IBM Spectrum Virtualize HyperSwap Configuration

Author: John Wilkinson

1 Overview of HyperSwap

The HyperSwap® high availability function in the IBM® Spectrum Virtualize software allows business continuity in the event of hardware failure, power failure, connectivity failure, or disasters such as fire or flooding. It is available on the IBM SAN Volume Controller, IBM Storwize® V7000, IBM Storwize V7000 Unified, and IBM Storwize V5000 products.

The HyperSwap function provides highly available volumes accessible through two sites at up to 300km apart. A fully-independent copy of the data is maintained at each site. When data is written by hosts at either site, both copies are synchronously updated before the write operation is completed. The HyperSwap function will automatically optimize itself to minimize data transmitted between sites and to minimize host read and write latency.

If the nodes or storage at either site go offline, leaving an online and accessible up-to-date copy, the HyperSwap function will automatically fail over access to the online copy. The HyperSwap function also automatically resynchronizes the two copies when possible.

The HyperSwap function builds on two existing technologies in the product: Non-Disruptive Volume Move (NDVM) function introduced in version 6.4 of the SVC software; and the Remote Copy features that includes Metro Mirror, Global Mirror, and Global Mirror with Change Volumes.

A HyperSwap function is already available when using the IBM DS8000 family of products together with PowerHA System Mirror for AIX or GDPS for z/OS. The HyperSwap functions on those environments make use of specific software on that host system.

The HyperSwap function in the SVC software works with the standard multipathing drivers that are available on a wide variety of host types, with no additional host support required to access the highly available volume. Where multipathing drivers support ALUA, the storage system will tell the multipathing driver which nodes are closest to it, and should be used to minimise I/O latency. You just need to tell the storage system which site a host is connected to, and it will configure host pathing optimally. This aspect is also enabled for the Enhanced Stretched Cluster function.

1.1 Comparison with Enhanced Stretched Cluster

Many of the aspects described so far are the same as those of the existing Enhanced Stretched Cluster function, introduced in version 7.2 of the SVC software. Following is a list of key differences between the Enhanced Stretched Cluster and HyperSwap functions.

	SVC and Storwize Stretched Cluster	SVC and Storwize HyperSwap
Products that function is available on	SVC only	SVC with 2 or more I/O groups; Storwize V7000 and Storwize V5000)
Complexity of configuration	CLI or GUI on single system; simple object creation	Multiple step CLI-based configuration on single system; Simple object creation through GUI and CLI to come in future release
Sites data stored on	2	2
Distance between sites	Up to 300km	Up to 300km
Independent copies of data maintained	2	2 (4 if additionally Volume Mirroring to two pools in each site)
Technology for host to access multiple copies and automatically fail over	Standard host multipathing driver	Standard host multipathing driver
Cache retained if only one site online?	No	Yes
Host-to-storage-system path optimization	Manual configuration of preferred node per volume prior to 7.5, automatic based on host site as HyperSwap from 7.5	Automatic configuration based on host site (requires ALUA/TPGS support from multipathing driver)
Synchronization and resynchronization of copies	Automatic	Automatic
Stale consistent data retained during resynchronization for disaster recovery?	No	Yes
Scope of failure and resynchronization	Single volume	One or more volumes, user configurable
Ability to use FlashCopy together with High Availability solution	Yes (though no awareness of site locality of data)	Limited: can use FlashCopy maps with HyperSwap volume as source, avoids sending data across link between sites
Ability to use Metro Mirror, Global Mirror, or Global Mirror together with High Availability solution	One remote copy, can maintain current copies on up to four sites	No
Maximum highly available volume count	4096	1024
Licensing	Included in base product	Requires Remote Mirroring license for volumes. Exact license requirements may vary by product.

The Enhanced Stretched Cluster function uses a “stretched” system topology, and the HyperSwap function uses a “hyperswap” topology. These both spread the nodes of the system across two sites, with storage at a third site acting as a tiebreaking quorum device.

The topologies differ in how the nodes are distributed across the sites:

- For each I/O group in the system, the “stretched” topology has one node on one site, and one node on the other site. The topology works with any number of I/O groups from 1 to 4, but as the I/O group is split into two locations, this is only available with SVC not Storwize.
- The “hyperswap” topology locates both nodes of an I/O group in the same site, making this possible to use with either Storwize or SVC products. Therefore to get a volume resiliently stored on both sites, at least two I/O groups are required.

The “stretched” topology uses fewer system resources, allowing a greater number of highly available volumes to be configured. However, during a disaster that makes one site unavailable, the SVC system cache on the nodes of the surviving site will be disabled.

The “hyperswap” topology uses additional system resources to support a full independent cache on each site, allowing full performance even if one site is lost. In some environments a “hyperswap” topology will provide better performance than a “stretched” topology.

Both topologies allow full configuration of the highly available volumes through a single point of configuration. The Enhanced Stretched Cluster function may be fully configured through either the GUI or the CLI. The HyperSwap function can initially be configured through the system CLI and will be fully configurable through the GUI and a simplified set of CLI commands in a future version of the SVC software. This document describes how to configure and monitor the HyperSwap function through the CLI.

1.2 Disaster Recovery

The HyperSwap function automatically controls synchronization and resynchronization of volume copies. Just before resynchronizing data to a volume copy, that copy will usually contain consistent but stale (out-of-date) data. The storage system will automatically retain that consistent data during the resynchronization process using Change Volume technology.

What this means is that if a problem occurs at the site with the online copy before resynchronization completes, taking that copy offline, you have the opportunity to manually enable read and write access to the consistent, older copy of data, allowing the use of this data for disaster recovery. This option would typically be taken if you know that the offline copy will remain offline for an extended period, and the consistent but older data is useful enough to keep your business running.

As normal with disaster recovery solutions that allow business continuity with older data, after the problem is resolved restoring access to the offline copy, you can choose to either revert to that now-online copy, which before the disaster held the latest copy of data, or continue to work on the stale data used during the disaster. With either choice, the other copy is resynchronized to match the chosen copy.

1.3 Volume Groups

One major advantage of the HyperSwap function, compared to Enhanced Stretched Cluster, is that it’s possible to group multiple volumes together for high availability. This is important where an application spans many volumes, and requires the data to be consistent across all those volumes.A scenario where the lack of this would impact availability is

1. Site 2 goes offline
2. Application continues to write to its volumes, changes only applied to site 1
3. Site 2 comes online again
4. Volumes are resynchronized back to site 2
5. Site 1 goes offline during the resynchronization

Site 1 is only site that has usable data. Site 2 may have usable data on some volumes but not others. If this process is continued, it is possible that neither site will have a complete copy of data, making a failure on either site impact production I/O.

