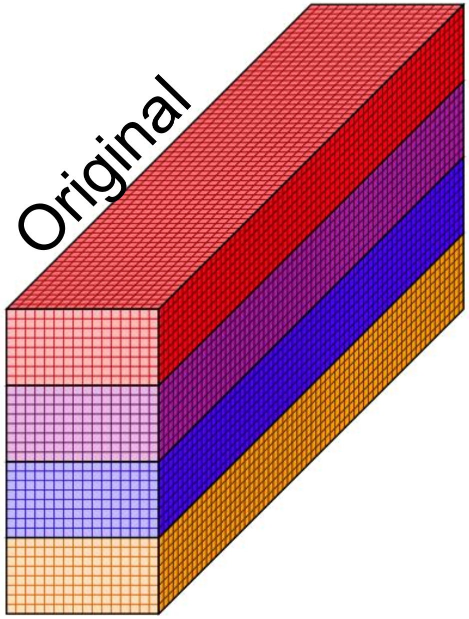
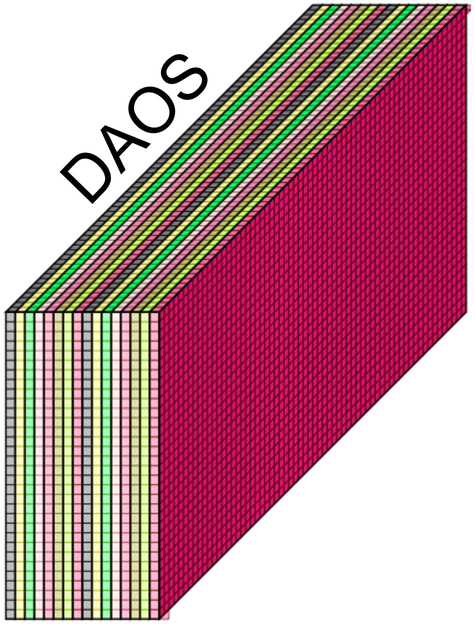
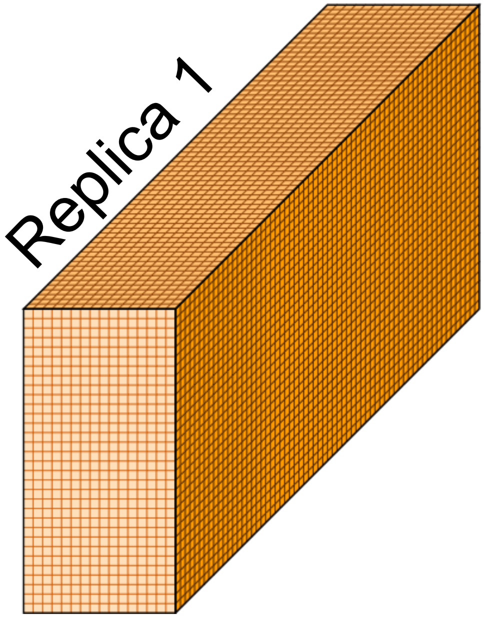


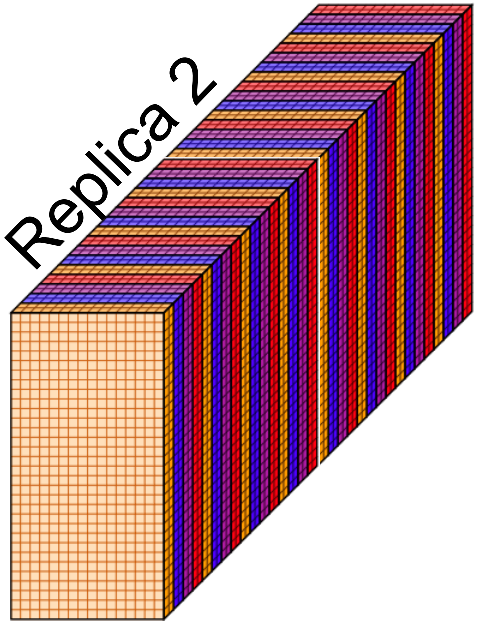
EFF-IO M7.5 Demo



Semantic Migration of Multi-dimensional Arrays

John Bent, Sorin Faibish, Xuezhao Liu, Harriet Qui, Haiying Tang, Jerry Tirrell, Jingwang Zhang, Kelly Zhang, Zhenhua Zhang



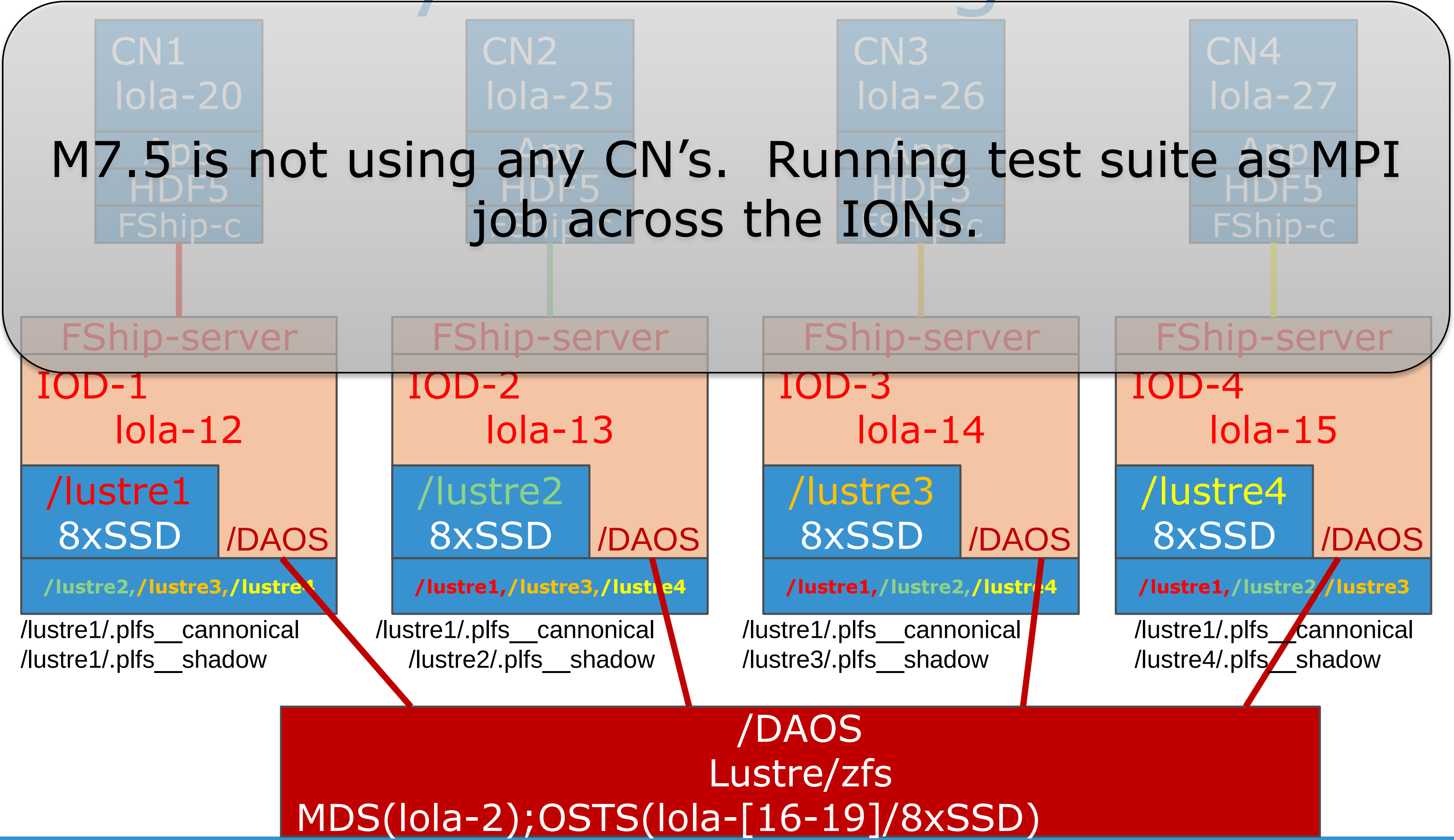


NOTICE: THIS MANUSCRIPT HAS BEEN AUTHORED BY EMC UNDER INTEL'S SUBCONTRACT WITH LAWRENCE LIVERMORE NATIONAL SECURITY, LLC WHO IS THE OPERATOR AND MANAGER OF LAWRENCE LIVERMORE NATIONAL LABORATORY UNDER CONTRACT NO. DE-AC52-07NA27344 WITH THE U.S. DEPARTMENT OF ENERGY. THE UNITED STATES GOVERNMENT RETAINS 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.

Demo System Configuration

- IONS: Four IONs
  - lola-[12-15].hpdd.intel.com
  - 8 SSD burst buffers on each
- DAOS: Global /mnt/daos mount on each
  - DAOS-lustre
  - MDS(lola-2);OSTS(lola-[16-19]/8xSSD)
- BB's: Local /mnt/lustre[1-4] on each ION
  - All IONs have lustre cross-mounts to all BB's
- CN's: Four CNs available via Mercury
  - lola-[20,25-27]
  - Our demo runs directly on ION's though without HDF
    - MPI job making calls directly to IOD API

Demo System Configuration



M7.5 is not using any CN's. Running test suite as MPI job across the IONs.

