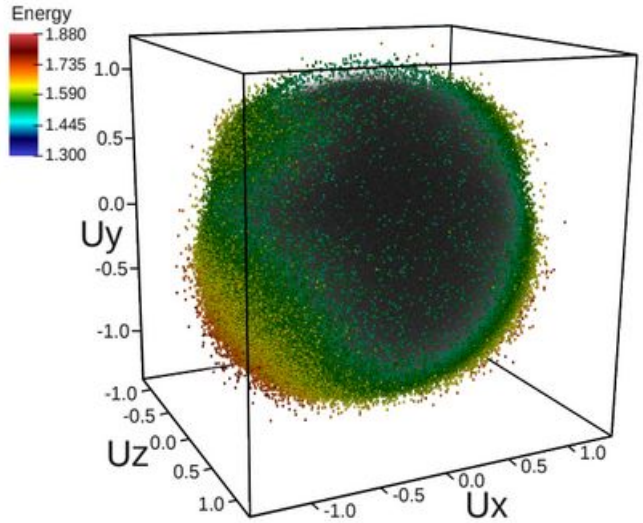
Parallel Query Service for Object-centric Data Management Systems

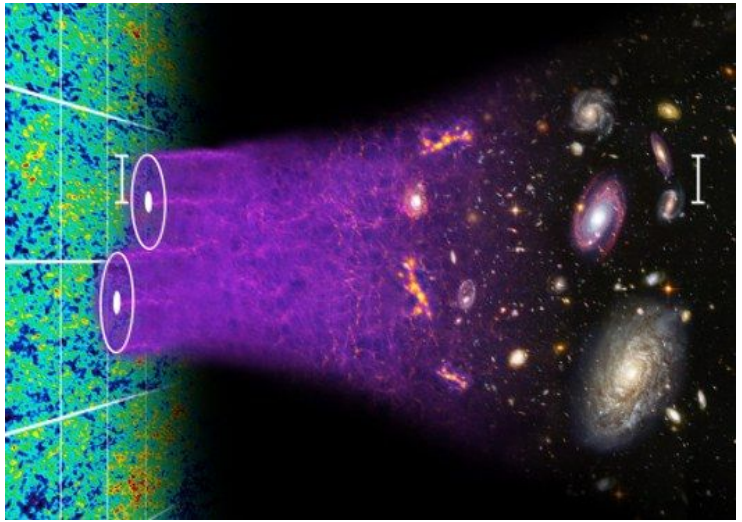
Houjun Tang, Suren Byna, Bin Dong, and Quincey Koziol
Lawrence Berkeley National Laboratory

Querying Scientific Data

Extract a **small fraction** of information from a **large amount** of data.



Vector Particle-In-Cell
3.3TB, 125 billion particles



Baryon Oscillation Spectroscopic Survey
3.2 TB, 25 million objects

Existing Query Solutions

BERKELEY LAB
Lawrence Berkeley National Laboratory

- DBMS, e.g. BerkeleyDB, PostgreSQL, MongoDB...
  - Efficient metadata queries.
  - Not optimized for multi-dimensional data queries.

- Multi-dimensional data indexing/querying system, e.g. SciDB, FastQuery
  - Targets large n-dimensional arrays, lack support for metadata queries.
  - Reading data may lead to significant overhead.

A **unified** data and metadata query system that provides **elastic**, **efficient**, and **scalable** query evaluations.