With Enhanced Stretched cluster, site 2's data would have been made inconsistent on a number of the volumes, by the attempt to resynchronize which did not complete. Even maintaining the old but consistent data, as described in the previous section, isn't enough to guarantee that site 2 has preserved consistency across multiple volumes, as the consistent data may have been captured at different points in time. If the site 1 data cannot be recovered, then some other solution will be needed to recover business operations.

Using volume groups to control the synchronization and failover across many volumes in an application guarantees that all volume copies on a site have data from the same point in time, allowing disaster recovery using that site's volume copies. It also guarantees that at least one site will have an up-to-date copy of every volume in the volume group. And it further guarantees that the other site, if it does not have an up-to-date copy of every volume, it will have a consistent copy of every volume for some out-of-date point-in-time.

1.4 Which high availability solution?

Both of the HyperSwap and Enhanced Stretched Cluster functions provide high availability for your volumes. They have different characteristics, so in some situations one may be more appropriate for your use than the other.

The Enhanced Stretched Cluster function should be used if

- you need synchronous or asynchronous replication using Metro Mirror or Global Mirror to a third site (and potentially fourth site too, if the remote system is also configured with a stretched system topology)
- you need the absolute maximum number of highly available volumes in the system

You should choose the HyperSwap function if

- you are using Storwize V7000 or V5000 products
- you need the full performance given by the storage system cache even during disasters
- you need a guarantee of application-wide failover characteristics
- you need the freedom to use the volume mirror function to convert between fully-allocated, thin-provisioned, and compressed volume types
- you want to have disaster recovery protection during resynchronization of the highly-available copies

2 Initial configuration

There are two steps required to configure a system for HyperSwap. The first step is to configure the components of the system correctly for the “hyperswap” topology. The second step is to create volumes that use that topology.

You should plan to complete all steps to configure sites for nodes, controllers, hosts and the system topology in one session. Do not leave a system in production if only some of these steps have been performed.

Note: in this section, the term “vdisk” will be used for an individual object created with the mkvdisk command. The term “volume” will be used for the whole LUN mapped to hosts that HyperSwap provides high availability for. This is non-standard terminology, being used in this document to distinguish between these. A future release of SVC will remove the need to create or view the individual objects, and so will get rid of the need to distinguish between them in this way.

2.1 Configuring a system for the HyperSwap system topology

You should configure several system objects before selecting the HyperSwap system topology: sites, nodes, controllers, and hosts.

2.1.1 Sites

The ‘site’ corresponds to a physical location that houses the physical objects of the system. In a client installation it might correspond to a true separate office, a different data center building, or simply different rooms or racked areas of a single data center that has been planned to have internal redundancy.

Parameters that specify a site exist in many of the commands discussed later. Sites were introduced in version 7.2 of the SVC software for the “stretched” system topology of the Enhanced Stretched Cluster function. If you have a system previously configured for the stretched system topology, many objects will already have their sites configured.

The user doesn’t have to create sites: there are 4 sites statically defined in the system:

Site ID	Default site name	Objects that can be in site	Purpose
None	*Has no name, cannot be renamed*	Hosts, nodes, controllers	The default site for objects when they are not assigned to a specific site. The HyperSwap topology requires objects to be in a site other than 0.
1	site1	Hosts, nodes, controllers	The first of two sites to perform high availability between. Has no implied preferences compared to site 2.
2	site2	Hosts, nodes, controllers	The second of two sites to perform high availability between. Has no implied preferences compared to site 1.
3	site3	Controllers	A third site providing quorum abilities to act as a tie-break between sites 1 and 2 when connectivity is lost.

Sites 1, 2, and 3 may be renamed from the default name as follows


```
chsite –name westDC 1
```

to rename site 1 as “westDC”. This may help you to understand and describe the location of objects in a more meaningful way. This document will use numbers for sites, but you don’t have to.

2.1.2 Nodes

With a HyperSwap system topology, all nodes in an I/O group have to belong to the same site. You should assign the nodes of at least one I/O group to each of sites 1 and 2.

So, to configure HyperSwap volumes, you will need a system with at least two fully-configured I/O groups. For SVC, this means at least 4 nodes. For Storwize products, at least two control enclosures are required.

For larger systems using the HyperSwap system topology, if most volumes are configured as HyperSwap volumes, it’s preferable to have 2 full I/O groups on each site, instead of 1 I/O group on one site and 2 on the other. This is to avoid the site with only 1 I/O group becoming a bottleneck.

If you aren’t using the HyperSwap function on every volume, then an asymmetric configuration of 2 or even 3 I/O groups on the site containing the non-HyperSwap volumes is possible, with only 1 I/O group on the other site.

Before the HyperSwap system topology may be selected, the site of every node has to be set:

```
chnode –site 1 node1 # (for SVC)
chnodecanister –site 1 node1 # (for Storwize V5000 and V7000)
```

This modifies the existing node 1 from its current site to site 1.

For SVC a node can be added directly into a site as follows:

```
addnode –panelname 103223 –iogrp 0 –site 2
```

Note that nodes may never be assigned to site 3.

For Storwize products, the act of adding a control enclosure in a specific site can automatically assign both the node canisters to that site.

```
addcontrolenclosure –iogrp 1 –sernum 1234567 –site site2
```

When a system is created, the initial node or node canister will be assigned to site 0 by default, and must be manually reconfigured to either site 1 or site 2.

Once the HyperSwap topology has been selected, a node must be in the same site as any other node in the I/O group, and the above commands will fail if that condition would be violated. When adding a new node in the HyperSwap topology, you must provide a site parameter to the addnode command.

> **Note**: when in the HyperSwap topology, if an I/O group contains copies of HyperSwap volumes it must stay in the same site even if all nodes have been deleted from that I/O group. New nodes must be added with the same site attribute as the deleted nodes. The only way to move an I/O group from one site to the other is to remove all HyperSwap volumes using that I/O group, delete the nodes from that I/O group, then re-add them to the I/O group but with the new site attribute.