Demo scope

- Semantic migration of arrays
  - Persist (IOD -> DAOS)
  - Fetch (DAOS -> IOD)
- Dimensional sequence transformation
  - Changing the access dimension
- Reading from a persisted / purge array
  - i.e. Reading directly from DAOS
- Query layout / sharding

Initialization 1/2

```
Script started on Fri 21 Mar 2014 09:21:30 AM PDT
jim.pedone@lola-3:/scratch/iod/M7.5_demo
root@lola-3 M7.5_demo]# ./M7.5_demo1.sh
Cleaning the BB's and DAOS to return to initial state.

M7.5 demonstration 1 -- Array Migration.
will run M7.5_case1_array_migration_ops on 4 IODS: lola-12 lola-13 lola-14 lola-15
> mpirun -np 4 -hosts lola-12,lola-13,lola-14,lola-15 -c demo_config -t
M7.5_case1_array_migration_ops
Fri Mar 21 09:21:41 PDT 2014

*********  Test Configuration   *********
           inputFile     = M7.5_case1_array_migration_ops
           nproc         =                  4
           all_async     =                 True
```

Initialization 2/2

```
========================================================================
# Demonstration: Array Migration, Data Reorganization and Layout Discovery
#
# This demonstration will cover these elements:
# Migration of array objects using structural descriptions between DAOS and IODs
# - Will be shown in both directions
# Reorganization of data layout during migration for array objects
# - E.g. Migrate using a layout which specifies a different dimension sequence
# Layout query of array objects
# - E.g. Returning structure set to place data when migrating to DAOS
#
# Out of scope demonstration elements:
# - Similar functionality for KV objects and blob objects will not be
# demonstrated at this time
# ========================================================================
```

Step 1: Create and open container

```
Running command: * contopen /containerA create
Call iod_container_open(), path: /containerA, open_mode: 0x83...
Finish to open container - passed (rc = 0)
```

Step 2.1 – 2.2: Skip Transaction 0

```
Running command: 0 transstart 0 write
Call iod_trans_start(), tid: 0x0, num_ranks: 0, mode: 0x2...
Finish to start transaction - passed (rc = 0)
# ######## Step 2.2 - finish TID0 for write
# Command format: tid transfinish flag
# flag ---------- only meaningful for writing transaction. 'normal' for commonly finish
> Press enter to continue...

Running command: 0 transfinish normal
Call iod_trans_finish(), tid: 0x0, num_ranks: 0, flag: 0
Finish to finish transaction - passed (rc = 0)
```

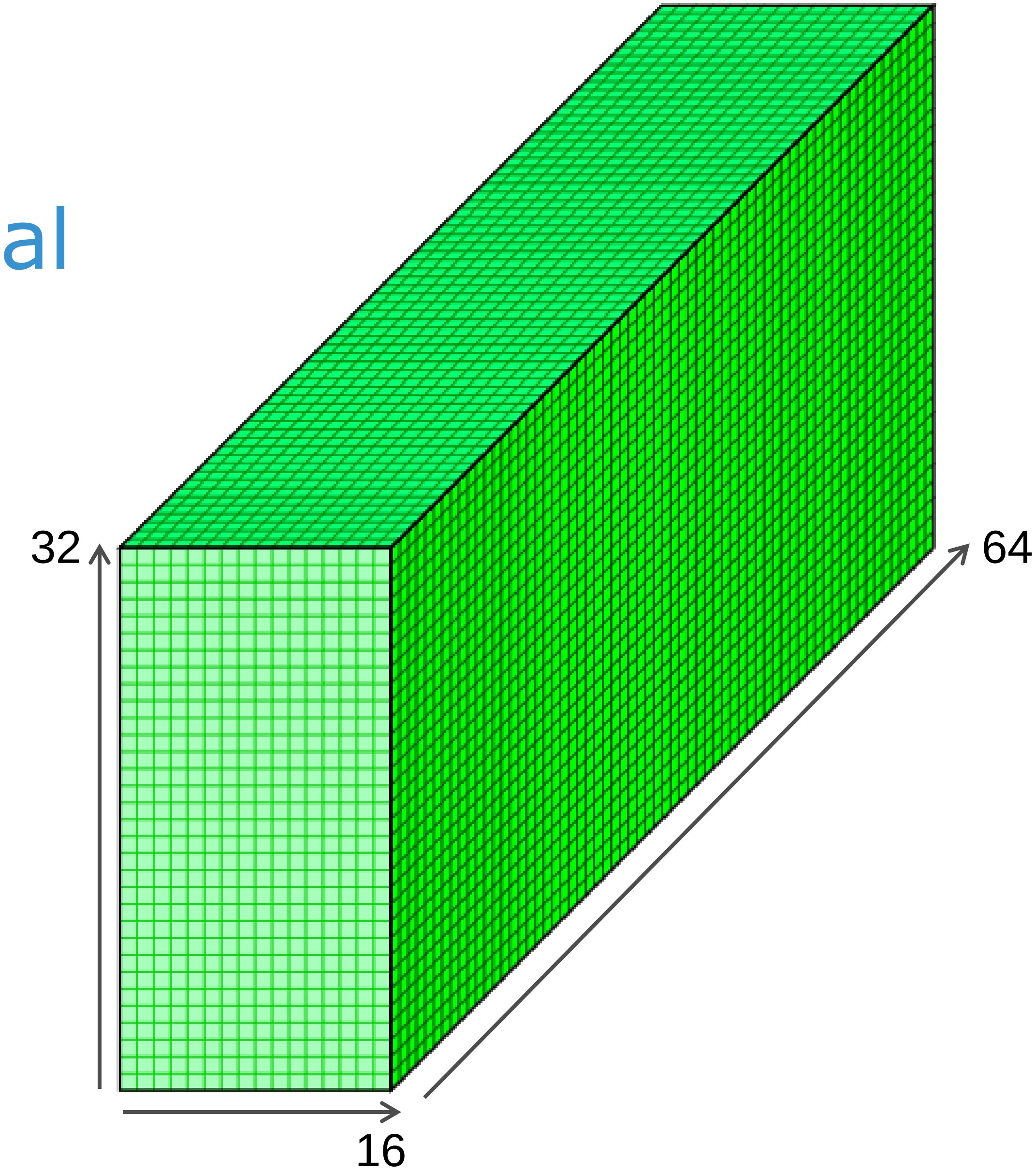
Step 2.3 – 2.4: Start TID=1, create array

Leader mode

```
Running command: 1 transstart 0 write
Call iod_trans_start(), tid: 0x1, num_ranks: 0, mode: 0x2...
Finish to start transaction - passed (rc = 0)

# ######## Step 2.4 - in TID1, create ARRAY object:
            cell_size=32 num_dims=3 (16 X 32 X 64)
> Press enter to continue...

Running command: 1 create array array_obj 32 3 16,32,64 16
Call iod_obj_create(), tid: 0x1, type: 0x3...
Array struct: cell_size: 512, num_dims:  3, firstdim_max: 16, current_dims: 16,32,64
Finish to create object - passed (rc = 0)
```

16MB array
(512x16x32x64)

