COM Verification

Alan G. Yoder, Ph.D.
SNIA Technical Council
Huawei Technologies, LLC

Outline

- Parity RAID
- Thin Provisioning
- Read-only Delta Snapshots
- Writeable Delta Snapshots
- Data Deduplication
- Compression
- Auto-tiering

Validating the COMs

- Step 1: select COM(s) to validate
- Step 2: generate data set if required
- Step 3: validate the COM(s)

Validating the COMs -- selection

- Parity RAID is REQUIRED
  - A bar to entry, not a COM to validate
  - Must be a customer-selectable option
  - No validation requirement
  - check the equipment data sheet
  - cross-check against the list of acceptable RAID types
    - (next slide)
- One COM is required for Online 3 and 4 systems
  - Pick the easiest one to do first
    - Thin provisioning and delta snapshots are good candidates
    - Dedup and compression are a bit more work
  - Validate all that are available for extra credit

Parity RAID

SNIA Green Storage Initiative

Common types of RAID

RAID 0	simple striping	not really RAID	
RAID 1	mirroring	NOT parity RAID	
RAID 4	parity on a separate drive	okay for ES	*only good for smaller drives*
RAID 5	parity striped across drives	okay for ES	
RAID 6	double parity	okay for ES	*protection against failures during RAID reconstruct*
"erasure codes"	non-XOR parity	okay for ES	
distributed parity	multiple parity, widely distributed [1]	okay for ES	
RAID 0+1, 1+0, RAID 10	striping+mirroring	NOT parity RAID	
replication	e.g. Hadoop, AWS	NOT parity RAID	

[1] usually "RAID groups" per se

Parity RAID

- Don’t get hung up on terminology
  - The intent of the spec is to allow any data protection scheme that uses less space for data protection than RAID 1, i.e. less than one full copy
- Do beware of imposters
  - Advocates of in-house replication in particular
  - The spec doesn’t cover DR’ed systems

    (DR = Disaster Recovery)

Validating the COMs – generating the data sets

- Download code
  - https://sourceforge.net/projects/sniadeduptest/
  - file is gendddata.c
- Compile code with gcc or Visual Studio
  - gcc -o gendddata gendddata.c
- Run the executable on the host to generate the datasets
  - 6GiB free space required
  - gendddata . (period means the current directory)
- This generates datasets in three directories
  - irreducibledataset/ compressibledataset/ dedupabledataset/
  - 17510 files in each directory

Verifying the COM– Thin Provisioning

1. Determine the amount of free space FS_{sot} available on the SUT as seen by the storage admin
   - Satisfy yourself that the admin is using a standard tool for this
2. On the SUT, allocate a container 15GB in size
   - The admin will do this
   - 15GB is the “nominal” size
     - The SUT should actually allocate considerably less than this
     - Check by determining the amount of free space using the same tool as in step 1
3. Mount the container on the host test machine
   - You need a filesystem mounted via iSCSI or FC
   - *The host system should indicate that it is a 15GB container*

Verifying the COM– Thin Provisioning

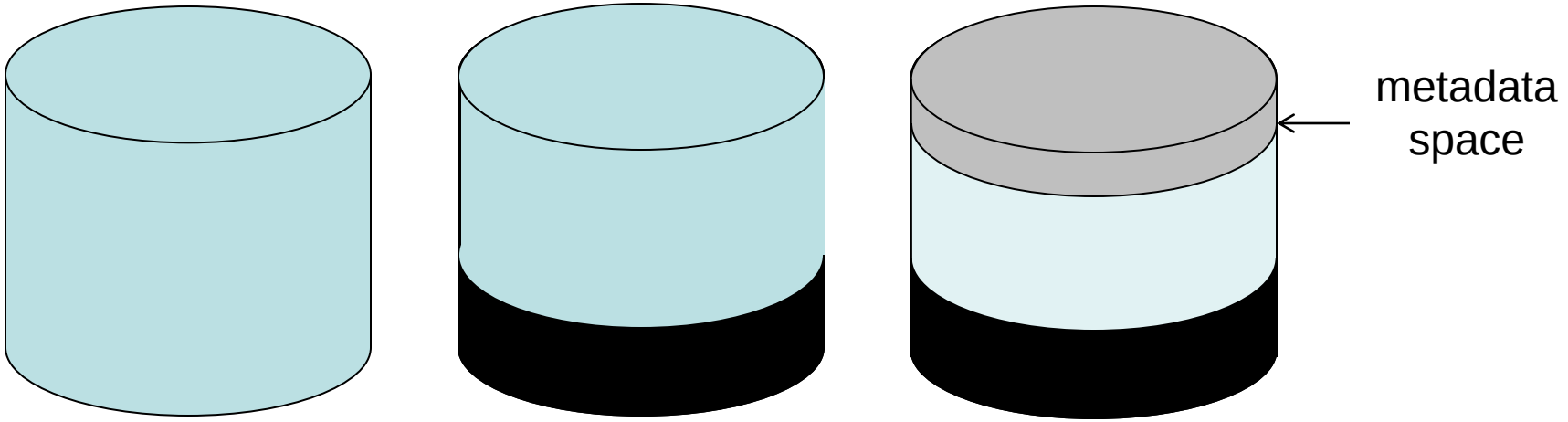
4. On the host, write the irreducible data set to the container