Typically, a HyperSwap configuration might contain 4 or 8 nodes, either with 1 I/O group on site 1 and 1 I/O group on site 2, or with 2 I/O groups on each of sites 1 and 2. It's possible to configure the system with more I/O groups on one site than the other, but beware that the site with fewer nodes may become a bottleneck.

Each I/O group should have sufficient bitmap capacity defined using the chiogrp command for the HyperSwap volumes in addition to the bitmap capacity requirements of other FlashCopy® and Global Mirror or Metro Mirror objects needed.

For each HyperSwap volume, for every 8GB logical capacity (not physical space), rounded up to the next greatest 8GB, you will need 4kB remote bitmap memory and 8kB flash bitmap memory defined in both I/O groups of the HyperSwap volume.

So for a 2-I/O-group system with the maximum 1024 HyperSwap volumes, each being 100GB, you will need to run the following commands to configure sufficient bitmap space.

```
chiogrp -feature flash -size 104 0
chiogrp -feature flash -size 104 1
chiogrp -feature remote -size 52 0
chiogrp -feature remote -size 52 1
```

2.1.3 Hosts

Host objects have a site parameter. This may be configured on existing host objects as follows:

```
chhost -site 1 hpA-209
```

which will define the ports of host "hpA-209" as being on site 1.

> **IMPORTANT:** The system will dynamically configure host multipathing so that hosts in site 1 will preferentially send I/O to nodes in site 1, and similarly for site 2. So for optimum performance, all of the WWPNs associated with this host object should be located on that site. For clustered host systems attached to both sites, you should define a host object per site to optimize the I/O for each physical server in the clustered host system.

New hosts can be added with a defined site:

```
mkhost -fcwwpn 210100E08B251EE6:210100F08C262EE7 -site 2
```

When HyperSwap volumes are mapped to a host using mkvdiskhostmap, the host must be assigned to either site 1 or site 2.

By default, host objects are associated with all I/O groups. If you use the –iogrp parameter for the mkhost command to override this, you will need to make sure that hosts accessing HyperSwap volumes are associated with at least the I/O groups that the master and auxiliary vdisks of the HyperSwap volumes are cached in. Missing out an association between the host and such an I/O group will prevent the host from being able to access HyperSwap volumes through both sites.

2.1.4 Controllers

An internal storage array, as can be created using the mkarray command, obtains its site information from the node canisters in the control enclosure it is attached to. For virtualized external storage, you have to tell the system where each storage controller is. You can do this for controllers that have been detected:


```
chcontroller –site 1 ds8k-1
```

Controllers should be assigned to site 1 or site 2 if they have any managed mdisks and the system is set to use the HyperSwap topology. Mdisks may only be assigned to storage pools if they are allocated from a storage controller with a well-defined site that matches that of the storage pool.

There's no checking that you've set the sites on all controllers when you change the system to use the HyperSwap topology, but the system won't let you create HyperSwap volumes unless all the mdisks in each storage pool have the site set up correctly. So it's a good idea to set up the controller sites before changing the topology.

Quorum storage must be available in site 3 so that connectivity from all nodes to that storage can be verified correctly. The system will automatically attempt to use storage on sites 1, 2, and 3 for the three quorum disks, selecting the one on site 3 to be the active quorum disk. If you override the choice of quorum disks, you must still select one on each site.

Note that the HyperSwap function requires quorum storage in site 3 to function correctly. This means that all products will need external storage, even if there is otherwise no need for external storage, as only external storage can be configured to site 3.

The controller providing the quorum storage has to specify "extended quorum" support on the SVC supported controller list for the installed software release.

2.1.5 Topology

All the above objects may be set to any of sites 1 or 2 (or 3 for controllers) when the system has been set to the standard system topology:

```
lssystem
 ...
 topology standard
 ...
```

Before the system may be set to the HyperSwap system topology, every node must have a site configured correctly, and it's advisable to set the site of every controller and host too for existing systems. For a new system, you might choose the HyperSwap topology early on in your initial configuration, which will help make sure that objects have their sites set correctly.

Once all the sites have been set, the system can be set to use the HyperSwap topology with this:

```
chsystem –topology hyperswap
```

You won't be able to change the topology back to the standard topology if there are any HyperSwap volumes defined.

2.1.6 Configuring synchronization rates

There are two primary factors that affect the rate synchronization, which are similar to those for the existing Metro Mirror and Global Mirror replication technologies.

2.1.6.1 Partnership bandwidth

The primary attribute to configure is the partnership bandwidth. Prior to the introduction of HyperSwap volumes, this could not be configured for intracluster relationships (i.e. with both copies located in the same system) such as the active-active relationships used for HyperSwap replication.

With HyperSwap-capable systems, the “local partnership” bandwidth can be configured, and represents the amount of physical bandwidth between sites used for synchronization.

For backwards compatibility, this defaults to 25MBps (200Mbit/s) dedicated to synchronization, which may be appropriate for a small environment. For larger systems, or systems with more bandwidth available between sites, you may want to increase this, using

```
chpartnership -linkbandwidthmbits 4000 -backgroundcopyrate 20 localCluster
```

specifying what the bandwidth between sites is, and how much may be used for synchronization. “localCluster” should be replaced by the name of the local system.

As with other types of partnership configuration, the system does not yet use the total amount of bandwidth available in any performance tuning, and only uses the resulting background copy bandwidth to determine HyperSwap synchronization rate. So the above command could be expressed as

```
chpartnership -linkbandwidthmbits 800 –backgroundcopyrate 100 localCluster
```

with the same effect, but the earlier one makes the configuration easier to understand and audit.

Note that the system will attempt to synchronize at this rate where possible if there are any active-active relationships that require synchronization (including resynchronization after a copy has been offline for a period of time). This is true no matter how much new host write data is being submitted requiring replication between sites, so be careful not to configure the synchronization rate so high that this synchronization bandwidth consumption impacts the amount needed for host writes.

2.1.6.2 Relationship bandwidth

The other control on how fast a relationship can synchronize is the system relationshipbandwidthlimit setting. This configures the maximum rate at which synchronization I/O will be generated for a volume. It can be seen with

```
IBM_2076::superuser> lssystem
...
relationship_bandwidth_limit 25
...
```

By default this is 25MBps: note that this is megabytes, not the megabits of the partnership configuration. This means that no matter how few relationships are synchronizing, the most synchronization I/O that will be generated per HyperSwap volume is 25MBps (technically, this is 25MBps of reads on the up-to-date copy, and 25MBps of writes on the other copy).