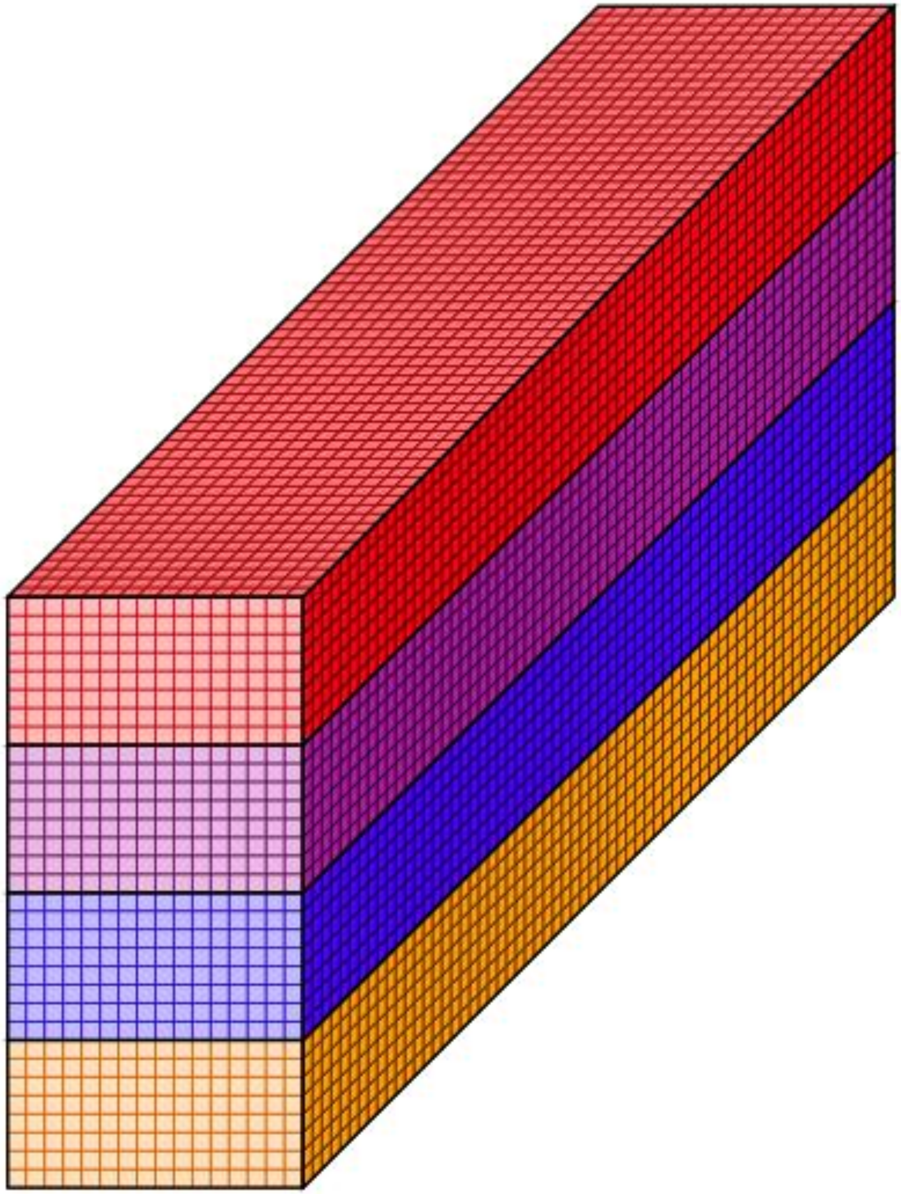
16 x 32 x 64 Multi-dimensional Array



Step 2.5: Write into array

```
# Call iod_array_write() to write the ARRAY object
# start is [0,0,0], stride is [8,8,8], count is [2,4,8], block is [8,8,8]
#
> Press enter to continue...

Running command: 1 write array_obj 0,0,0 8,8,8 2,4,8 8,8,8
Rank 0 ID 0: start[ 0,0,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 1 ID 0: start[ 0,8,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 2 ID 0: start[ 0,16,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 3 ID 0: start[ 0,24,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Call iod_array_write(), oh: 123, tid: 0x1
Rank 0 ID 0: iod_array_write() returns 0
Rank 1 ID 0: iod_array_write() returns 0
Rank 3 ID 0: iod_array_write() returns 0
Rank 2 ID 0: iod_array_write() returns 0
Finish to write array - passed (rc = 0)
```

Write Pattern



Rank 3

Rank 2

Rank 1

Rank 0

Step 2.6: Prepare semantic persist

```
# location ------ "DAOS" (i.e. IOD_LOC_CENTRAL)
# type ---------- currently only support stripped layout (i.e.IOD_LAYOUT_STRIPED).
# target_start -- the offset in DAOS shard list, currently IOD ignores this value
                  (chooses one internally for load balance).
# target_num ---- number of storage targets (i.e. DAOS shards). If 0, it defaults to all shards.
# stripe_size --- resharding granularity. The number of cells for contiguous layout array.
# dims_seq ------ For contigous layout, it is an array with num_dims size (i.e. a 4D array may have
# [3][2][1][0] as its logical dimension sequence)
Running command: 1 setlayout array_obj DAOS,striped,0,2,128,<1,2,0>
Call iod_obj_set_layout(), tid: 0x1
Layout to set:
location   type      target_start  target_num  stripe_size   dims_seq
DAOS       striped     1             8           2048          1, 2, 0
Finish to set object layout - passed (rc = 0)
```

Step 2.7: Commit TID=1

```
Running command: 1 transfinish normal
Call iod_trans_finish(), tid: 0x1, num_ranks: 0, flag: 0
Finish to finish transaction - passed (rc = 0)
```

Step 3.2: Query Eventual Persist Layout

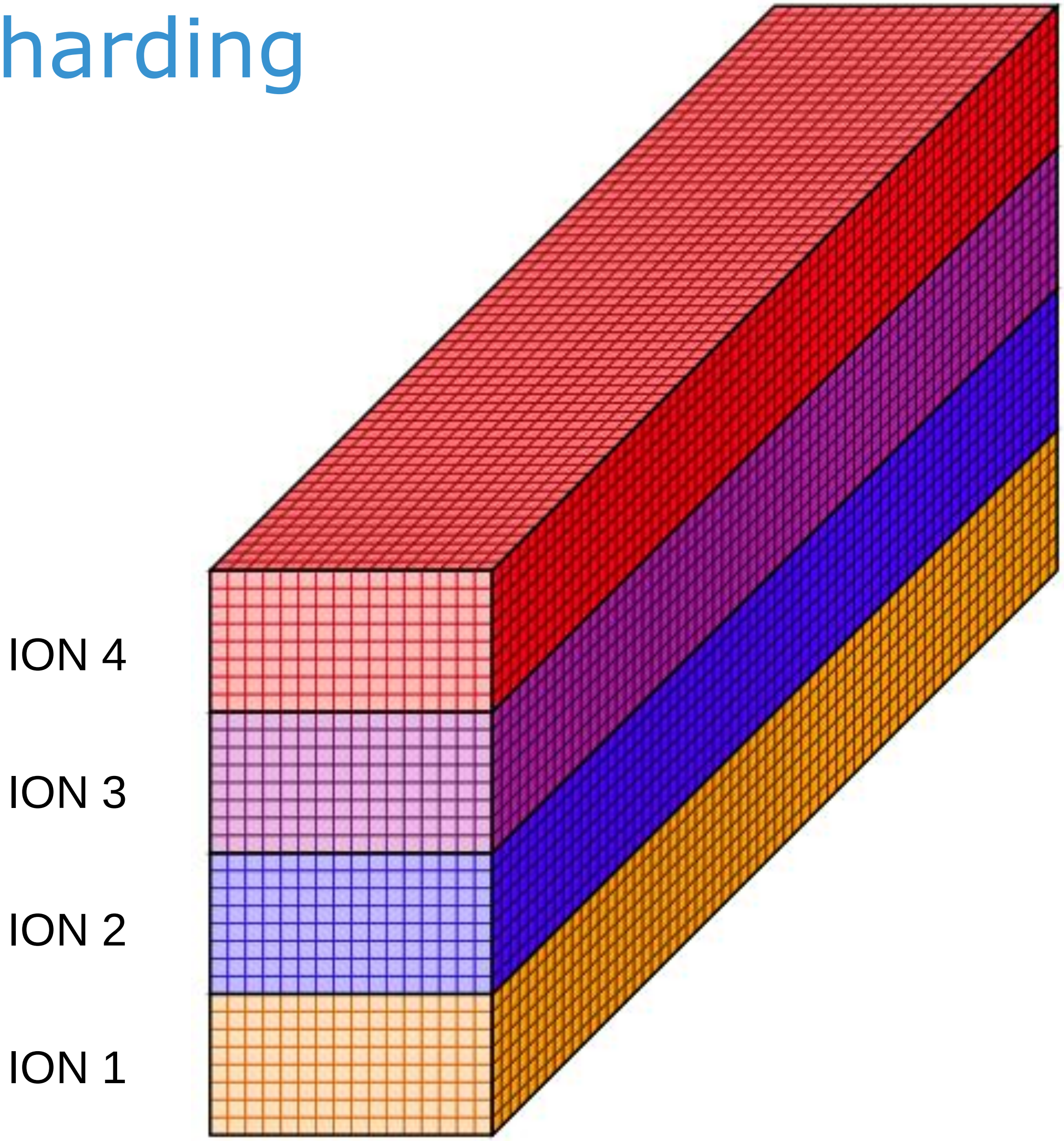
```
Running command: 1 getlayout array_obj
At tid 0x1: dump IOD returned layout:
location    type        target_start  target_num  stripe_size   dims_seq
DAOS        striped     1             8           2048          1, 2, 0
Rank 0 ID 0: get_layout verification succeed
Finish to get object layout - passed (rc = 0)
```

