Milestone 7.1: HDF5 Index Plugin API Demonstration

Ruth Aydt, Mohamad Chaarawi, Quincey Koziol,
Jerome Soumagne

03/24/2014

NOTICE: THIS MANUSCRIPT HAS BEEN AUTHORED BY HDF UNDER THE INTEL SUBCONTRACT WITH LAWRENCE LIVERMORE NATIONAL SECURITY, LLC WHO IS THE OPERATOR AND MANAGER OF LAWRENCE LIVERMORE NATIONAL LABORATORY UNDER CONTRACT NO. DE-AC52-07NA27344 WITH THE U.S. DEPARTMENT OF ENERGY. THE UNITED STATES GOVERNMENT RETAINS AND THE PUBLISHER, BY ACCEPTING THE ARTICLE OF PUBLICATION, ACKNOWLEDGES THAT THE UNITED STATES GOVERNMENT RETAINS A NON-EXCLUSIVE, PAID-UP, IRREVOCABLE, WORLD-WIDE LICENSE TO PUBLISH OR REPRODUCE THE PUBLISHED FORM OF THIS MANUSCRIPT, OR ALLOW OTHERS TO DO SO, FOR UNITED STATES GOVERNMENT PURPOSES. THE VIEWS AND OPINIONS OF AUTHORS EXPRESSED HEREIN DO NOT NECESSARILY REFLECT THOSE OF THE UNITED STATES GOVERNMENT OR LAWRENCE LIVERMORE NATIONAL SECURITY, LLC.

Background

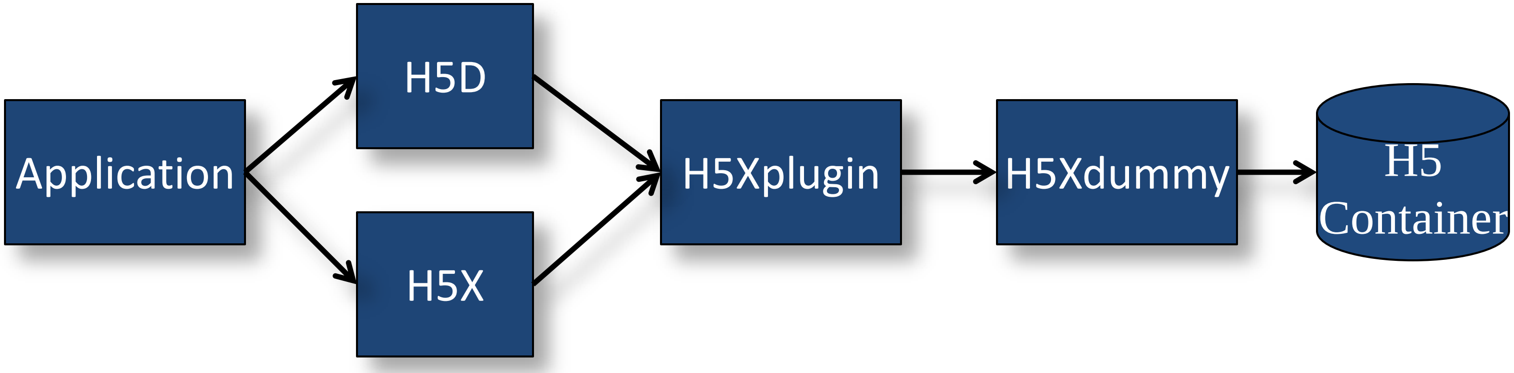
- Milestone 6.3 - Basic Analysis Shipping:
  - Queries sent to I/O node – for enumeration of the data that is present locally on each node
  - Queries operate on dataset elements
    - Restriction still valid for this quarter, but operations to be extended to attributes/links next quarter
  - Analysis performed on the requested data
- Problem
  - Time for applying query onto a dataset is O(n/p)
    - n is number of elements in the dataset
    - p is number of I/O nodes (assuming equal distribution of data)
  - Repeated for every query operation
- Solution
  - Add index to datasets

HDF5 Index Plugin API Extension

- Added indexing capabilities to H5Datasets, using a plugin mechanism
  - Similar to I/O filters available for chunked datasets
- Principal features:
  - Indexing packages may access H5Datasets within an H5Container more efficiently than current public HDF5 API
    - Still under development, any extensions will be designed and delivered in Q8, when third-party index plugins added
  - Indexing packages may create, write and read index information “anonymously” within the H5Container, storing it outside of group hierarchy
  - Applications may register and remove indexing plugins, query available indexing packages and choose which package(s) to use when creating indices

Indexing Demo

- What we demo
  - A simple and not-to-smart index plugin that verifies access to H5Dataset information within an H5Container
  - Index creation, access and updates within that container
  - An application using a chosen plugin to create an index for an H5Dataset, querying its associated index, and verification that the index is used to help fulfill the query inside the view