If your system has storage that cannot handle the additional 25MBps of IO, you can configure this to a lower value like this

```
chsystem –relationshipbandwidthlimit 10
```

If you want to accelerate synchronization when there aren’t many HyperSwap volumes synchronizing, you may want to increase it to a higher value:

```
chsystem –relationshipbandwidthlimit 200
```


2.1.7 Further details

Other than all nodes in an I/O group having to be in the same site, the requirements for the HyperSwap topology are very similar to those of the Enhanced Stretched Cluster topology. The Redbook "IBM SAN Volume Controller Enhanced Stretched Cluster with VMware", http://www.redbooks.ibm.com/redbooks/pdfs/sg248211.pdf, contains many details on the requirements of the Enhanced Stretched Cluster topology, including the quorum storage (particularly sections 3.3.4, 3.3.6, and 4.9-4.11), and the connectivity between sites (section 3.4).

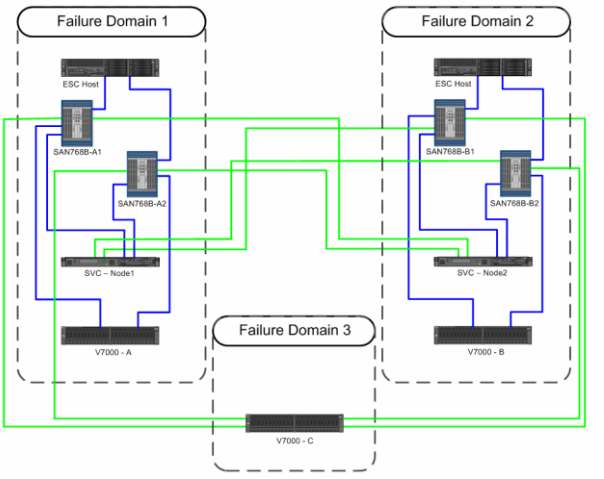

Figure 1 - SAN Volume Controller stretched cluster configuration (originally Figure 3-2 in http://www.redbooks.ibm.com/redbooks/pdfs/sg248211.pdf). Replacing each of Node1 and Node2 with an SVC or Storwize I/O group would give a valid HyperSwap topology.

2.2 Creating HyperSwap volumes

If you have set up the system topology correctly as in the previous section, there are four key steps in creating a HyperSwap volume:

- (optionally) Use mkrcconsistgrp to allow multiple HyperSwap volumes to copy consistently together
- Use mkvdisk to create the different vdisk objects required
- Use addvdiskaccess to allow the HyperSwap volume to be accessed on either site.
- Use mkrcrelationship to create an active-active relationship to coordinate replication
- Use chrcrelationship to associate change volumes with the HyperSwap volume

2.2.1 Making a consistency group

A consistency group may be created. This allows all HyperSwap volumes for a specific application to fail over together, ensuring that at least one site has an up-to-date copy of every volume for the application.

```
mkrcconsistgrp –name hsConsGrp0
```

If all HyperSwap volumes are standalone – that is, they each are fully independent and can operate individually, this step may be omitted.

2.2.2 Making the vdisks

2.2.2.1 New Master vdisk

Each HyperSwap volume needs a master vdisk. This vdisk can be created when required to hold new application data. It is also possible to use an existing vdisk, to add HyperSwap function to an existing application without impacting that application.

To create a new vdisk:

```
mkvdisk –name hsVol0Mas –size 1 –unit gb –iogrp 0 –mdiskgrp
siteA-ds8k -accessiogrp 0:1
```

This will be the vdisk that holds the initial copy of the data, which is then replicated to the other site. For a completely new HyperSwap volume, it doesn't matter which site this vdisk is created on.

What is important is that the site of the controllers (providing the mdisks in the storage pool) matches the site of the caching I/O group. That must also be true if you use an existing vdisk.

The master vdisk should be mapped to all hosts on both sites that will need access to the HyperSwap volume. You should ensure that the master vdisk is not written to until after the mkrcrelationship command is issued. One way to do this is to delay mapping the master vdisk to any host until after the active-active relationship has been created.

In this example mkvdisk command, most parameters are as normal and can be configured according to your needs. If you need the HyperSwap volume to be compressed or have particular EasyTier characteristics, specify that here. The –accessiogrp parameter is important, as it will allow the HyperSwap volume to be accessed on both sites. Specify the caching I/O groups that you will use for the auxiliary vdisk, as well as that which you have specified for the master vdisk.

2.2.2.2 Existing Master vdisk

If you're using an existing master vdisk, it will normally only have access through its own I/O group. To allow access to the HyperSwap volume through both sites, you will need to add access to the HyperSwap volume through the auxiliary vdisk's I/O group too.

```
addvdiskaccess –iogrp 1 hsVol0Mas
```

This part of the process is not policed, but must be completed in order for the HyperSwap volume to provide high availability through nodes on both sites. This step is only performed for the master vdisk.

2.2.2.3 Auxiliary vdisk

Next, whether an existing or new master vdisk is being used, we need an auxiliary vdisk for the HyperSwap volume. This must be the same size as the master vdisk, but using storage from the other site, and in an I/O group on the other site.

```
mkvdisk –name hsVol0Aux –size 1 –unit gb –iogrp 1 –mdiskgrp siteB-ds8k
```

Do not map the auxiliary vdisk to any hosts.

Normally, the master and auxiliary vdisks should be on similarly performing storage. If this is not possible, write performance will be dictated by the slower of the two, and read performance will be that of the vdisk currently acting as the master of the HyperSwap volume.

It is possible to use an auxiliary vdisk of a different provisioning type to the master vdisk, for example mixing a fully-allocated vdisk with a thin-provisioned vdisk, or compressed and non-compressed thin provisioned vdisks. This is not a recommended configuration.

2.2.2.4 Change volumes

Two thin-provisioned vdisks are also required to act as change volumes for this HyperSwap volume. These must be the same logical size as the master vdisk.

```
mkvdisk –name hsVol0MasCV –size 1 –unit gb –iogrp 0 –mdiskgrp
siteA-ds8k -rsize 0% -autoexpand
mkvdisk –name hsVol0AuxCV –size 1 –unit gb –iogrp 1 –mdiskgrp
siteB-ds8k -rsize 0% -autoexpand
```

One change volume is created with the same I/O group as the master vdisk, and a storage pool in the same site (not necessarily the same storage pool as the master vdisk). Another change volume is created in the auxiliary vdisk's I/O group, and a storage pool in the same site. The system will not police that the same pool is used for a change volumes as the master/aux vdisk, but future versions may police this.

