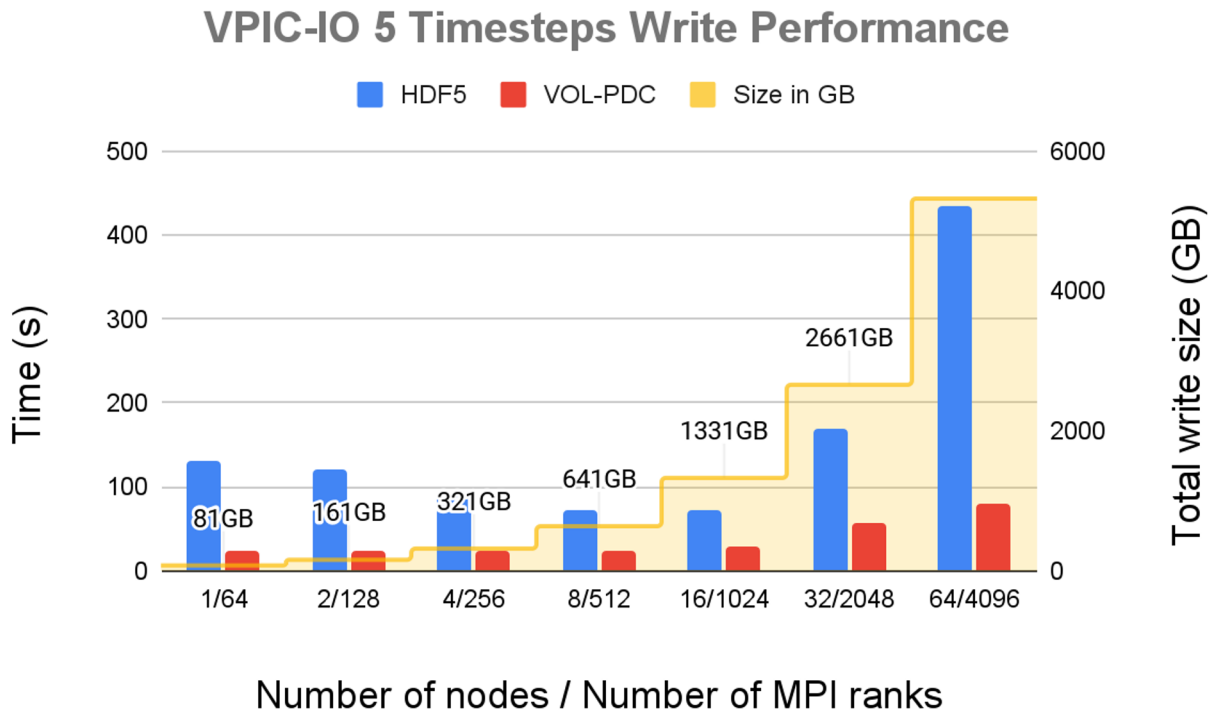
Local metadata copy only

Start async data transfer, server returns immediately after receiving requests

Wait for data transfer to be finished



Evaluation on Perlmutter with VPIC-IO





Conclusion and Future Work

- PDC is an object-centric data management system that provides data and metadata optimizations.
- HDF5 PDC VOL connector enables HDF5 applications to utilize PDC optimizations automatically and transparently. Performance benefits shown with the VPIC-IO benchmark.
- *Future Work*
 - Support HDF5 Asynchronous I/O APIs.
 - Extract hints from HDF5 properties and apply in PDC.
 - Performance evaluation on scientific workflows and applications.