Step 3.3: Query Semantic Shards on IONs

```
Running command: 1 querymap array_obj
Call iod_obj_query_map(), tid: 0x1
Rank 0 in tid 0x1: dump IOD returned map:

obj_id            type      n_range
6000000171b6056d  ARRAY     64

range:  0, start:  0,  0,  0,  end:  0,  7, 63,  n_cell:   512, loc: /mnt/lustre1/.plfs__shadow
range:  1, start:  0,  8,  0,  end:  0, 15, 63,  n_cell:   512, loc: /mnt/lustre2/.plfs__shadow
range:  2, start:  0, 16,  0,  end:  0, 23, 63,  n_cell:   512, loc: /mnt/lustre3/.plfs__shadow
range:  3, start:  0, 24,  0,  end:  0, 31, 63,  n_cell:   512, loc: /mnt/lustre4/.plfs__shadow
range:  4, start:  1,  0,  0,  end:  1,  7, 63,  n_cell:   512, loc: /mnt/lustre1/.plfs__shadow
range:  5, start:  1,  8,  0,  end:  1, 15, 63,  n_cell:   512, loc: /mnt/lustre2/.plfs__shadow
range:  6, start:  1, 16,  0,  end:  1, 23, 63,  n_cell:   512, loc: /mnt/lustre3/.plfs__shadow
range:  7, start:  1, 24,  0,  end:  1, 31, 63,  n_cell:   512, loc: /mnt/lustre4/.plfs__shadow
<snip>
range: 56, start: 14,  0,  0,  end: 14,  7, 63,  n_cell:   512, loc: /mnt/lustre1/.plfs__shadow
range: 57, start: 14,  8,  0,  end: 14, 15, 63,  n_cell:   512, loc: /mnt/lustre2/.plfs__shadow
range: 58, start: 14, 16,  0,  end: 14, 23, 63,  n_cell:   512, loc: /mnt/lustre3/.plfs__shadow
range: 59, start: 14, 24,  0,  end: 14, 31, 63,  n_cell:   512, loc: /mnt/lustre4/.plfs__shadow
range: 60, start: 15,  0,  0,  end: 15,  7, 63,  n_cell:   512, loc: /mnt/lustre1/.plfs__shadow
range: 61, start: 15,  8,  0,  end: 15, 15, 63,  n_cell:   512, loc: /mnt/lustre2/.plfs__shadow
range: 62, start: 15, 16,  0,  end: 15, 23, 63,  n_cell:   512, loc: /mnt/lustre3/.plfs__shadow
range: 63, start: 15, 24,  0,  end: 15, 31, 63,  n_cell:   512, loc: /mnt/lustre4/.plfs__shadow
```

Result of write pattern

Semantic Sharding on IONs



ION 4

ION 3

ION 2

ION 1

Step 3.3: Query Semantic Shards on IONs

```
# Verify data in BB’s:
[root@lola-3 scripts]# plfs_query /tmp/iod_plfs/containerA/6000000110246979/1 \
            | grep dropping.data | xargs du -h
4.0M
           /mnt/lustre7/.plfs__shadow///containerA/6000000110246979/1/hostdir.1/droppin
g.data.1395627045.428540.lola-7.lola.whamcloud.com.114393
4.0M
           /mnt/lustre6/.plfs__shadow///containerA/6000000110246979/1/hostdir.2/droppin
g.data.1395627006.948952.lola-6.lola.whamcloud.com.112589
4.0M
           /mnt/lustre5/.plfs__shadow///containerA/6000000110246979/1/hostdir.3/droppin
g.data.1395626999.206627.lola-5.lola.whamcloud.com.57576
4.0M
           /mnt/lustre3/.plfs__shadow///containerA/6000000110246979/1/hostdir.5/droppin
g.data.1395626997.366273.lola-3.lola.whamcloud.com.109024
```

Step 4.1: Persist container at TID=1

```
Running command: 1 persist
Call iod_trans_persist(), coh: 25, tid : 0x1...
Finish to persist container - passed (rc = 0)
```

```
# Verify proper scatter-gather migration:
> [root@lola-3 scripts]# egrep 'reqmsg|ACK' /tmp/iod.debug.log | head
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO handling reqmsg, type: 1, msg_len: 63.(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO ENTER: _container_ops_trans_start_reqmsg_handler (rank 0)(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO EXIT : _container_ops_trans_start_reqmsg_handler, rc: 0.(rank 0)(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO handling reqmsg, type: 1, msg_len: 63.(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO ENTER: _container_ops_trans_start_reqmsg_handler (rank 0)(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO EXIT : _container_ops_trans_start_reqmsg_handler, rc: 0.(rank 0)(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO sending ACK req_id 2, result: 0, piggyback: ffffffffffffffff from 0 to 0 ... (rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO handling reqmsg, type: 2, msg_len: 32.(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO ACK arrived, from IOD rank 0, req_id: 2, result: 0.(rank 0)
2014/03/23-16:52:33.07 lola-3 plfs[107367] IOD  INFO ENTER: iod_reqmsg_ack_handler (rank 0)(rank 0)
```

Step 4.1: Persist container at TID=1

```
# Verify data on DAOS:
> [root@lola-3 scripts]# ./daos_query.sh /mnt/daos/containerA
[root@lola-3 scripts]# daos_query.sh /mnt/daos/containerA
/scratch/iod/tests/output/daos_query.sh/Mar.23.2014.19.56
### ls -l /mnt/daos/containerA
-rw-r--r-- 1 root root 4294967296 Mar 23 19:55 /mnt/daos/containerA
### du -h /mnt/daos/containerA
27M          /mnt/daos/containerA
#### daos_ctl run -c /mnt/daos/containerA Cor,Cq,Eq,Cc: 1
HCE is 1
#### daos_ctl run -c /mnt/daos/containerA Cor,Cq,Eq,Cc: 8
8 shards
#### daos_ctl run -c /mnt/daos/containerA Cor,Sq1:0,Sl1:0,Cc: 141,
Shard 0 has 141, objects and uses 16244736 (15 MB)
Objects in shard 0: [0:2377900603251621889] [0:2377900603251621890] [0:2377900603251621891]
[0:2377900603251621892] [0:6917529032887005706]
#### daos_ctl run -c /mnt/daos/containerA Cor,Sq1:1,Sl1:1,Cc: 5480448
Shard 1 has 135, objects and uses 5480448 (5 MB)
Objects in shard 1: [1:6917529032887005706]
<snip>
#### daos_ctl run -c /mnt/daos/containerA Cor,Sq1:7,Sl1:7,Cc: 5480448
Shard 7 has 135, objects and uses 5480448 (5 MB)
Objects in shard 7: [7:6917529032887005706]
```