5. After a suitable amount of time, not to exceed 1 hour
   - determine the amount of free space FS_{eot} available on the SUT as seen by the storage admin
   - Use the same free space determination as in step 1

6. Calculate the amount of formatted capacity I_{com} used by the test
   - $I_{com} = FS_{sot} – FS_{eot}$
   - If I_{com} is less than 3GB, then the SUT passes the test.

Delta Snapshot verification

Basic idea:



(1) 15GB volume (2) put some data (3) take snapshot

(4) verify that metadata space is not over 2.5GB (1/6 of 15GB)

Delta Snapshot verification, cont.

- New development:
  - 15GB is a pretty small container in 2014
  - Some systems use more than 2.5GB for the base metadata
    - They assume TB-scale volumes, and 2.5GB is 0.25% of 1TB
- Therefore:
  - It is permissible to scale the sizes listed in these materials
  - E.g. 150GB volume, 20GB data, 25GB metadata limit

Verifying the COM—RO Delta SSs

- **7.4.5.4.1 Heuristic 1: Readonly delta snapshots**

- The method varies according to where the SUT places snapshots.
  - Follow the appropriate procedure to set up and take the snapshot
  - Correct sequences are either
    - step1—for separate containers—then step 3, or
    - step 2—for same container—then step 3

Verifying the COM—RO Delta SSs

1. For a SUT which places snapshots in separate containers:

a. On the SUT, create two containers, each 15GB in size.
    a. One in the live data container/partition, one in the snapshot container
b. On a host, mount a filesystem on the first container, via iSCSI or FC
c. Determine the amount of free space FS_{sot} available on the SUT as seen by the storage administrator.
    a. Satisfy yourself that the admin is using a standard tool for this
d. Copy all the files in the irreducible data set to the first container.
e. Perform a readonly delta snapshot of the first container and expose it through the second container, disabling any optional background copying mechanism. As an example, the snapshot of lun1 may be exposed as lun2.
f. Perform whatever steps are necessary to mount the second container as a file system. Open a small file on this file system (one of the files just copied), read some data from it, and close the file. Confirm that the file has been successfully read.

Verifying the COM—RO Delta SSs

2. For a SUT which places snapshots on the originating container

a. On the SUT, create a container of 15GB in size.
b. On a host, mount a filesystem on the first container, via iSCSI or FC
c. Determine the amount of free space FS_{sot} available on the SUT as seen by the storage administrator.
    a. Satisfy yourself that the admin is using a standard tool for this
d. Copy all the files in the irreducible data set to the container.
e. Perform a read-only snapshot of the container, disabling any optional background copying mechanism.
    a. The admin will do this
f. Perform whatever steps are necessary to mount the container as a file system. Open a file in the snapshot (i.e. one of the files in the irreducible data set), read some data from it, and close the file. Confirm that the file portion has been successfully read.

Verifying the COM—RO Delta SSs

3. Finally (after performing (1) or (2)

a. Determine the amount of free space, FS_{eot}, available on the container containing the snapshot, as seen by the storage administrator.
b. Calculate the space required for the snapshot
    a. $I_{com} = FS_{sot} - FS_{eot}$
c. If I_{com} is less than 2.5 GB then the SUT passes the test

Verifying the COM—R/W Delta SSs

- **7.4.5.4.2 Heuristic 2: Writeable delta snapshots**

- The method varies according to where the SUT places snapshots.
  - Follow the appropriate procedure to set up and take the snapshot
  - Correct sequences are either
    - step1 then step 3, or
    - step 2 then step 3