Index Plugin Class

```
typedef struct {
    unsigned version;     /* Version number of the index plugin class struct */
                          /* (Should always be set to H5X_CLASS_VERSION, which
                           *  may vary between releases of HDF5 library) */
    unsigned id;          /* Index ID (assigned by The HDF Group, for now) */
    const char *idx_name; /* Index name (for debugging only, currently) */
    H5X_type_t type;      /* Type of data indexed by this plugin */

    /* Callbacks */
    void *(*create)(hid_t file_id, hid_t dataset_id, hid_t xcpl_id,
        hid_t xapl_id, size_t *metadata_size, void **metadata);
    herr_t (*remove)(hid_t file_id, hid_t dataset_id, size_t metadata_size,
        void *metadata);

    void *(*open)(hid_t file_id, hid_t dataset_id, hid_t xapl_id,
        size_t metadata_size, void *metadata);
    herr_t (*close)(void *idx_handle);

    herr_t (*pre_update)(void *idx_handle, hid_t dataspace_id, hid_t xxpl_id);
    herr_t (*post_update)(void *idx_handle, const void *buf, hid_t dataspace_id,
            hid_t xxpl_id);
    herr_t (*query)(void *idx_handle, hid_t query_id, hid_t xxpl_id,
        hid_t *dataspace_id);
} H5X_class_t;
```

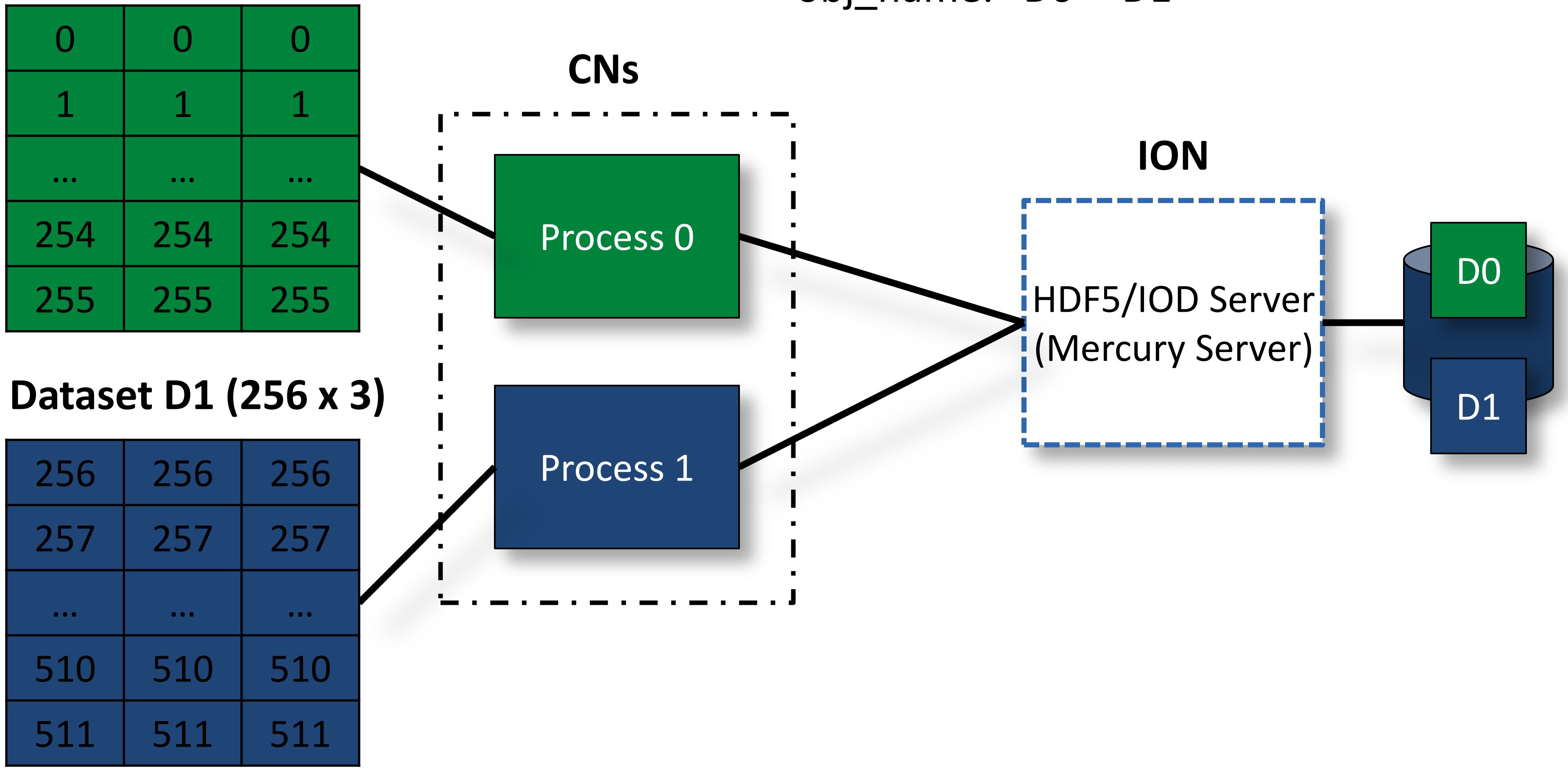
Index API

```
H5_DLL herr_t H5Xregister(const H5X_class_t *idx_class);
H5_DLL herr_t H5Xunregister(unsigned plugin_id);

H5_DLL herr_t H5Xcreate(hid_t file_id, unsigned plugin_id, hid_t scope_id,
        hid_t xcpl_id);
H5_DLL herr_t H5Xcreate_ff(hid_t file_id, unsigned plugin_id, hid_t scope_id,
        hid_t xcpl_id, hid_t trans_id, hid_t estack_id);

H5_DLL herr_t H5Xremove(hid_t file_id, unsigned plugin_id, hid_t scope_id);
H5_DLL herr_t H5Xremove_ff(hid_t file_id, unsigned plugin_id, hid_t scope_id,
        hid_t trans_id, hid_t estack_id);

H5_DLL herr_t H5Xget_count(hid_t scope_id, hsize_t *idx_count);
H5_DLL herr_t H5Xget_count_ff(hid_t scope_id, hsize_t *idx_count,
        hid_t rcxt_id, hid_t estack_id);
```