Semantic Sharding on DAOS

KEY

Shard0

Shard1

Shard2

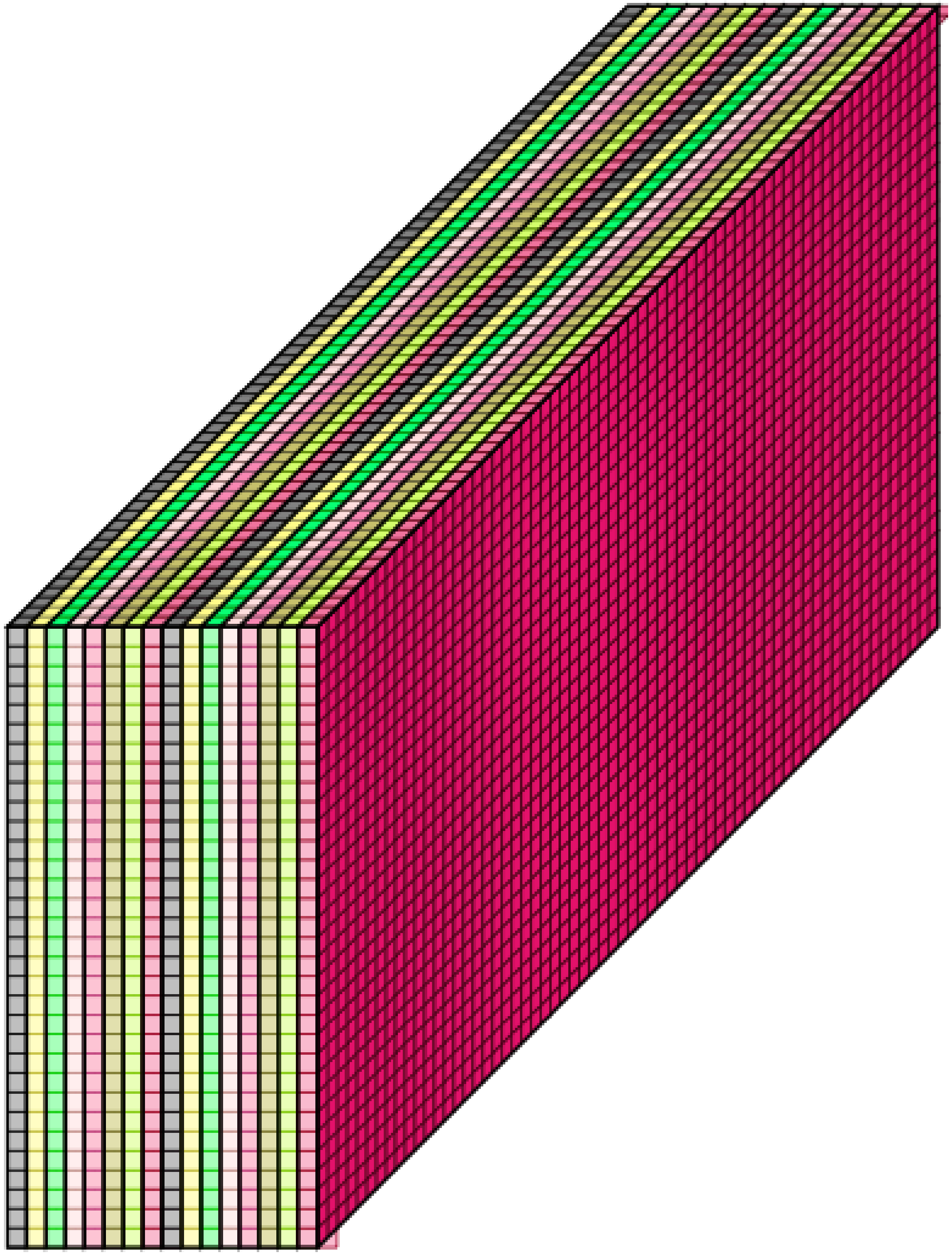
Shard3

Shard4

Shard5

Shard6

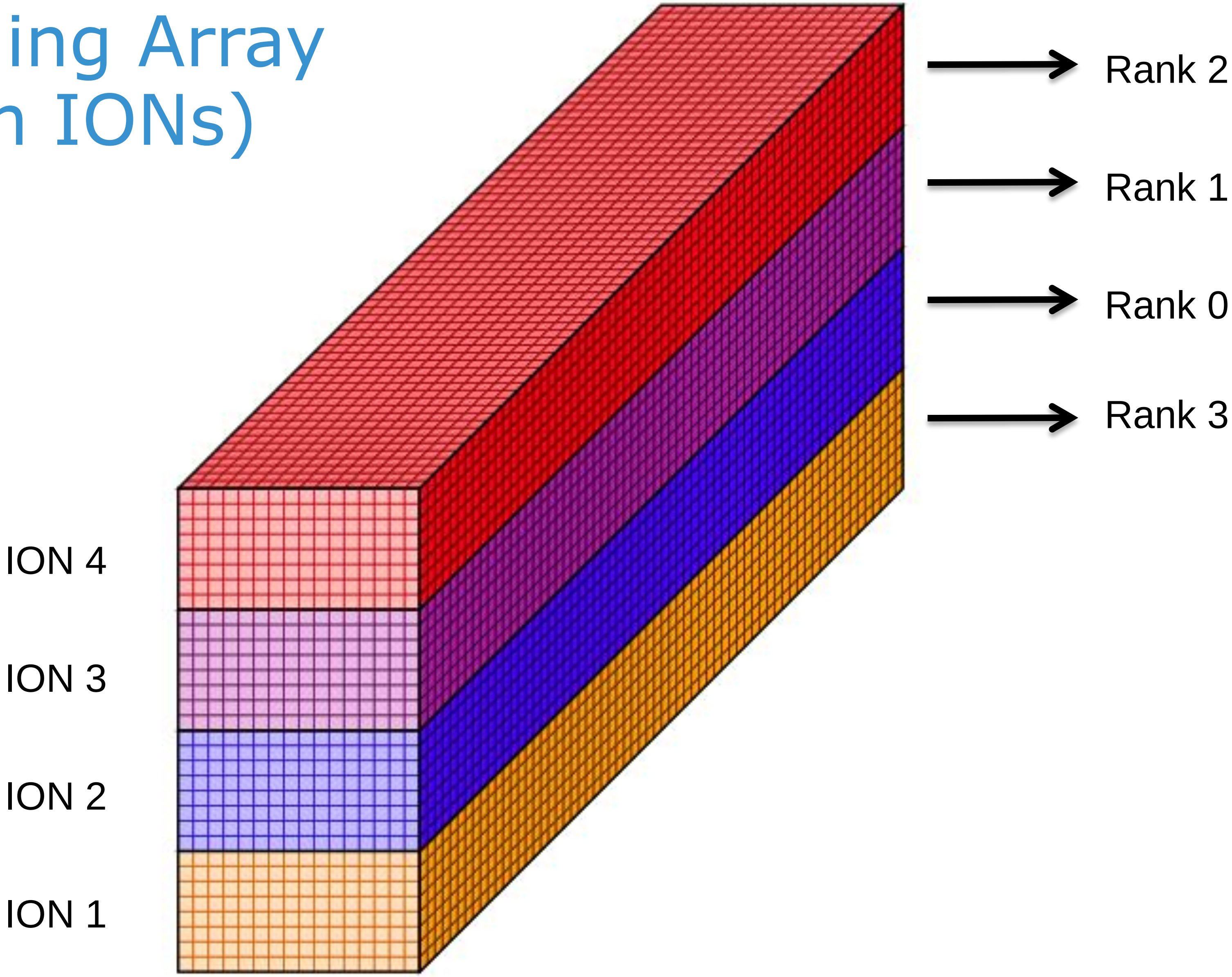
Shard7



Step 4.2: Read array (from IONs)

```
Running command: 1 read array_obj 0,0,0 8,8,8 2,4,8 8,8,8
Rank 0 ID 0: start[ 0,8,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 1 ID 0: start[ 0,16,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 2 ID 0: start[ 0,24,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 3 ID 0: start[ 0,0,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Call iod_array_read(), oh: 234, tid: 0x1
Rank 2 ID 0: iod_array_read() returns 0
Rank 1 ID 0: iod_array_read() returns 0
Rank 0 ID 0: iod_array_read() returns 0
Finish to read array - passed (rc = 0)
Rank 3 ID 0: iod_array_read() returns 0
```

Reading Array (from IONs)



Step 4.3/4.4: Purge Array / Query map

```
Running command: 1 purge array_obj
Call iod_obj_purge(), oh: 242, tid: 0x1...
Finish to purge object - passed (rc = 0)

Running command: 1 querymap array_obj
Call iod_obj_query_map(), tid: 0x1
Rank 0 in tid 0x1: dump IOD returned map:
obj_id            type      n_range
6000000171b6056d  ARRAY     0
Finish to query object map - passed (rc = 0)
```

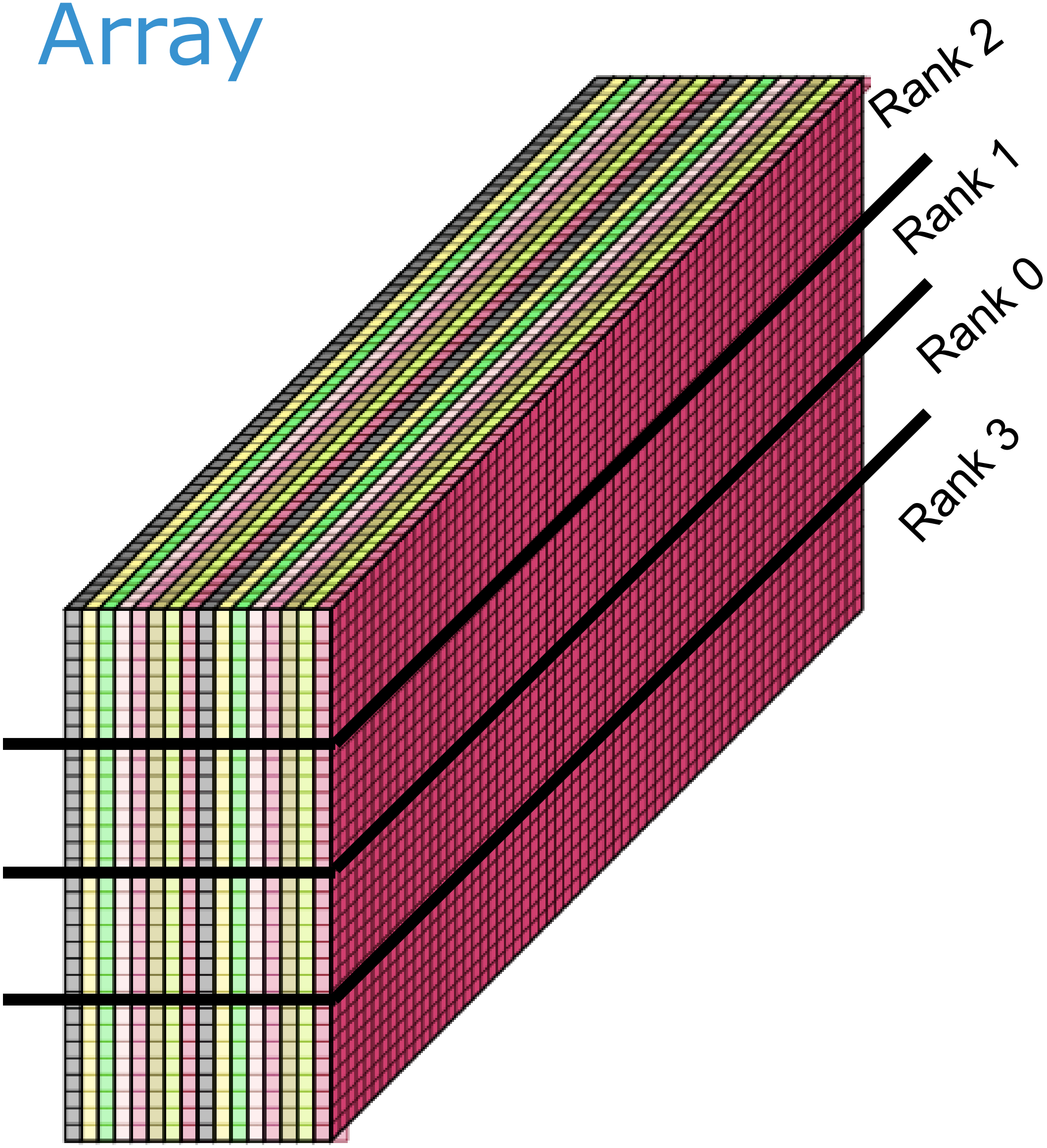
No data is IOD resident

```
# Verify data purged:
[root@lola-3 scripts]# plfs_query /tmp/iod_plfs/containerA/6000000110246979/1 \
           | grep dropping.data
[root@lola-3 scripts]#
```