Verifying the COM—R/W Delta SSs

1. For a SUT which places snapshots in separate containers:

a. On the SUT, create two containers, each 15GB in size.
    a. One in the live data container/partition, one in the snapshot container
b. On a host, mount a filesystem on the first container, via iSCSI or FC
c. Determine the amount of free space FS_{sot} available on the SUT as seen by the storage administrator.
    a. Satisfy yourself that the admin is using a standard tool for this
d. Copy all the files in the irreducible data set to the first container.
e. Perform a writeable snapshot of the first container and expose it through the second container, disabling any optional background copying mechanism. As an example, the snapshot of lun1 may be exposed as lun2.
f. Perform whatever steps are necessary to mount the second container as a file system. Open a small file on this file system (i.e. one of the files in the irreducible data set), write a few characters to it, and close the file. Confirm that the file has been successfully written with its new contents.

Verifying the COM—R/W Delta SSs

2. For a SUT which places snapshots on the originating container

a. On the SUT, create a container of 15GB in size.
b. On a host, mount a filesystem on the first container, via iSCSI or FC
c. Determine the amount of free space FS_{sot} available on the SUT as seen by the storage administrator.
   - Satisfy yourself that the admin is using a standard tool for this
d. Copy all the files in the irreducible data set to the container.
e. Perform a read-only snapshot of the container, disabling any optional background copying mechanism.
   - The admin will do this
f. Perform whatever steps are necessary to mount the container as a file system. Open a file in the snapshot (i.e. one of the files in the irreducible data set), write a few characters to it, and close the file. Confirm that the file has been successfully written.

Verifying the COM—R/W Delta SSs

3. Finally (after performing (1) or (2)

a. Determine the amount of free space, FS_{eot}, available on the container containing the snapshot, as seen by the storage administrator.
b. Calculate the space required for the snapshot
    a. $I_{com} = FS_{sot} - FS_{eot}$
c. If I_{com} is less than 2.5 GB and the small file was successfully written onto the read-only delta snapshot destination, then the SUT passes the test.

Verifying the COM—Dedup

- Same basic idea as Delta Snapshots
  - Verify that the metadata overhead is not too high
- All variations acceptable for ES on online systems
  - Must meet performance criterion
    - It's not a high bar – most systems should pass
- Same scaling issue as with snapshots
  - 15GB too small on some systems
  - Some mathematical difficulties with simple scaling however
    - work in progress
  - Test a different COM if this is an issue
    - only need one for ES

Verifying the COM—Dedup

a. On the SUT, create a container 15GB in size.
b. For a SUT exporting block storage, perform whatever steps are necessary to make the container visible on the host from which tests are being run, and create and mount a local file system on that container.
c. For a SUT exporting file storage, mount the container on the host via any chosen file protocol such as NFS or CIFS.
d. Determine the amount of free space FS_{sot} available on the container as seen by the storage administrator.
e. Write the deduplication-reducible data set (the files in the dedupabledataset/ directory) to the container.
f. Wait a suitable amount of time as specified by the TEST SPONSOR for non-inline deduplication processes to have completed.
g. Determine the amount of free space FS_{eot} available on the container as seen by the storage administrator.
h. Calculate the amount of formatted capacity saved by data deduplication
   - $I_{com} = (1 – [(FS_{eot} - FS_{sot}) / S_{ds}]) * 100\%$ Note: S_{ds} = sizeof(dedupabledataset/*)
i. If I_{com} is greater than 10%, then the SUT passes the test.

Verifying the COM—Compression

a. On the SUT, create a container 15GB in size.
b. For a SUT exporting block storage, perform whatever steps are necessary to make the first container visible on the host from which tests are being run, and create and mount a local filesystem on that container.
c. For a SUT exporting file storage, mount the container on the host via any chosen file protocol such as NFS or CIFS.
d. Determine the amount of free space FS_{sot} available on the container as seen by the storage administrator.
e. Write the compression-reducible data set (the files in the compressibledataset/ directory) to the container.
f. Wait a suitable amount of time as specified by the TEST SPONSOR for non-inline compression processes to have completed.
g. Determine the amount of free space FSeot available on the container as seen by the storage administrator.
h. Calculate the amount of formatted capacity saved by compression
   - $I_{com} = (1 – [(FS_{eot} - FS_{sot}) / S_{ds}]) * 100\%$ Note: S_{ds} = sizeof(compressibledataset/*)
i. If I_{com} is greater than 10%, then the SUT passes the test.

Proposed new COM (info only)

- Auto-tiering
  - A technology that automagically moves cold data to fat, slow (more energy efficient) drives
- The hot banding test already rewards auto-tiering
- No current plans to promote it as a COM

Q & A