Do not map the change volumes to any hosts.

The change volumes are used to store differences between the copies while resynchronizing the HyperSwap volume copies, and normally only require enough storage performance to satisfy the resynchronization rate. If access is enabled to the stale copy during resynchronization, as outlined in section 4 below, a portion of host reads and writes will be serviced by the change volume storage, but this will decrease towards zero within a short period of time.

This means that change volumes may be stored on lower-performing storage than the master and auxiliary vdisks without significantly impacting host I/O performance.

The change volumes will normally consume capacity equal to the initially-specified rsize. During resynchronization, the change volume at the stale copy will grow as it retains the data needed to revert to the stale image. It will grow to consume the same amount of storage as the quantity of changes between the two copies, thus a stale copy that needs 20% of its data changed to be synchronized with the up-to-date copy will have its change volume grow to consume 20% of its logical size. After resynchronization, the change volume will automatically shrink back to the initially-specified rsize.

2.2.3 Making the active-active relationship

This step is simply to add the HyperSwap volume's master and auxiliary vdisks to a new active-active relationship.

Active-active relationships are a special kind of relationship that can only be used in HyperSwap volumes. They can't be configured through the GUI, you can't manually start or stop them (other than in the disaster recovery scenarios outlined in section 4), and you can't convert them into Metro Mirror or Global Mirror relationships.

If the master disk already contains application data, then use the following command to create the relationship (note the system name in this example is “testCluster”):

```
mkrcrelationship –master hsVol0Mas –aux hsVol0Aux –cluster
testCluster -activeactive –name hsVol0Rel
```

At this point the auxiliary vdisk will go offline, as from now on it will only be accessed internally by the HyperSwap function. The master vdisk will remain online.

If the master vdisk has not been written to yet, use the –sync parameter to avoid the initial synchronization process:

```
mkrcrelationship –master hsVol0Mas –aux hsVol0Aux –cluster
testCluster -activeactive –name hsVol0Rel –sync
```

Note: You should not use the –nofmtdisk parameter of the mkvdisk command to disable the quick initialization of fully-allocated volume data for HyperSwap volumes. If it is necessary, ensure that the –sync parameter of the mkrcrelationship command is omitted so that the system fully synchronizes the two copies, even if neither has been written to.

If you are creating many HyperSwap volumes that you want to be part of a new consistency group, you should add the active-active relationships to the group as you create them. A relationship created with a –consistgrp parameter will be added into the specified consistency group as long as that consistency group has a state value of inconsistent_stopped if the –sync flag was omitted, or consistent_stopped if the –sync flag was provided. All the other relationships in that group must have been similarly created and have not had change volumes configured yet.

```
mkrcrelationship –master hsVol0Mas –aux hsVol0Aux –cluster
testCluster -activeactive –name hsVol0Rel –consistgrp hsConsGrp0
```

A relationship created with a –sync flag and with a –consistgrp parameter will be added into the specified consistency group as long as that consistency group has a state value of consistent_stopped, essentially meaning that all other relationships in that group have been similarly created and have not had change volumes configured yet.

```
mkrcrelationship –master hsVol0Mas –aux hsVol0Aux –cluster
testCluster -activeactive –name hsVol0Rel –consistgrp hsConsGrp0 -sync
```

See “Adding to a consistency group” later for details of how to add or remove an active-active relationship to or from a consistency group after the relationship has been created.

2.2.4 Adding the change volumes

Next, we add the two change volumes.

```
chrcrelationship –masterchange hsVol0MasCV hsVol0Rel
chrcrelationship –auxchange hsVol0AuxCV hsVol0Rel
```

At this point the active-active relationship will start replicating automatically. If the relationship was created *without* the –sync flag, perhaps because the master vdisk already had application data, then the relationship will synchronize the existing data from the master vdisk to the aux vdisk. This initial sync process does not use the change volumes.

The change volume will be used to enable automatic resynchronization after a link outage or other fault causes replication to stop. The HyperSwap function will internally join these two vdisks

together so they can both be accessed through the master vdisk's host maps, using whichever of the vdisks has an up-to-date copy of the data.

2.2.5 Adding to a consistency group

Active-active relationships are added to consistency groups in just the same way as Metro Mirror and Global Mirror relationships. This can either be done when the relationship is created, as mentioned previously, or at a point after the relationship has been created.

```
chrcrelationship –consistgrp hsConsGrp0 hsVol0Rel
```

You can't mix and match relationship types in a consistency group. When adding an active-active relationship to a consistency group, the group must either be empty or only contain active-active relationships.

You also can't mix relationships with different states. For HyperSwap, that means that you can only add a relationship to a consistency group if the relationship has a copy in each site of the same state as the consistency group. In turn, this means:

- The active-active relationship's state attribute must match the active-active consistency group's state attribute
- If the state is not consistent_synchronized, the site of the vdisk acting as the primary copy of the active-active relationship must be the same as the site of the vdisks acting as the primary copies of the relationships in the active-active consistency group. Further details on active-active relationship replication direction are given in section 3.2.
- If the state is consistent_synchronized, and the site of the primary vdisk of the active-active relationship is not the same as the primary site of the consistency group, the relationship will have its direction switched as it is added so the primary site matches.
- If the site of the master vdisk of the active-active relationship does not match the site of the master vdisks of the relationships in the consistency group, the role of the master and auxiliary vdisks in the active-active relationship are swapped. Host access will continue to be provided through the same vdisk ID and host maps, which will now be the auxiliary vdisk of the relationship. The relationship ID will be retained even though this will now match the auxiliary vdisk ID. The master and auxiliary roles will be restored if the relationship is removed from the consistency group.

If you need to remove that HyperSwap volume from the consistency group, so it can fail over independently, you can do so:

```
chrcrelationship –noconsistgrp hsVol0Rel
```

2.3 Converting from Stretched Cluster

Converting a system from a stretched system topology to a hyperswap system topology has some additional challenges. Note that this only applies to SVC, as Storwize systems cannot be configured with the stretched topology.