Current Data Management Systems

BERKELEY LAB
Lawrence Berkeley National Laboratory

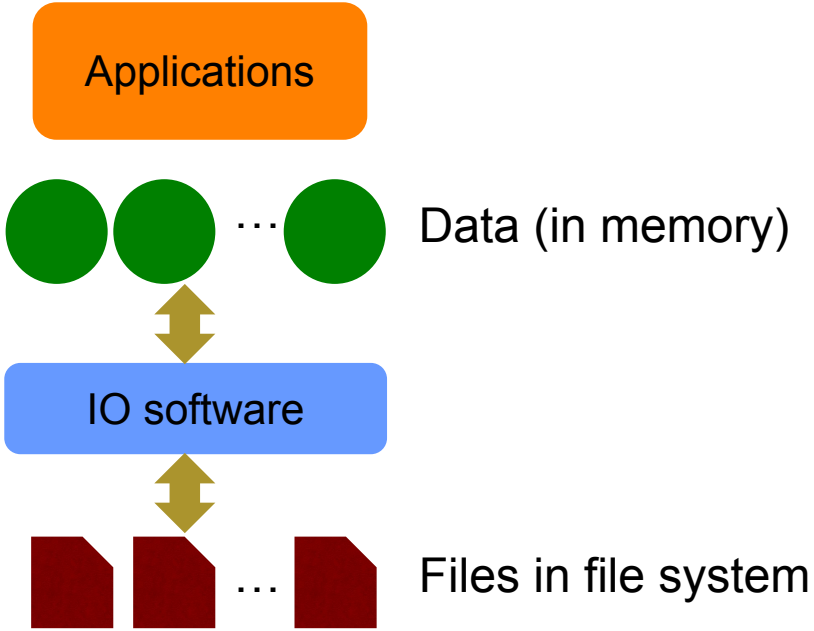
Hardware

- Memory
- Node-local storage
- Shared burst buffer
- Disk-based storage
- Campaign storage
- Archival storage (HPSS tape)

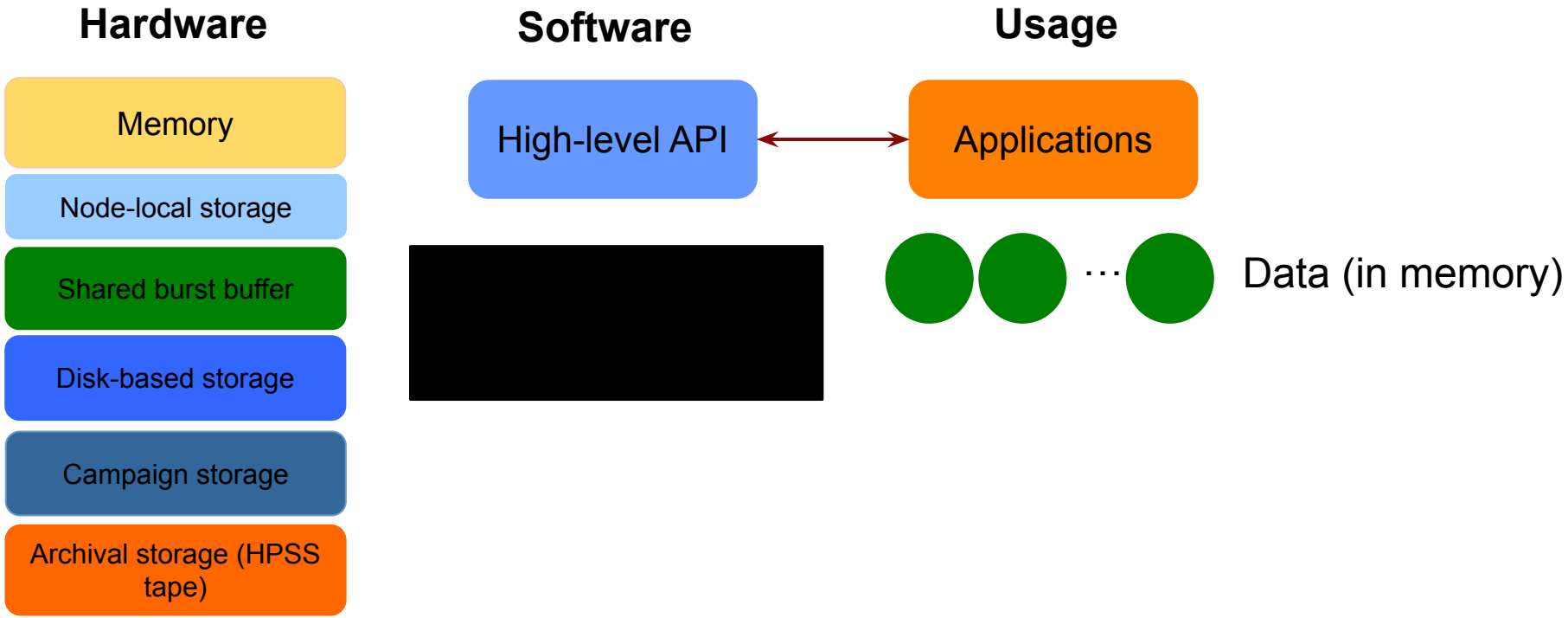
Software

- High-level lib (HDF5, etc.)
- IO middleware (POSIX, MPI-IO)
- IO forwarding
- Parallel file systems