Demo example

Create datasets using H5Dcreate
filename: “eff_index_file.h5”
obj_name: “D0” “D1”

Dataset D0 (256 x 3)



0	0	0
1	1	1
…	…	…
254	254	254
255	255	255

Dataset D1 (256 x 3)

256	256	256
257	257	257
…	…	…
510	510	510
511	511	511

Demo example

```
Dataset Create D0 IOD ID
8070450532247928832, axe id 4
Operation 4 Dependencies:
Operation 4 will finish tasks 3 through 3
```

```
Start dataset create D0 at 131
Creating Dataset ID 7000000000000000 at (OH
131 ID 3000000000000000) with Data
integrity ENABLED
now creating the dataset with cellsize 4
num dimensions 2
PLIST Checksums key = 3EAC876B8026D430
value = D9312565A35FF37F
Link Count Checksums key = 011CCB7EF3D765ED
value = 932B19357AFA6A7F
Object Type Checksums key =
D1CF0DB19F2DBE8B  value = 473D440E2D0FA47C
Datatype Checksums key = 7820B38A4ED06624
value = 1C0A664818A3EC44
Dataspace Checksums key = A1CB6591D90DEC51
value = 70185FB121870545
Link Type Checksums key = 7B2FF7323D008A2A
value = B2AEF41EE7D316E5
Done with dset create, sending response to
client
```

Demo example

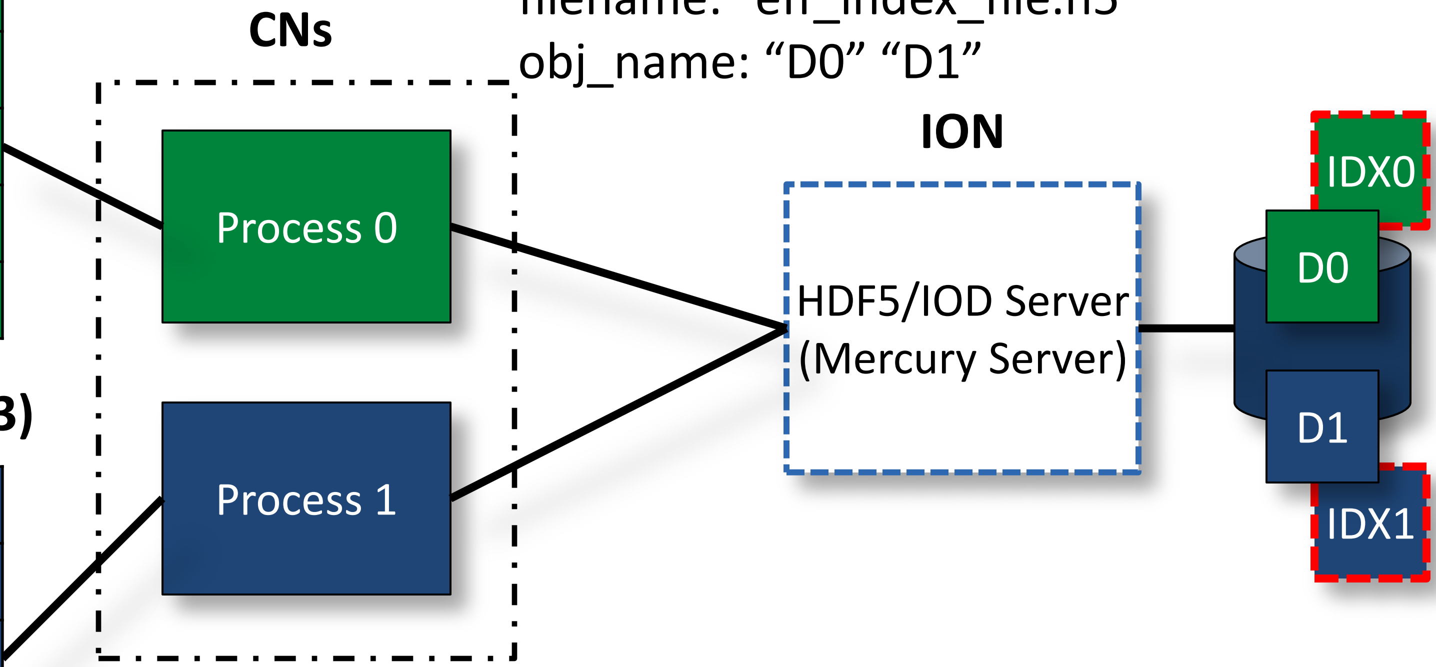
Create indices using H5Xcreate (call index create, create anonymous datasets for index information)
filename: “eff_index_file.h5”
obj_name: “D0” “D1”