Step 4.5: Read Array (from DAOS)

```
Running command: 1 read array_obj 0,0,0 8,8,8 2,4,8 8,8,8
Rank 0 ID 0: start[ 0,8,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 1 ID 0: start[ 0,16,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 2 ID 0: start[ 0,24,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 3 ID 0: start[ 0,0,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Call iod_array_read(), oh: 258, tid: 0x1
Rank 0 ID 0: iod_array_read() returns 0
Finish to read array - passed (rc = 0)
Rank 3 ID 0: iod_array_read() returns 0
Rank 1 ID 0: iod_array_read() returns 0
Rank 2 ID 0: iod_array_read() returns 0
```

Step 4.5: Read Array (from DAOS)



Step 5.1/5.2: Fetch Array into 2 replicas

```
# layout contains: location,type,target_start,target_num,strip_size,<dims_seq>
# location ------- "BB" (i.e. IOD_LOC_BB)
# type ----------- only support stripped layout(i.e.IOD_LAYOUT_STRIPED)
# target_start --- the beginning IOD
# target_num ----- the number of IODs. If 0, it defaults to all IODs
# stripe_size ---- resharding granularity. The number of cells for contiguous layout array
# dims_seq ------- For contiguous layout, it is an array with num_dims size
#
# ######## Step 5.1 - prefetch object 1st: using one IOD and default dims_seq
> Press enter to continue...

Running command: 1 fetch array_obj tag1 BB,striped,0,1,64,<>
Call iod_obj_fetch(), tid: 0x1...
Rank 0: check all information saved for fetch:
fake_tag  tag                location    type       target_start  target_num  stripe_size   dims_seq
tag1      0x6000000000000001  BB        striped     0            1           64           (nil)
Finish to fetch object - passed (rc = 0)

Running command: 1 fetch array_obj tag2 BB,striped,0,0,1024,<2,0,1>
Call iod_obj_fetch(), tid: 0x1...
Rank 0: check all information saved for fetch:
fake_tag  tag                location    type       target_start  target_num  stripe_size   dims_seq
tag1      0x6000000000000001  BB        striped     0            1           64           (nil)
tag2      0x6100000000000001  BB        striped     0            4           1024           2, 0, 1
Finish to fetch object - passed (rc = 0)
```

Step 5.3: Query Replica 1 Shards on IONs

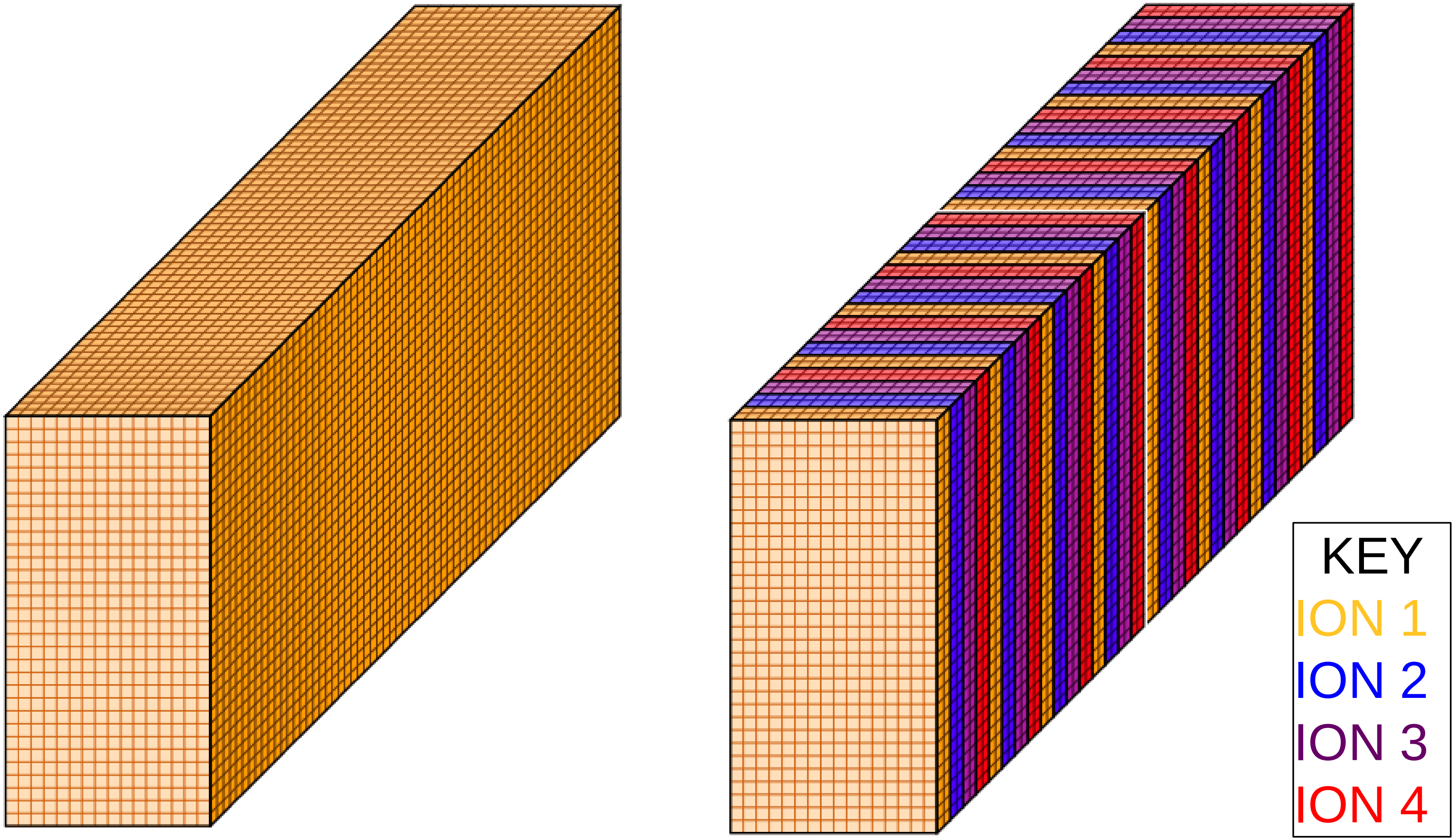
```
Running command: * querymap array_obj tag1
Call iod_obj_query_map(), tag: 0x6000000000000001
Rank 0 in tid 0x6000000000000001: dump IOD returned map:
obj_id            type      n_range
6000000171b6056d   ARRAY      1
range:   0, start:   0,  0,  0,  end:  15, 31, 63,  n_cell:  32768, loc: /mnt/lustre3/.plfs__shadow
Finish to query object map - passed (rc = 0)
```

Step 5.4: Query Replica 2 Shards on IONs

```
# ######## Step 5.4 - Query object's map: 2nd prefetched copy (should striped on all IODs)
> Press enter to continue...

Running command: * querymap array_obj tag2
Call iod_obj_query_map(), tag: 0x6100000000000001
Rank 0 in tid 0x6100000000000001: dump IOD returned map:
obj_id            type      n_range

6000000171b6056d   ARRAY     32
range:   0, start:   0,  0,  0,  end:  15, 31,  1,  n_cell:   1024, loc: /mnt/lustre1/.plfs__shadow
range:   1, start:   0,  0,  2,  end:  15, 31,  3,  n_cell:   1024, loc: /mnt/lustre2/.plfs__shadow
range:   2, start:   0,  0,  4,  end:  15, 31,  5,  n_cell:   1024, loc: /mnt/lustre3/.plfs__shadow
range:   3, start:   0,  0,  6,  end:  15, 31,  7,  n_cell:   1024, loc: /mnt/lustre4/.plfs__shadow
range:   4, start:   0,  0,  8,  end:  15, 31,  9,  n_cell:   1024, loc: /mnt/lustre1/.plfs__shadow
range:   5, start:   0,  0, 10,  end:  15, 31, 11,  n_cell:   1024, loc: /mnt/lustre2/.plfs__shadow
range:   6, start:   0,  0, 12,  end:  15, 31, 13,  n_cell:   1024, loc: /mnt/lustre3/.plfs__shadow
range:   7, start:   0,  0, 14,  end:  15, 31, 15,  n_cell:   1024, loc: /mnt/lustre4/.plfs__shadow
<snip>
range:  28, start:   0,  0, 56,  end:  15, 31, 57,  n_cell:   1024, loc: /mnt/lustre1/.plfs__shadow
range:  29, start:   0,  0, 58,  end:  15, 31, 59,  n_cell:   1024, loc: /mnt/lustre2/.plfs__shadow
range:  30, start:   0,  0, 60,  end:  15, 31, 61,  n_cell:   1024, loc: /mnt/lustre3/.plfs__shadow
range:  31, start:   0,  0, 62,  end:  15, 31, 63,  n_cell:   1024, loc: /mnt/lustre4/.plfs__shadow
Finish to query object map - passed (rc = 0)
```