Usage

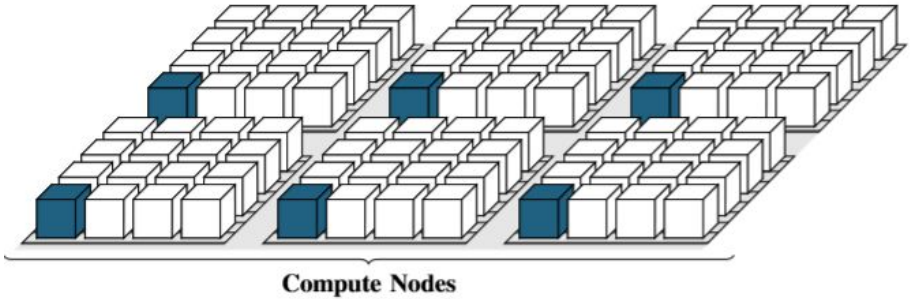


Object-centric Data Management Systems

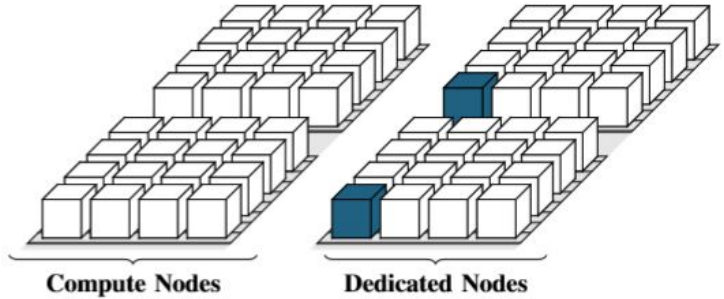


Data management in PDC

- PDC servers run in background, manages data and metadata.
- Data objects can be stored on different layers of memory hierarchy.
- Large data objects are decomposed into smaller regions.
- Metadata is cached in server's memory and persisted to storage.
- Application send requests through linked PDC client library.



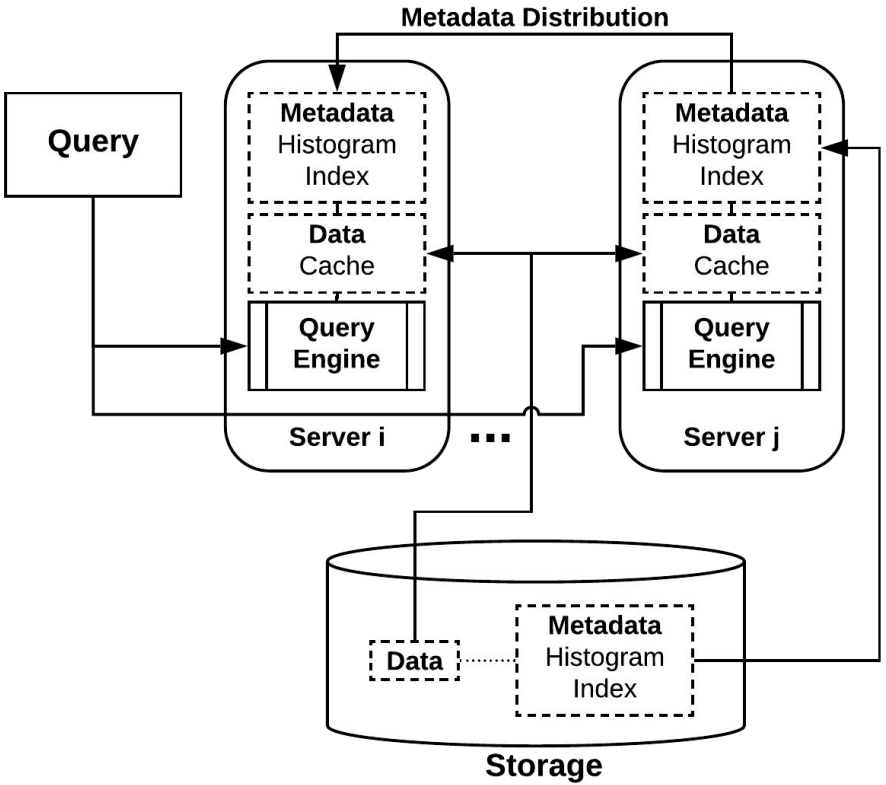
(a) Shared server modes: servers and clients are located on the same node.



(b) Dedicated server modes: servers are on separate nodes.