Dataset D0 (256 x 3)

0	0	0
1	1	1
…	…	…
254	254	254
255	255	255

Dataset D1 (256 x 3)

256	256	256
257	257	257
…	…	…
510	510	510
511	511	511



***set_index_info** with index metadata created (stored on server using KVs for FF)*

Demo example

```
Calling H5X_dummy_create
Anon Dataset Create IOD ID
8070450532247928833, axe id 5
Operation 5 Dependencies:
Operation 5 will finish tasks 4 through 4
Index handle set to dataset
Set index info, axe id 6
Operation 6 Dependencies:
Operation 6 will finish tasks 5 through 5
Index info is now set
```

```
Start anon dataset create at 131
Creating Dataset ID 7000000000000001 at (OH
131 ID 3000000000000000) with Data
integrity ENABLED
now creating the dataset with cellsize 4
num dimensions 2
PLIST Checksums key = 3EAC876B8026D430
value = D9312565A35FF37F
Link Count Checksums key = 011CCB7EF3D765ED
value = 932B19357AFA6A7F
Object Type Checksums key =
D1CF0DB19F2DBE8B  value = 473D440E2D0FA47C
Datatype Checksums key = 7820B38A4ED06624
value = 1C0A664818A3EC44
Dataspace Checksums key = A1CB6591D90DEC51
value = 70185FB121870545
Done with dset create, sending response to
client
Start dataset set_index_info
Done with dataset set_index_info, sending
response to client
```

Demo example

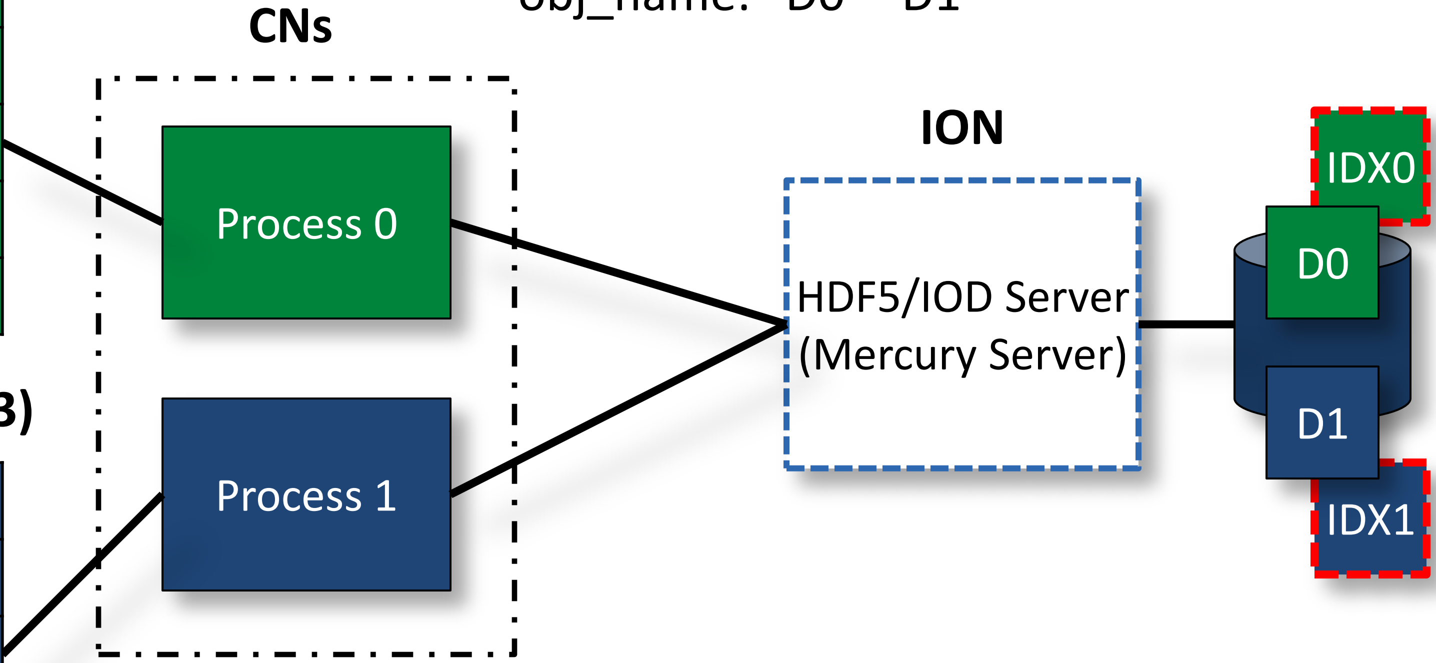
Write data using H5Dwrite
(call index pre_update and post_update)
filename: “eff_index_file.h5”
obj_name: “D0” “D1”