2.3.1 Node movement

The main layout change between the two system topologies is how I/O groups are configured. Consider a two-site environment with two nodes located at each site. With a stretched system topology, each site would contain one node from I/O group 0 and one node from I/O group 1. With a

hyperswap system topology, one site contains both nodes from I/O group 0, and the other contains both nodes from I/O group 1.

To convert between these topologies, you could physically move nodes around, so that the nodes forming an I/O group continue to form that I/O group in the new topology.

Alternatively, you can leave the node hardware in the same place, and just swap which I/O groups one node in site 1 and one node in site 2 are part of. **Warning:** this can result in data corruption if care is not taken as follows.

If you reassign nodes to different I/O groups, and hosts accessing volumes through that node will not necessarily be aware that the SCSI LUNs available through that node's ports refer to different volumes. The hosts may then write data to the wrong volume, corrupting that volume.

This risk can be removed by swapping the WWNNs of the two nodes being swapped. This should be done through the front panel for 2145-CG8 and earlier hardware, or with the "satask chvpd -wwnn" command on any hardware, and should happen after both nodes have been removed from their old I/O groups and not yet added to their new I/O groups. This step is best performed using the service assistant GUI.

It is likely that the zoning will need to be reconfigured after the WWNNs have been swapped.

2.3.2 Converting stretched volumes to HyperSwap volumes

If you want to convert a stretched volume to a HyperSwap volume, you can

- delete the volume copy on one site
- create a second volume on that site
- create an active-active relationship between the two volumes with the existing volume as the master volume
- create change volumes for the active-active relationship so the relationship can start replication

This loses HA capability during the initial synchronization of the relationship, and requires moving lots of data around unnecessarily.

If it's possible to quiesce host writes for a while, perhaps by shutting down host applications, HA can be retained, and the data can avoid being copied by

- quiescing host I/O
- running `splitvdiskcopy <stretchedvolume>` to convert the two volume copies into two independent volumes
- creating an active-active relationship between these two volumes with the existing volume as the master volume, providing the `-sync` flag to ensure that the data is not recopied
- unquiescing host I/O (can be done now or at the end)
- creating change volumes for the active-active relationship so the relationship can start replication

A second method to avoid resynchronization is to run


```
splitvdiskcopy –activeactive <stretchedvolume>
```

on the existing volume. This will convert the stretched volume to a HyperSwap volume while pausing I/O processing, and so this does not require shutting down or otherwise quiescing host I/O. Change volumes must still be created before this new HyperSwap volume will continue to replicate, which means that HA will be dropped until these change volumes are created and any changes made to the master vdisk replicated. The auxiliary vdisk will continue to provide disaster recovery capability during this period.

Note: the –activeactive option of the splitvdiskcopy command may not be available on the initial release of the HyperSwap function.

This section applies to conversion of stretched volumes created under either a standard or stretched topology, and so it can be used to avoid resynchronization when converting a system from using non-enhanced or Enhanced Stretched Cluster.

3 Using HyperSwap Volumes

3.1 Initial synchronization

When a HyperSwap volume has just been created and the change volumes haven't been configured yet, the relationship state and vdisks' state are:

```
IBM_2145:dizzy:superuser>lsrcrelationship 0
id 0
name rcrel0
master_cluster_id 0000020076805032
master_cluster_name dizzy
master_vdisk_id 0
master_vdisk_name vdisk0
aux_cluster_id 0000020076805032
aux_cluster_name dizzy
aux_vdisk_id 6
aux_vdisk_name vdisk6
primary master
consistency_group_id 0
consistency_group_name rccstgrp0
state inconsistent_stopped
bg_copy_priority 50
progress 0
freeze_time
status change_volumes_needed
sync
copy_type active_active
cycle_period_seconds 300
cycling_mode
master_change_vdisk_id
master_change_vdisk_name
aux_change_vdisk_id
aux_change_vdisk_name

IBM_2145:dizzy:superuser>lsvdisk
id name    IO_group_id IO_group_name status   mdisk_grp_id mdisk_grp_name
capacity type    FC_id FC_name RC_id RC_name vdisk_UID
fc_map_count copy_count fast_write_state se_copy_count RC_change
compressed_copy_count
```



```
0  vdisk0  0           io_grp0       online   0            mdiskgrp0
250.00MB striped               0     rcrel0
6005076801DA0140C800000000000000 0           1          empty            0
no        0
1  vdisk1  0           io_grp0       online   0            mdiskgrp0
250.00MB striped
6005076801DA0140C800000000000001 0           1          empty            0
no        0
6  vdisk6  1           io_grp1       offline  1            mdiskgrp1
250.00MB striped               0     rcrel0
6005076801DA0140C800000000000006 0           1          empty            0
no        0
7  vdisk7  1           io_grp1       online   1            mdiskgrp1
250.00MB striped
6005076801DA0140C800000000000007 0           1          empty            0
no        0

IBM_2145:dizzy:superuser>lsvdisk 6
id 6
name vdisk6
IO_group_id 1
IO_group_name io_grp1
status offline
...
copy_id 0
status online
```

Note the progress attribute of the relationship being 0, showing that the relationship hasn't copied any data, and the status attribute being "change_volumes_needed", showing that the lack of change volumes is impacting the ability to copy. The relationship state is inconsistent_stopped, which confirms that the relationship is not yet copying.

The auxiliary vdisk, vdisk6 is offline, and will remain offline. The vdisk copy object has a status of online, however, showing that the underlying storage is online. These object states will be further described later in this document.

Having configured change volumes, the relationship's state will be:

```
IBM_2145:dizzy:superuser>lsrcrelationship 0
id 0
name rcrel0
master_cluster_id 0000020076805032
master_cluster_name dizzy
master_vdisk_id 0
master_vdisk_name vdisk0
aux_cluster_id 0000020076805032
aux_cluster_name dizzy
aux_vdisk_id 6
aux_vdisk_name vdisk6
primary master
consistency_group_id 0
consistency_group_name rccstgrp0
state inconsistent_copying
bg_copy_priority 50
progress 8
freeze_time
status online
sync
copy_type active_active
```



```
cycle_period_seconds 300
cycling_mode
master_change_vdisk_id 1
master_change_vdisk_name vdisk1
aux_change_vdisk_id 7
aux_change_vdisk_name vdisk7
```

The status is online, the progress value is increasing, and the state has become inconsistent_copying.