Queries in PDC

- **Metadata query**
  - Previous paper: “SoMeta: Scalable Object-Centric Metadata Management for High Performance Computing”
- **Data query**
  - Single variable
  - Multi variable
  - Get number of hits
  - Get selection
  - Get value



PDC-query Interface

BERKELEY LAB
Lawrence Berkeley National Laboratory

```
// Create a one-sided data query
pdcquery_t *PDCquery_create(pdcid_t obj_id, pdcquery_op_t op, pdctype_t type,
void *value);
// Combine queries
pdcquery_t *PDCquery_and(pdcquery_t *query1, pdcquery_t *query2);
pdcquery_t *PDCquery_or(pdcquery_t *query1, pdcquery_t *query2);
// Set query region constraint
perr_t PDCquery_set_region(pdcquery_t *query, pdcregion_t *region);
// Query operations
perr_t PDCquery_get_nhits(pdcquery_t *query, uint64_t *n);
perr_t PDCquery_get_selection(pdcquery_t *query, pdcselection_t *sel);
perr_t PDCquery_get_data(pdcid_t obj_id, pdcselection_t *sel, void *data);
perr_t PDCquery_get_data_batch(pdcid_t obj_id, pdcselection_t *sel, uint64_t
batch_size, void *data);
pdchistogram_t *PDCquery_get_histogram(pdcid_t obj_id);
```

Query Evaluation Strategies

BERKELEY LAB
Lawrence Berkeley National Laboratory

- **Full scan**
  - Straightforward parallel implementation.
  - Go over all elements and check against query condition.
  - Slow for single variable and simple query condition.
- **Data reorganization w/ sorting**
  - Requires data preparation, extra storage.
  - Eliminates the need to go through all elements.
  - Best performance for single variable query.
- **Bitmap index**
  - Requires index building in advance.
  - Go through index instead of data.
  - Best performance if actual values are not required.

Optimization?

BERKELEY LAB
Lawrence Berkeley National Laboratory

- **Full scan**
  - Skip the inspection of some amount of data?
- **Data reorganization**
  - Speedup the evaluation process for multivariate query conditions?
- **Index**
  - Skip the evaluation of some indexes?
  - Evaluate the highly selective variable first?

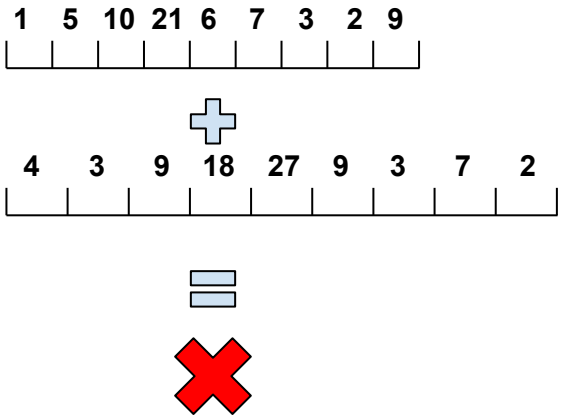
Histogram

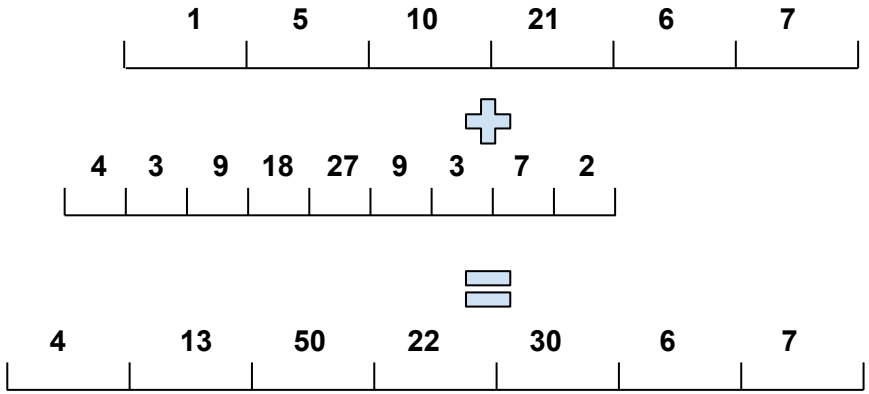
BERKELEY LAB
Lawrence Berkeley National Laboratory

- Generate a histogram for each PDC region
  - Done at data creation time or during server “free” time - *asynchronously*
- Use histogram to get max/min value of a region
- Use histogram to estimate the selectivity of each variable
  - Re-order the query evaluation, prune as many regions as possible.

- Generating a global one is costly, and needs coordination for updates.
  - Can we generate local region-specific histograms that can be easily merged into a global one?