Dataset D0 (256 x 3)

0	0	0
1	1	1
…	…	…
254	254	254
255	255	255

Dataset D1 (256 x 3)

256	256	256
257	257	257
…	…	…
510	510	510
511	511	511



Demo example

```
Calling H5X_dummy_pre_update
Dataset Write, axe id 7
Operation 7 Dependencies:
Operation 7 will finish tasks 6 through 6
Calling H5X_dummy_post_update
Dataset Write, axe id 8
```

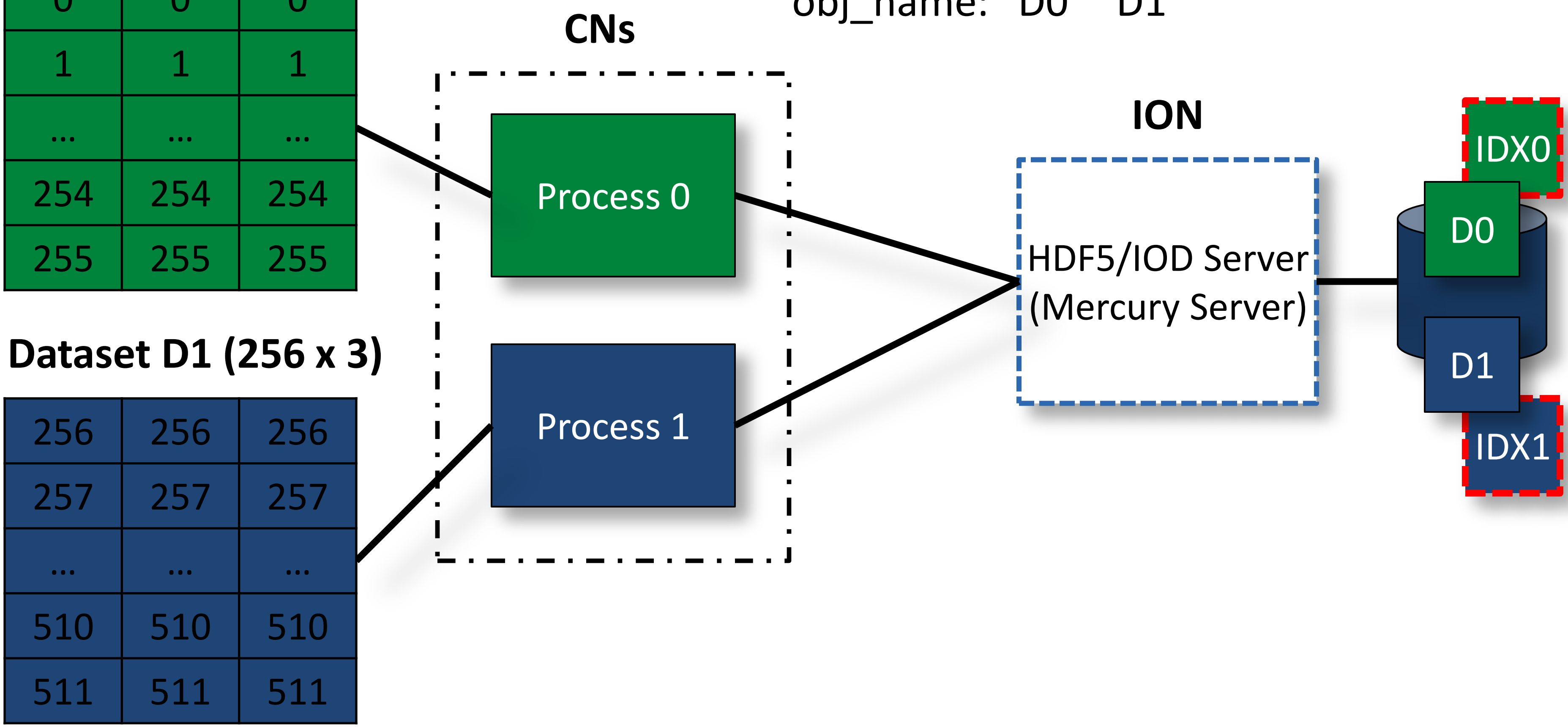
```
Start dataset Write on OH 211 OID
7000000000000000
Dim 0:  start 0   stride 256   block 256
count 1
Dim 1:  start 0   stride 3   block 3
count 1
Done with dset write, sending 0 response to
client
Start dataset Write on OH 235 OID
7000000000000001
Dim 0:  start 0   stride 256   block 256
count 1
Dim 1:  start 0   stride 3   block 3
count 1
Done with dset write, sending 0 response to
client
```