Replica 1

Replica 2



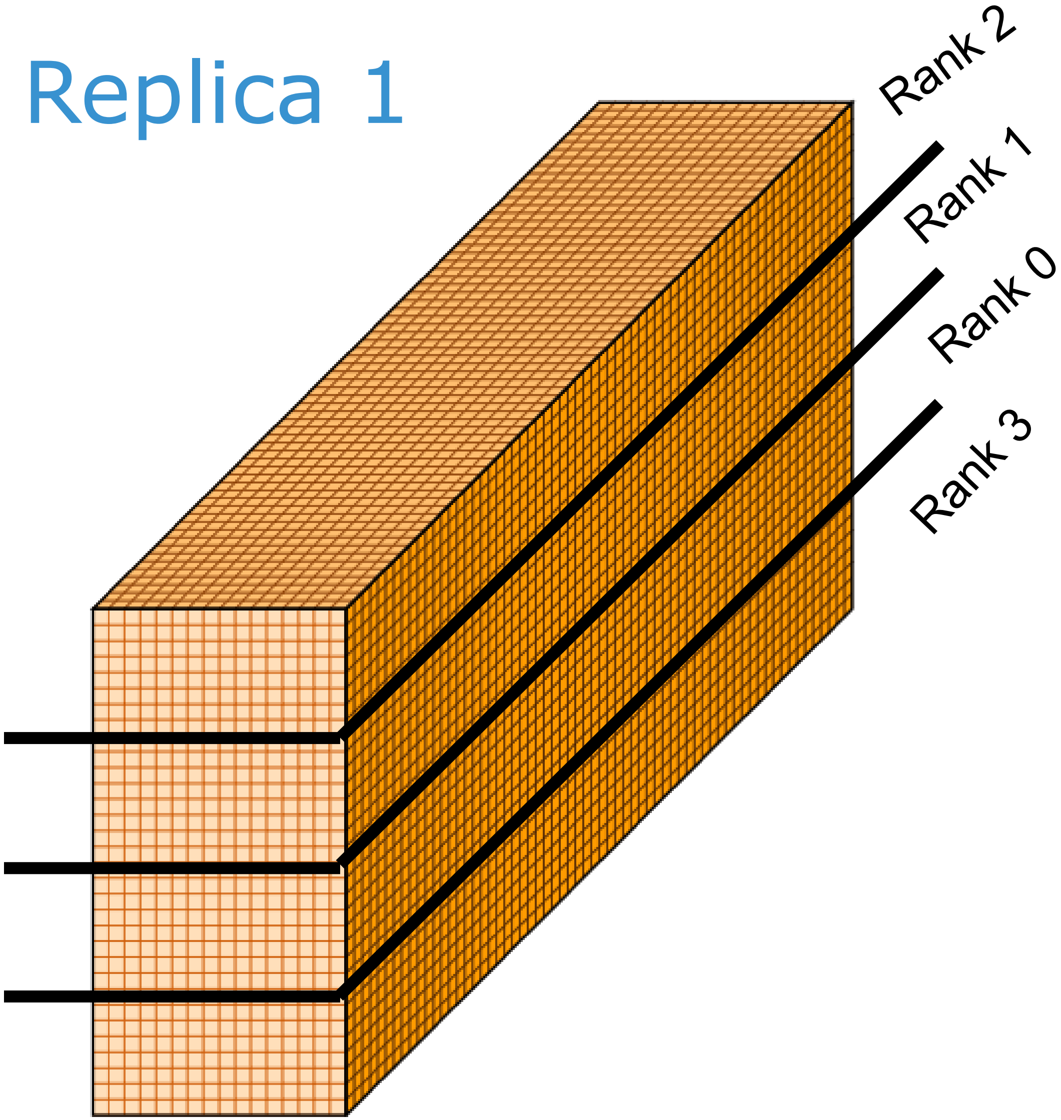
Step 5.5: Read Replica 1

```
# ######## Step 5.5 – read data of the 1st prefetched copy with its tag
Check prefetched copy of object array_obj(0x6000000171b6056d) at TAG = 0x6000000000000001
Rank 0 ID 0: start[ 0,8,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 1 ID 0: start[ 0,16,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 2 ID 0: start[ 0,24,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 3 ID 0: start[ 0,0,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Call iod_array_read(), oh: 330, tid: 0x6000000000000001
Rank 0 ID 0: iod_array_read() returns 0
Rank 0 ID 0: array IO data verification succeed
Finish to read array - passed (rc = 0)
Finish to readprefetched copy - passed (rc = 0)

Rank 2 ID 0: iod_array_read() returns 0
Rank 1 ID 0: iod_array_read() returns 0
Rank 3 ID 0: iod_array_read() returns 0
```

Step 5.6: Read Replica 2

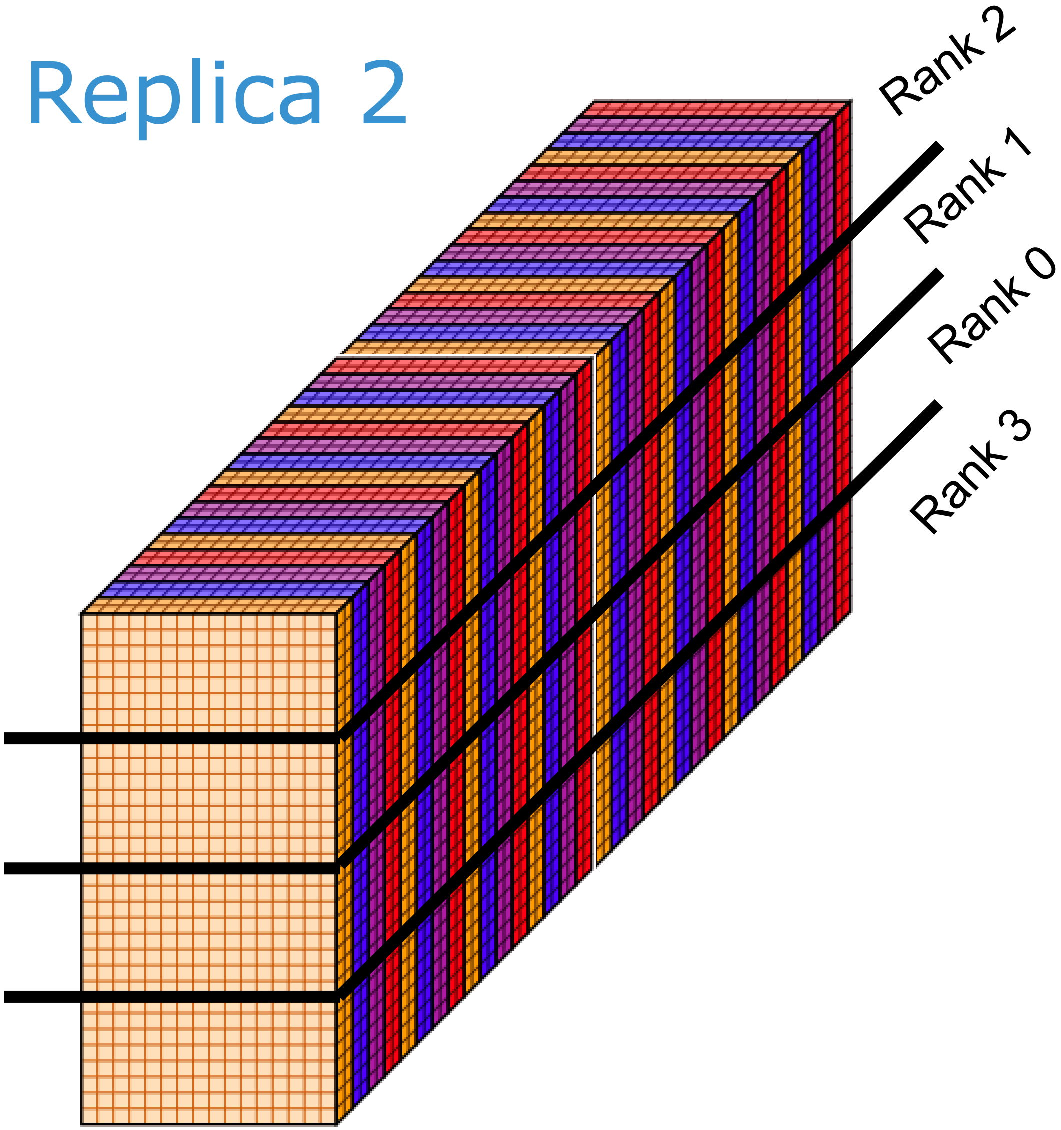
```
# ######## Step 5.6 – read data of the 2nd prefetched copy with its tag
Check prefetched copy of object array_obj(0x6000000171b6056d) at TAG = 0x6100000000000001
Rank 0 ID 0: start[ 0,8,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 1 ID 0: start[ 0,16,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 2 ID 0: start[ 0,24,0 ], stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Rank 3 ID 0: start[ 0,0,0 ],  stride[ 8,8,8 ], count[ 2,1,8 ], block[ 8,8,8 ]
Call iod_array_read(), oh: 338, tid: 0x6100000000000001
Rank 1 ID 0: iod_array_read() returns 0
Rank 2 ID 0: iod_array_read() returns 0
Rank 0 ID 0: iod_array_read() returns 0
Finish to read array - passed (rc = 0)
Finish to readprefetched copy - passed (rc = 0)
```