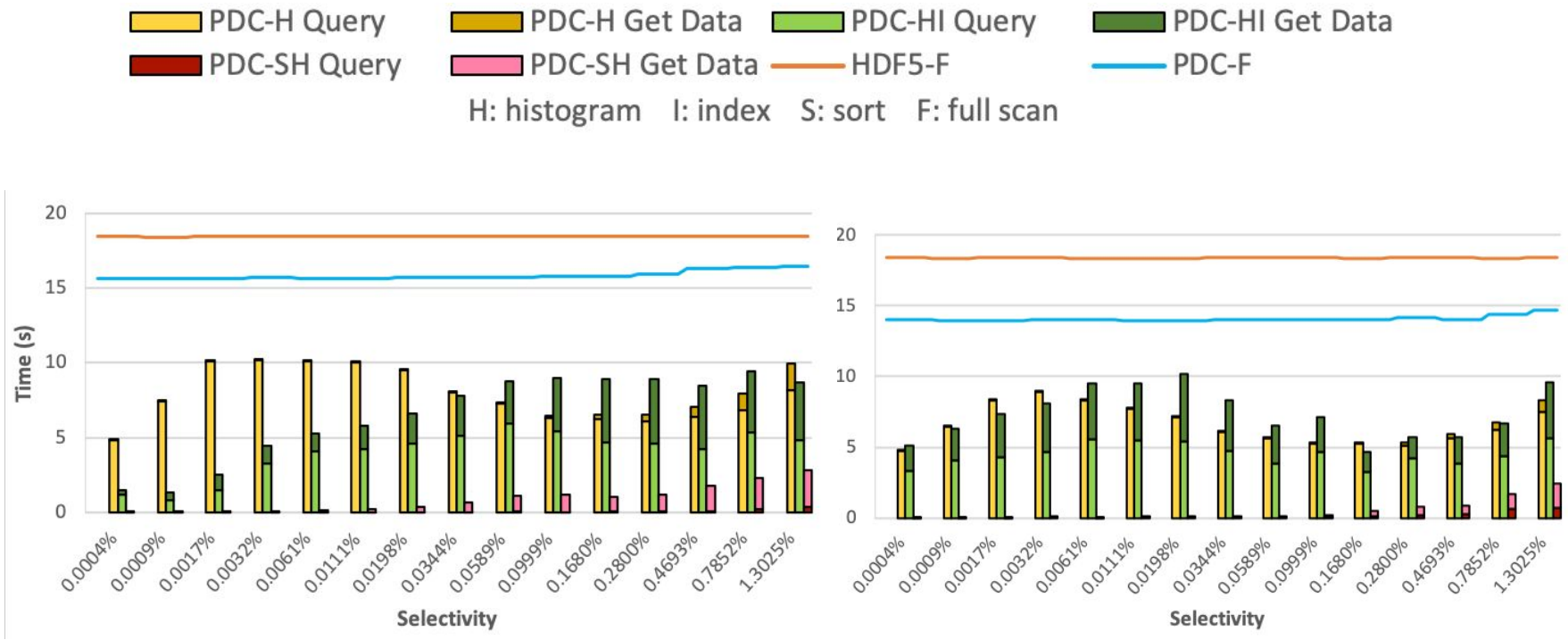
Mergeable Histogram





The bin width of different histograms must be same or divisible.
Use random sampling to get approximate min/max and make them aligned with bin boundaries of other histograms.
Both use values from pre-defined sets, 2^n and $N \pm 2^n$.