Demo example

Close dataset using H5Dclose (call index close)

filename: “eff_index_file.h5”

obj_name: “D0” “D1”



Dataset D0 (256 x 3)

0	0	0
1	1	1
…	…	…
254	254	254
255	255	255

Dataset D1 (256 x 3)

256	256	256
257	257	257
…	…	…
510	510	510
511	511	511

Demo example

```
Dataset Close IOD ID 8070450532247928832,
axe id 9
Operation 9 Dependencies:
Operation 9 will finish tasks 8 through 8
Calling H5X_dummy_close
Dataset Close IOD ID 8070450532247928833,
axe id 10
Operation 10 Dependencies:
Operation 10 will finish tasks 9 through 9
```

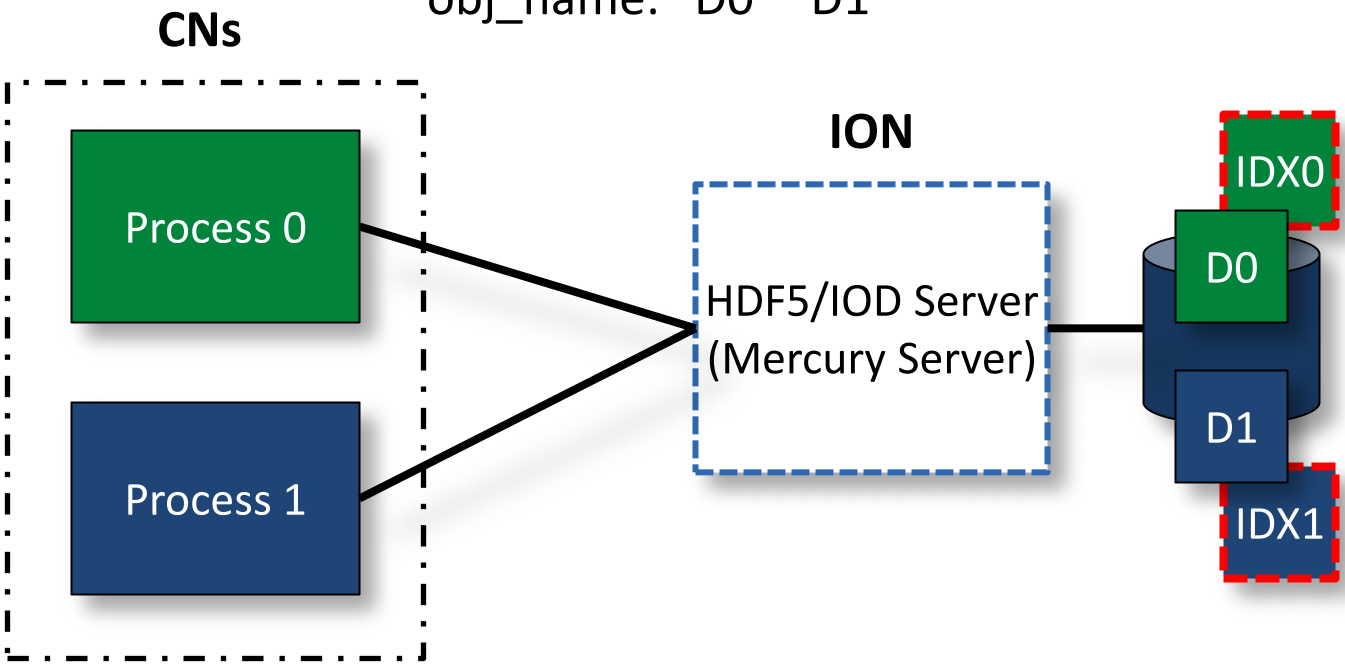
```
Start dataset Close 218 211
Done with dset close, sending response to
client
Start dataset Close 242 235
Done with dset close, sending response to
client
```

