How to Read the Entirety of Each Volume

The issue documented in

http://www-01.ibm.com/support/docview.wss?rs=591&uid=ssg1S1004863

requires the user to read the entirety of the compressed volumes to validate whether any medium errors exist on the volume.

When the host attempts to read any of the unrecoverable data, the system will return an error to the host. This error is known as a medium error.

IMPORTANT:

- These action plans should not be started until the system has been upgraded to V7.3.0.6 or later.
- Performing data validation will generate additional IO operations to the disks. Running too many validation activities in parallel may overload the backend storage and cause performance problems.

There are two classes of approach that can be used to perform this validation:

1. Using the Copy Services Features in the SVC or Storwize system
2. Using Host applications

Using the Copy Services Features in the SVC or Storwize System

a/ If the volume already has two copies using volume mirroring

Note: This is the best process to use if the volume is already mirrored, because it does not require any additional capacity, and it will be able to repair medium errors that are found as long as the other copy does not have a medium error in the same location on the volume.

There is an online process which will read the entirety of both copies of the volume, and compare the contents. If this process detects any medium errors on copy A then the data is copied from copy B to copy A to repair the medium error.

This process can be started by right clicking on either of the copies in the GUI and selecting "Validate Volume Copies". You should then choose the option called "Generate Event of Differences". The CLI command for this option is:

```
svctask repairvdiskcopy -validate <volume ID or volume name>
```

The progress of the validation can be monitored in the Running tasks pod in the GUI, or using the following CLI:

```
svcinfo lsrepairvdiskcopyprogress
```

b/ If the volume is not currently mirrored

The Copy services features of SVC are all capable of reading the entirety of a volume, to create a second copy of the data. The recommended copy service to use to perform this validation is Volume Mirroring. This is because the other copy services will stop when they find the first medium error, but the Volume Mirroring feature will read the entirety of the volume and provide details of all medium errors in the volume.

To perform the validation you should right click on the volume in the GUI and select "Volume Copy Actions" -> "Add Mirrored Copy" and follow the wizard. The new copy can be thin provisioned, compressed or fully allocated, and can be in any storage pool which contains sufficient capacity. However be aware that if the second copy is in a lower performance storage pool, there is a small chance that the write performance of the volume might be reduced by adding the slower copy. The CLI for adding a volume copy is `svctask addvdiskcopy.` Please review the command line help for full details about how to use this command.

If the volume mirroring system detects any medium errors on the volume it will log one or more of the following errors against the mirrored copy:

060007 / 1850 "Compressed volume copy has bad blocks"
029001 / 1840 "MDisk medium errors exist"

The progress of the mirroring can be monitored in the Running tasks pod in the GUI, or using the following CLI:

```
svcinfo lsvdisksyncprogress
```

Once the mirror has been completed, and if no medium errors were detected, then either of the copies can be deleted by right clicking on the copy to be deleted and selecting "Delete this Copy". The CLI is

```
svctask rmvdiskcopy –copy <copy id> <volume id or name>
```

Please take care when picking the copy you wish to delete.

Note: that if any medium errors exist on the source volumes for any of the other copy services (FlashCopy, Metro Mirror, Global Mirror or Global Mirror with Change volumes), then the synchronisation phase of these features will stop with an error if the synchronisation process detects a medium error.

Using Host Applications

There are a number of host applications that can be used to read the entirety of a volume, or the entirety of a filesystem. Many of these will be dependent upon the specific combination of operating system, filesystem and application which is present on each host. A small number of applications may also have the ability to repair a medium error.

Please be aware that if an application attempts to read data from disk and receives a medium error, the application may not continue running, therefore it may not be appropriate to read all of the data using the production application.

Here are some generic tools that can be used to read the entirety of a volume or a filesystem

- DD

  AIX, Linux and other Unix based operating systems have a built in command called dd. This command will read data from either a file or a raw volume, and output the data somewhere else. Below is an example of using dd to read the entirety of a linux scsi device (called "/dev/sda") and discard the data (by sending it to /dev/null). There are many more features of dd that can be used to modify its behaviour for your specific needs

  ```
  dd if=/dev/sda of=/dev/null
  ```

  If DD finds a medium error it will return an "Input/output error".

  Checking the kernel log for the presence of "Sense Key : Medium Error" messages using the "dmesg" command will confirm whether the Input/output error was a result of a medium error.

- Host Backups

  A full backup of the host (not an incremental backup) will either read all of the volume or it will read all of the files in the filesystem. Either of these will be sufficient to validate that there are no medium errors present in any of the essential files in the filesystem.

  NOTE: If using a full host backup to look for medium errors, it is essential to ensure that the previous backup is not lost, in case the previous backup is needed to recover the data which has suffered medium errors.

  If the backup detects a medium error it will likely appear in the backup logs as the inability to read a file, but the exact nature of the error will depend upon the

backup software. It is likely that you will need to examine the backup logs to determine whether any files could not be read and backed up.

If medium errors are detected by any of the above processes contact IBM support for more assistance.