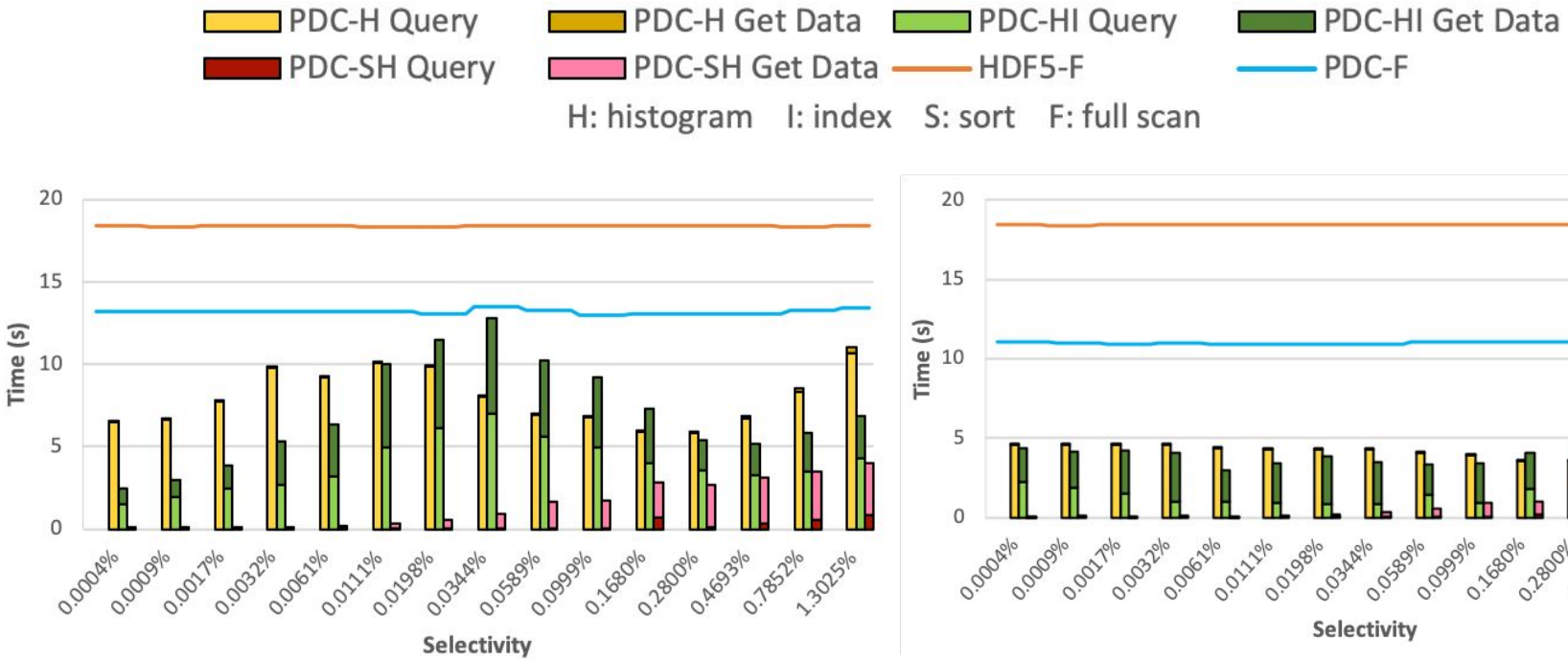
Region Size



4MB

8MB

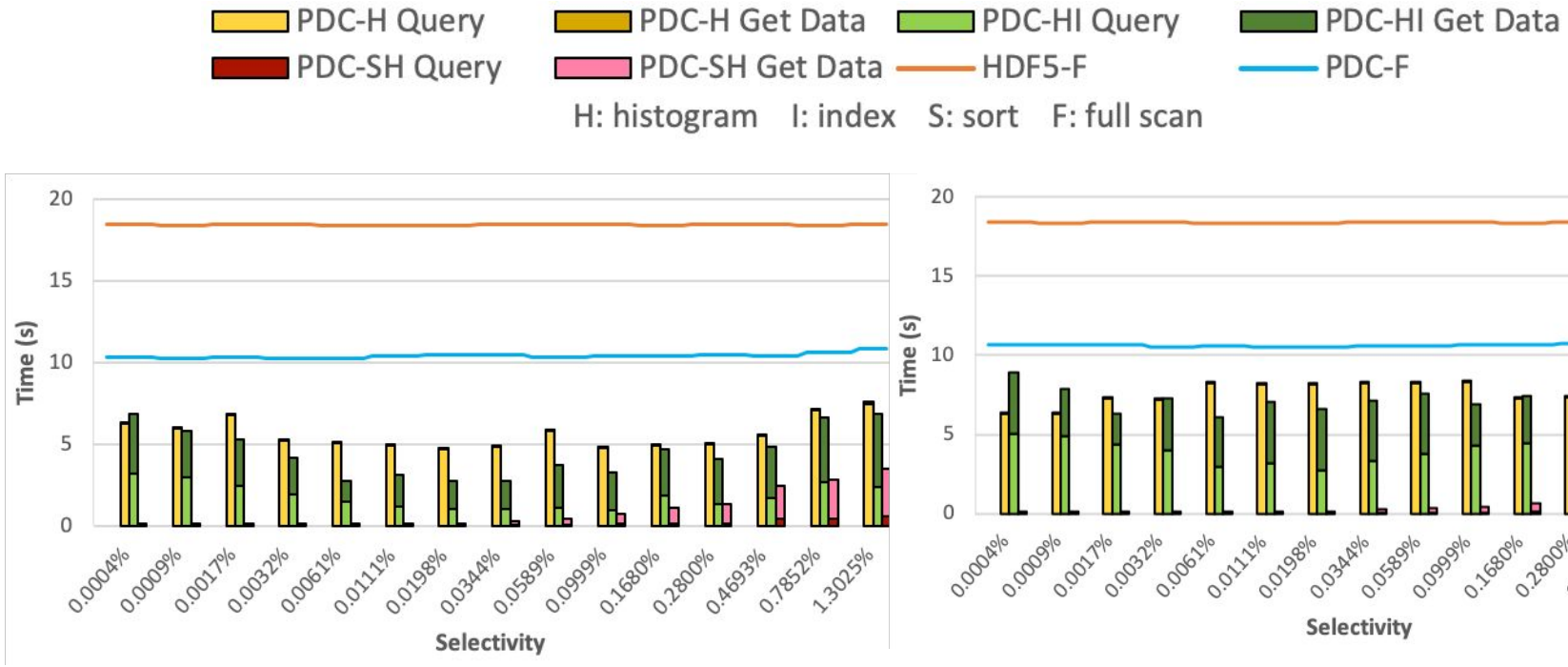
Region Size



16MB

32MB

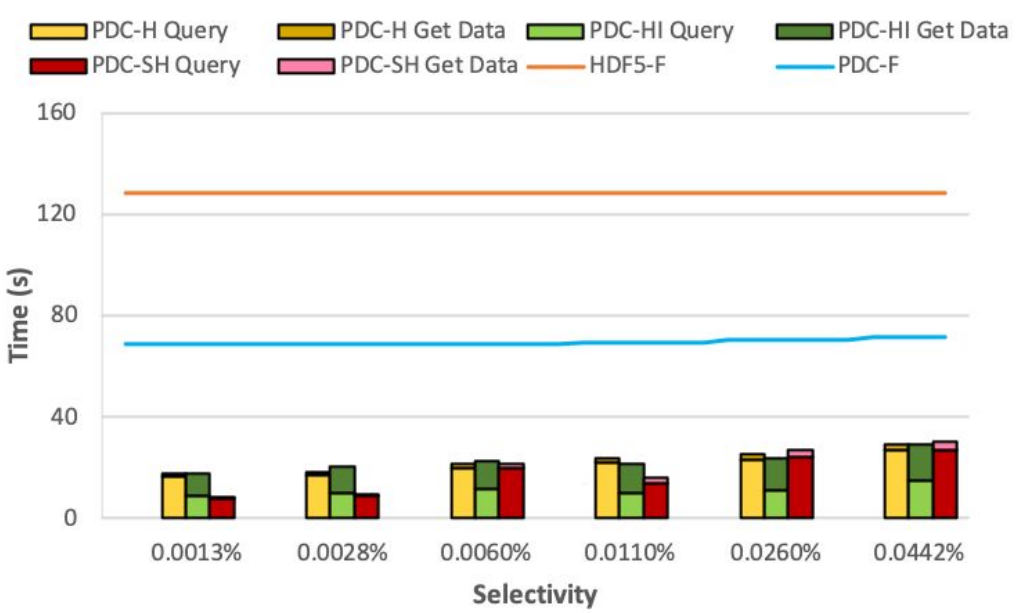
Region Size



64MB

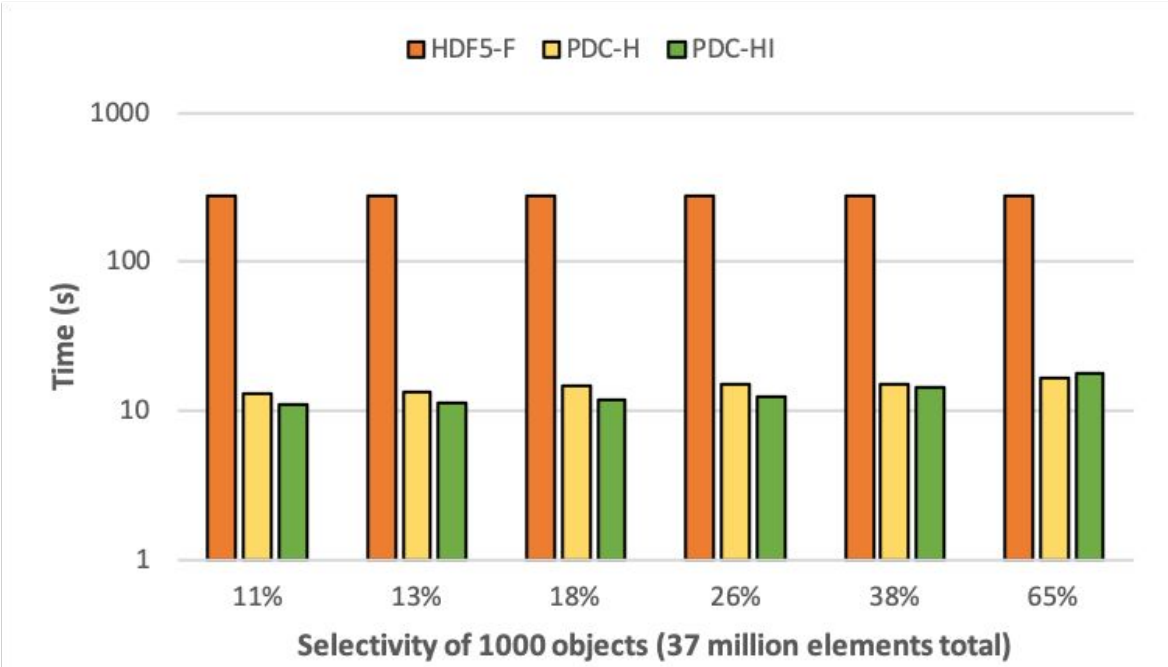
128MB

Results - Multivariate Query



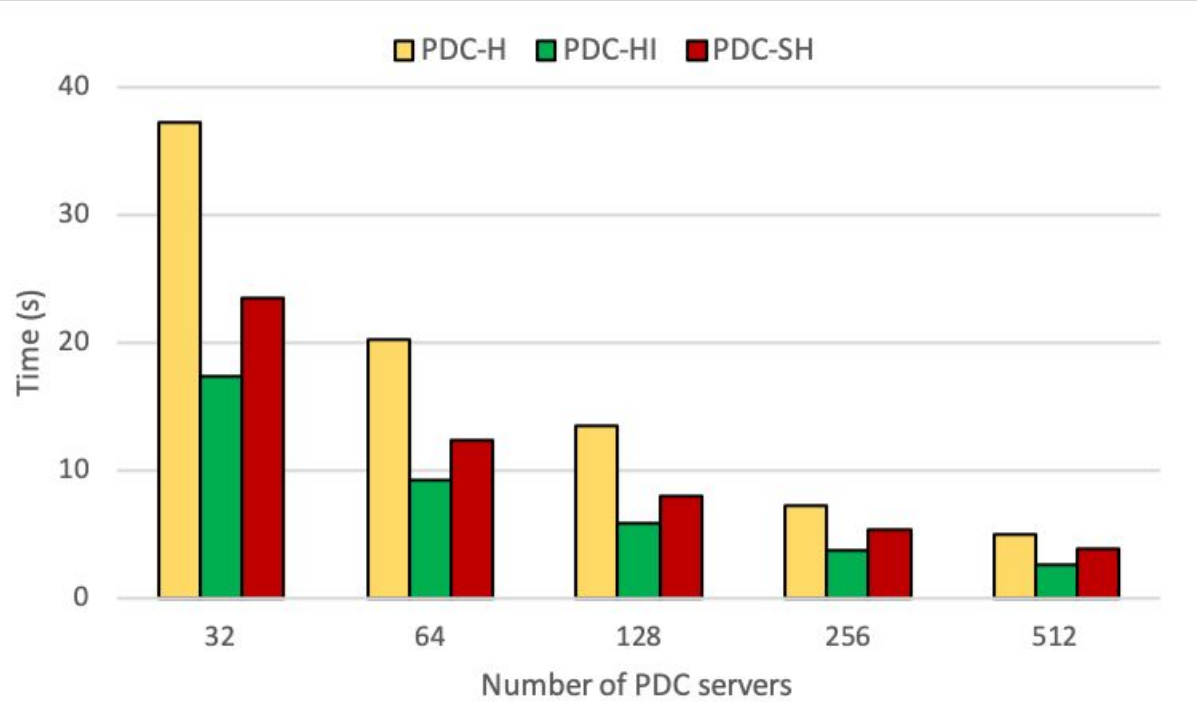
PDC-H: PDC with **H**istogram only, **PDC-HI**: PDC with **Hi**stogram and Fastbit **I**ndex, **PDC-SH**: PDC with **S**orted data (sorted by the 'energy' object) and **H**istogram. **HDF5-F**: amortized time to evaluate the 6 queries with HDF5 **F**ull scan. **PDC-F**: amortized time to evaluate the 6 queries with PDC **F**ull scan.

Results - Metadata + Data Queries



Comparison of queries with both metadata (fixed selectivity on 1000 objects) and data conditions (varied selectivity from 11% to 65%) on the H5BOSS dataset.

Results - Multivariate Scaling



Query time comparison for a multi-object query condition with 0:011% selectivity using different number of PDC servers.

Conclusion

BERKELEY LAB
Lawrence Berkeley National Laboratory

- Data querying is a crucial tool for **efficient information retrieval** that **enhances scientific productivity**
- **PDC-query provides a highly efficient and scalable query service**
  - Designed for object-centric data management systems with simple APIs
  - Novel optimizations using mergeable histograms on top of existing approaches such as data reorganization and indexing.
  - Single variable queries on sorted data have the best performance, index with histogram good if not retrieving values.
  - Multivariate queries with indexes or histograms have similar performance when data needs to be retrieved.

Thanks!

Questions?