Demo example

Open dataset using H5Dopen (call index open)

filename: “eff_index_file.h5”

obj_name: “D0” “D1”



get_index_info *to retrieve index metadata and open index*

Demo example

```
Dataset Open D0 LOC ID 3458764513820540928,
axe id 14
Operation 14 Dependencies:
Operation 14 will finish tasks 13 through
13
Get index info, axe id 15
Operation 15 Dependencies:
Operation 15 will finish tasks 14 through
14
Get index info ret is: 0
Index count is: 1
Plugin ID is: 1
Calling H5X_dummy_open
Dataset open by token 16: ID
8070450532247928833
Operation 16 Dependencies:
Operation 16 will finish tasks 15 through
15
Index handle set to dataset
```

```
Start dataset open D0 at (OH 138 ID
3000000000000000)
…
Value CS iod = 1C0A664818A3EC44 computed =
1C0A664818A3EC44
Key CS iod = A1CB6591D90DEC51 computed =
A1CB6591D90DEC51
Value CS iod = 70185FB121870545 computed =
70185FB121870545
Done with dset open, sending response to
client
Start dataset get_index_info
Found index, index count is 1!
Get index info ret is: 0
Index count is: 1
Plugin ID is: 1
Done with dataset get_index_info, sending
response to client
Start Object Open by token =
7000000000000001
Done with object open by token, sending
response to client
```

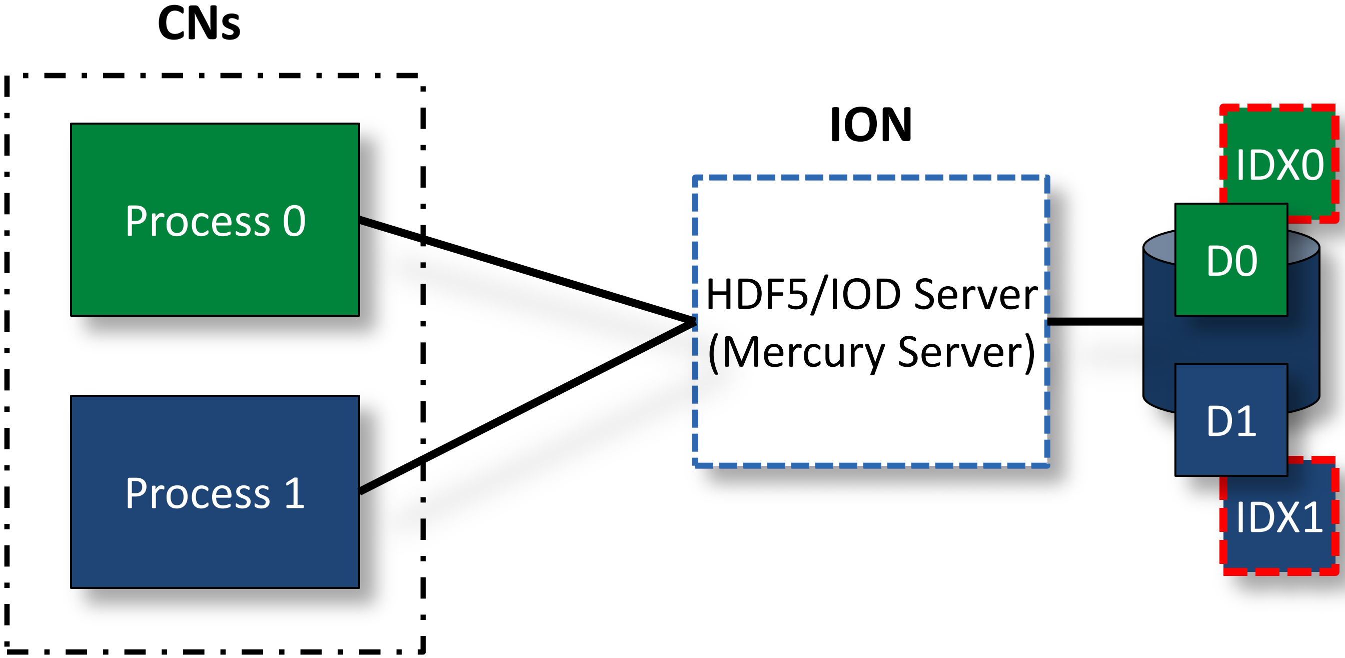
Demo example

Create a view from each process with H5Vcreate (call index query)

filename: “eff_index_file.h5” obj_name: “D0” “D1”

query: (39.1 < x < 42.1) OR (295 < x < 298)

```
Element |40|
matches query
…

Element |296|
matches query
…
```