Reading Replica 1



KEY
ION 1
ION 2
ION 3
ION 4

Reading Replica 2



KEY

ION 1

ION 2

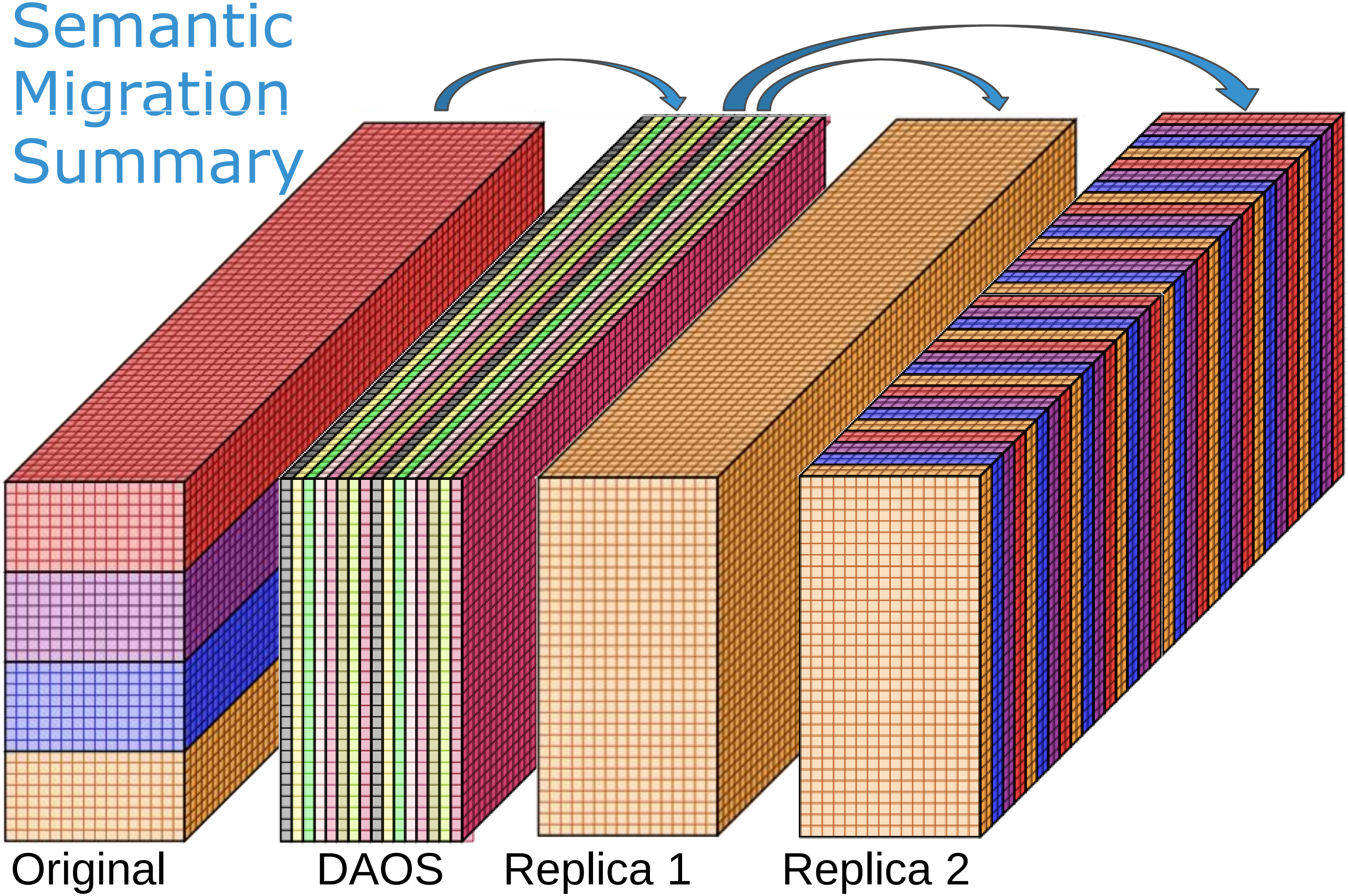
ION 3

ION 4

Step 5.7/6: Cleanup / Fini

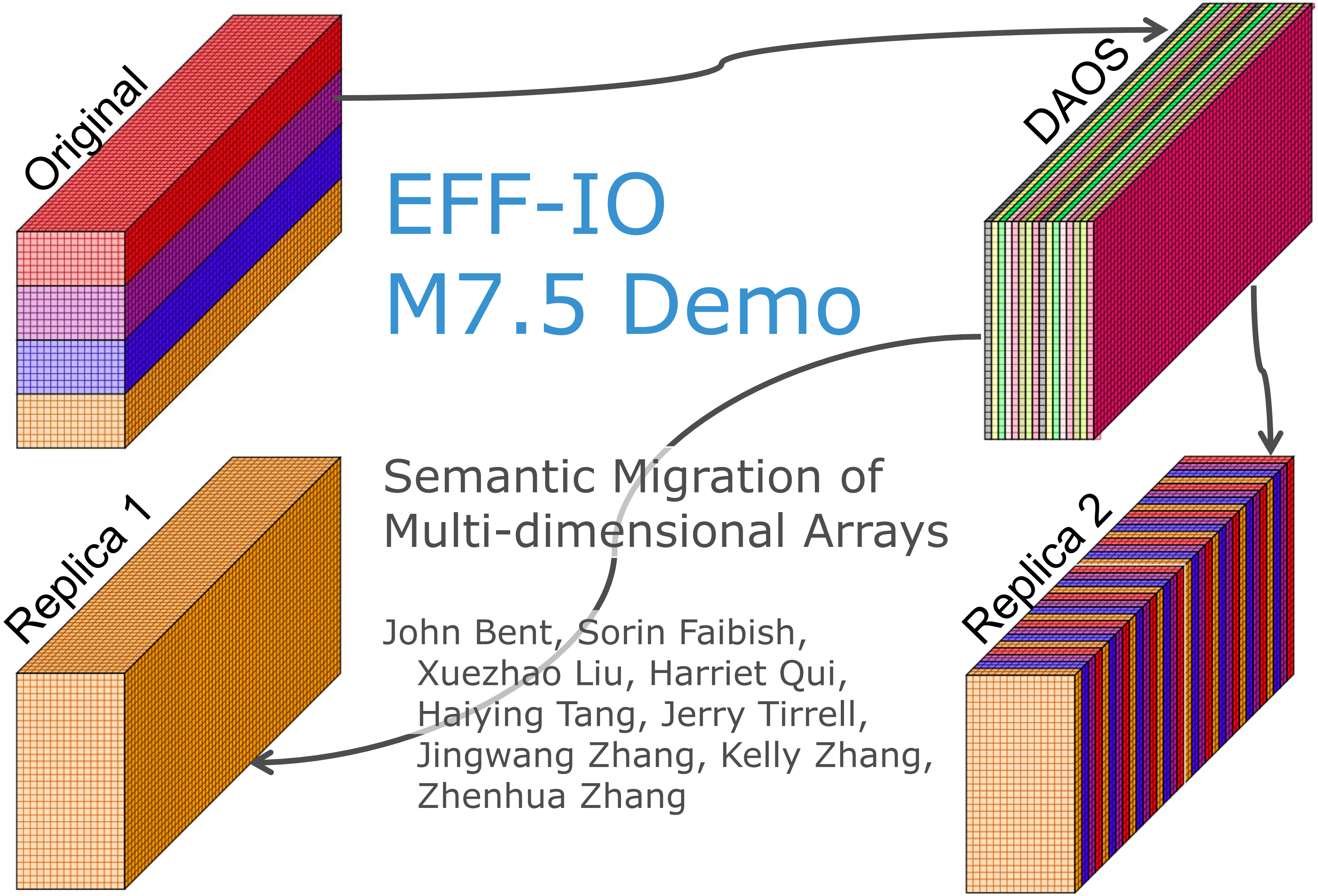
```
# ######## Step 5.7 - finish TID1 for read
> Press enter to continue...

Running command: 1 transfinish normal
Call iod_trans_finish(), tid: 0x1, num_ranks: 0, flag: 0
Finish to finish transaction - passed (rc = 0)
# -----------------------------------------------------------------------
# ######## Step 6: close the container and quit.
> Press enter to continue...

Running command: * contclose
Call iod_container_close(), handle: 25...
Finish to close container - passed (rc = 0)
> Press enter to continue...

Running command: * quit

M7.5_case1_array_migration_ops: overall_result============= 0(Success)
M7.5_case1_array_migration_ops: overall_result============= 0(Success)
M7.5_case1_array_migration_ops: overall_result============= 0(Success)
M7.5_case1_array_migration_ops: overall_result============= 0(Success)
```

Semantic Migration Summary



EMC^2®

EFF-IO M7.5 Demo



Semantic Migration of Multi-dimensional Arrays

John Bent, Sorin Faibish,
Xuezhao Liu, Harriet Qui,
Haiying Tang, Jerry Tirrell,
Jingwang Zhang, Kelly Zhang,
Zhenhua Zhang