The HyperSwap volume is synchronizing.The progress field will increase from 0 to 100, then the HyperSwap volume will show

```
IBM_2145:dizzy:superuser>lsrcrelationship 0
id 0
name rcrel0
master_cluster_id 0000020076805032
master_cluster_name dizzy
master_vdisk_id 0
master_vdisk_name vdisk0
aux_cluster_id 0000020076805032
aux_cluster_name dizzy
aux_vdisk_id 6
aux_vdisk_name vdisk6
primary master
consistency_group_id 0
consistency_group_name rccstgrp0
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type active_active
cycle_period_seconds 300
cycling_mode
master_change_vdisk_id 1
master_change_vdisk_name vdisk1
aux_change_vdisk_id 7
aux_change_vdisk_name vdisk7

IBM_2145:dizzy:superuser>lsvdisk
id name    IO_group_id IO_group_name status   mdisk_grp_id mdisk_grp_name
capacity type    FC_id FC_name RC_id RC_name vdisk_UID
fc_map_count copy_count fast_write_state se_copy_count RC_change
compressed_copy_count
0  vdisk0  0           io_grp0       online   0            mdiskgrp0
250.00MB striped many  many    0     rcrel0
6005076801DA0140C800000000000000 2           1          empty            0
no        0
1  vdisk1  0           io_grp0       online   0            mdiskgrp0
250.00MB striped many  many    0     rcrel0
6005076801DA0140C800000000000001 2           1          empty            0
yes       0
6  vdisk6  1           io grp1       offline  1            mdiskgrp1
250.00MB striped many  many    0     rcrel0
6005076801DA0140C800000000000006 2           1          empty            0
no        0
7  vdisk7  1           io_grp1       online   1            mdiskgrp1
250.00MB striped many  many    0     rcrel0
6005076801DA0140C800000000000007 2           1          empty            0
yes       0
```


With the relationship state finally consistent_synchronized, both copies are up-to-date, meaning that the HyperSwap volume is now highly available, and can survive either site failing. Note that lsvdisk doesn't show anything different here to signify that both copies are usable.

3.2 Relationship direction

You may have noticed that the active-active relationship has a "primary" attribute just like regular Metro Mirror and Global Mirror relationships. This will be set to either "master" or "aux". With an active-active relationship, the vdisk in one I/O group acts as the primary, supplying data for reads, and serializing writes. All reads and writes must be initially processed by that I/O group. This is the method by which writes are consistently applied to the volumes.

So that this doesn't add any extra delay to host reads and writes, the HyperSwap function will adjust which I/O group's vdisk should act as the primary to the one on the same site as the host or hosts performing the majority of I/O to that volume.

From an initially created HyperSwap volume, the master vdisk will act as the primary. If there is 10 minutes of more I/O being submitted to the auxiliary vdisk's site, the system will switch the direction of the relationship. From outside the system, the only visible effects of this are that the hosts on that site will have improved read and write performance, and the active-active relationship for that HyperSwap volume will have a primary attribute of "aux".

So after initial configuration of a HyperSwap volume, there may be a period of 10 minutes of increased traffic between the sites, but it will be resolved after that initial training period.

For a HyperSwap volume that has had I/O ongoing to the primary vdisk's site for some time, there has to be a period of 20 minutes of the majority of I/O being submitted to the secondary vdisk's site for the active-active relationship to swap directions.

HyperSwap volumes in consistency groups all switch direction together. So the direction that a set of active-active relationships in a consistency group will replicate will depend on which of the two sites has the majority of host I/O across all HyperSwap volumes.

Note that "the majority of I/O" is currently a comparison of number of sectors written to rather than a count of IOs. A 75% majority is required to switch to provide some hysteresis and prevent frequent alternating of direction.

VMware systems can share datastores between multiple virtualized hosts using a single SVC or Storwize volume. To minimize cross-site I/O traffic, make sure that a datastore is only used for virtual machines primarily running on a single site, as this will let HyperSwap orient the replication optimally.

3.3 Site failure

Normally, the storage and nodes on both sites will be online, and both copies of every HyperSwap volume (i.e. the master and auxiliary vdisk of every active-active relationship) will contain up to date data. If a site fails such that the Storwize system nodes, the storage, the Fibre Channel connectivity, or a combination are unavailable, through hardware failure, power failure, or site inaccessibility, HyperSwap will preserve access to volumes through the remaining site.

Starting with a fully synchronized HyperSwap volume (i.e. one with an active-active relationship with a state of “consistent_synchronized”), if the storage or nodes for the vdisk on one site goes offline:

- the state of the active-active relationship will become consistent_copying
- host I/O will pause for less than a second in a normal case (this can extend to multiple seconds in some cases, particularly with larger consistency groups)
- if the offline vdisk was the primary copy, the direction of the relationship will switch to make the online copy the primary
- the progress value of the active-active relationship will count down from 100% as the copies become more different (for example, if 10% of the HyperSwap volume was modified while one copy was offline, the progress value will show 90)
- the master vdisk will remain online and the auxiliary vdisk will remain offline when viewed through lsvdisk, regardless of which copy is no longer accessible

When that offline copy is restored, the progress value will count back up to 100 as the HyperSwap volume is resynchronized. Once it has been resynchronized the state of the active-active relationship will become consistent_synchronized once more.

No manual actions are required to make this process occur.

For HyperSwap volumes in consistency groups, a single volume with an inaccessible copy will cause this error recovery procedure to take place on every volume in the consistency group. Each active-active relationship will become consistent_copying. If one HyperSwap volume in a group already has one copy offline, and then a different HyperSwap volume in the same group has a copy on the other site go offline, HyperSwap will not be able to hide the second offline from host I/O activity, and that second HyperSwap volume will go offline.

3.4 Deleting HyperSwap volumes

To delete a HyperSwap volume containing data that is no longer required, simply delete each vdisk created for the HyperSwap volume, using the –force option to rmvdisk. Alternatively, each created object may be individually deleted or deconfigured in the reverse order to how they were created and configured, which will avoid the need for the –force option to rmvdisk.

If the data will still be required after deconfiguring the HyperSwap object, use the procedures described in section 5.

Once every HyperSwap volume has been removed or converted to a single copy volume, the topology of the system can be reverted to the standard topology.

3.5 FlashCopy with HyperSwap

FlashCopy can be used to take point-in-time copies of HyperSwap volumes.