Demo example

```
Calling H5X_dummy_query
Dataset Read, axe id 17
Operation 17 Dependencies:
Operation 17 will finish tasks 16 through
16
Element |40| matches query
Element |40| matches query
Element |40| matches query
Element |41| matches query
Element |41| matches query
Element |41| matches query
Element |42| matches query
Element |42| matches query
Element |42| matches query
Created dataspace from index with 9
elements
View Create at IOD ID 8070450532247928832,
axe id 18
```

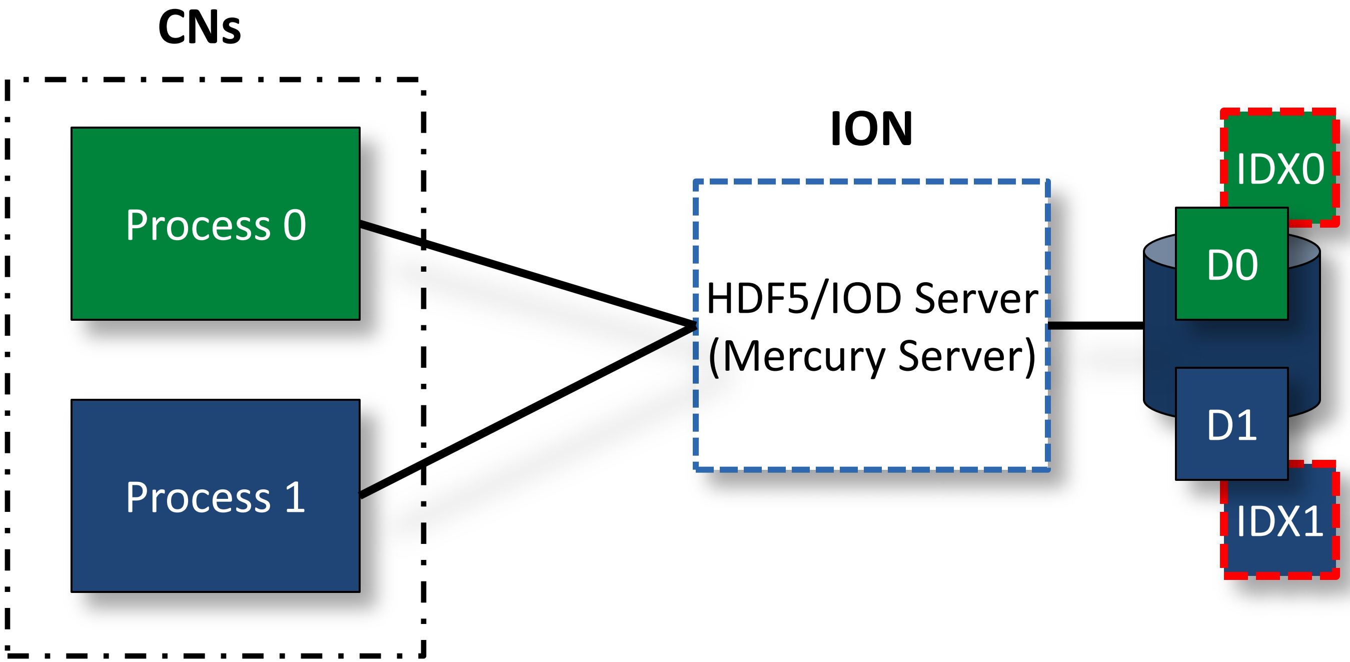
```
Start dataset Read on OH 378 OID
7000000000000001
Dim 0:  start 0   stride 256   block 256
count 1
Dim 1:  start 0   stride 3   block 3
count 1
IOD checksum  = 5FC96A7BCEF02F7D  Checksum
Computed = 5FC96A7BCEF02F7D
Done with dset read, checksum
5FC96A7BCEF02F7D, sending response to
client
```

Demo example

Close view and close datasets etc
(call index close)

filename: “eff_index_file.h5”

obj_name: “D0” “D1”



Demo example

```
Dataset Close IOD ID 8070450532247928832, axe id 18
Operation 18 Dependencies:
Operation 18 will finish tasks 17 through 17
Calling H5X_dummy_close
Dataset Close IOD ID 8070450532247928833, axe id 19
Operation 19 Dependencies:
Operation 19 will finish tasks 18 through 18
```

```
Start dataset Close 346 355
Done with dset close, sending response to client
Start dataset Close 378 387
Done with dset close, sending response to client
```

Conclusion – Q8 Tasks

- Indexing capabilities will be integrated into analysis shipping
- Two Third-Party Index Package Plugins:
  - Alacrity
  - FastBit
- Query/View/Index APIs extended to support Attributes/Link Names queries as well as dataset elements

Questions