A FlashCopy map with a HyperSwap volume as its source may not cross sites. This means that a FlashCopy mapping where the target volume is on site 1 must use the vdisk of the HyperSwap volume on site 1 as its source, and likewise for site 2. It’s not possible for a FlashCopy map with a HyperSwap volume as its source to copy data between sites.

For example, if a HyperSwap volume has vdisk 10 providing data on site 1, and vdisk 11 on site 2, FlashCopy maps can be created as follows:

```
mkfcmap –source 10 –target 12 ...
mkfcmap –source 11 –target 13 ...
```

where volumes 12 and 13 are single copy volumes already created on sites 1 and 2 respectively. These two FlashCopy maps can both be used independently to take point-in-time copies of the HyperSwap volume on the two sites. The system provides no coordination of these maps.

When triggering the FlashCopy map, the copy of the HyperSwap volume on the same site as the FlashCopy target volume must be either

- a primary copy of an active-active relationship in any state
- a secondary copy of an active-active relationship in consistent_synchronized

If access has been enabled to an old but consistent copy of the HyperSwap volume, a FlashCopy map may only be triggered on the site that contains that copy.

A FlashCopy map may not be created with a HyperSwap volume as its target. If necessary, delete the active-active relationship to convert the HyperSwap volume to a regular volume before creating and triggering the FlashCopy map.

4 Disaster Recovery with HyperSwap

The HyperSwap function will automatically use both copies to provide continuous host access to data, providing that both copies are up to date. If one copy is up to date, and the other is stale, and the up-to-date copy goes offline, the system cannot automatically use the remaining copy to provide high availability to the volume.

However, the user can choose to enable access to that stale copy. This is telling the system to rewind the state of that volume to the point in time of that stale copy.

As this is a step-wise change to the volume state, care needs to be taken with this process. Don’t do it if any data or state from the volume is cached in host systems, and ideally shut down host systems using the volume before taking these steps. Running these commands without these precautions will almost certainly crash your applications and corrupt the stale copy.

Let’s go back to the example relationship from earlier, during a resynchronization between sites:

```
IBM_2145:dizzy:superuser>lsrcrelationship 0
id 0
name rcrel0
master_cluster_id 0000020076805032
master_cluster_name dizzy
master_vdisk_id 0
master_vdisk_name vdisk0
aux_cluster_id 0000020076805032
aux_cluster_name dizzy
aux_vdisk_id 6
```



```
aux_vdisk_name vdisk6
primary master
consistency_group_id 0
consistency_group_name rccstgrp0
state consistent_copying
bg_copy_priority 50
progress 81
freeze_time 2015/03/04/12/16/47
status online
sync out_of_sync
copy_type active_active
cycle_period_seconds 300
cycling_mode
master_change_vdisk_id 1
master_change_vdisk_name vdisk1
aux_change_vdisk_id 7
aux_change_vdisk_name vdisk7
```

Here, the site of the secondary copy had previously been offline, had returned online, and the HyperSwap volumes are still resynchronizing. The consistent_copying state of the volume shows a resynchronization where the secondary copy contains a stale image, and the value contained in the freeze_time field shows when that image dates from. The progress value is increasing towards 100 as the resynchronization process continues.

Now, the site of the primary copy goes offline:

```
IBM_2145:dizzy:superuser>lsrcrelationship 0
...
state consistent_copying
bg_copy_priority 50
progress 83
freeze_time 2015/03/04/12/16/47
status primary_offline
sync out_of_sync
...

IBM_2145:dizzy:superuser>lsvdisk
id name    IO_group_id IO_group_name status   mdisk_grp_id mdisk_grp_name
capacity type    FC_id FC_name RC_id RC_name vdisk_UID
fc_map_count copy_count fast_write_state se_copy_count RC_change
compressed_copy_count
0  vdisk0  0           io_grp0       offline   0            mdiskgrp0
250.00MB striped many  many    0     rcrel0
6005076801DA0140C800000000000000 2           1          empty            0
no        0
```

With the only up-to-date copy of the volume offline, the active-active relationship can't switch direction to keep the HyperSwap volume online, so the master vdisk is now offline.

At this point, you look at the freeze_time value. If data from that date is not useful, for example it is from too long ago, or before a recent vital update, it may be best to wait until the offline up-to-date copy of the volume can be brought back online. However, if the stale data is useful, and it's likely that the up-to-date copy of the volume will remain online for an extended period of time, you can choose to enable access to the stale copy of the volume. Before running this command, make sure that no data or state from this volume is cached on host systems.

```
stoprcrelationship –access rcrel0
```


At this point, the data presented to hosts from this volume immediately changes to that stored on the stale copy. One way to think of this is that the volume has been consistently rolled back to the point in time denoted by the freeze_time value.

The volume continues to be readable and writable at this point. You can bring up your business applications again, and continue from this stale image.

Replication is paused, even if the up-to-date copy becomes online again. This is because the previously-stale image, which is now being accessed by hosts, and the previously up-to-date copy, which contains some changes not present on the previously-stale image, are now divergent copies. The two copies were the same at the freeze_time point in time, but then each had different writes applied. Either copy might be the one that the user wants to keep in the long term.

So the system allows the user to choose which copy is more useful to them. This choice will be made based on how much data was missing on the stale copy compared to the up-to-date copy, and how much progress has been made on the stale copy since access was enabled to it.

The first step is determining which copy has the stale copy, which is currently accessible to hosts. This will be either the master or auxiliary copy, and is visible under the “primary” attribute of the active-active relationship.

Next, consider which copy you want to retain. The next two sections consider the options.

4.1 Keep using the copy hosts are currently accessing; discard the old “up-to-date copy”

The disaster recovery using the stale copy has been a success. More than that, you’ve made useful business progress using that copy, and you value that more than any data the old up-to-date copy has that the copy you’re using doesn’t.

Or maybe there’s so little difference between the two copies that you might as well continue using what you’ve got. Switching back to the up-to-date copy is a disruptive operation for hosts, and you don’t want to have that impact.

So you’re keeping the stale copy, and discarding the up-to-date copy. Here’s the command to do that:

```
startrcrelationship -primary <current_primary> -force <relationship>
```

where <current_primary> is the current primary value of the active-active relationship, and will be “master” or “aux”. The “-force” flag is there because once you make your decision, that loses the ability to use the copy that’s not the primary, so it’s telling the system that you’re aware that this can’t be reverted. In our example, that would be

```
startrcrelationship -primary aux -force 0
```

There’s no need to quiesce host I/O or take any further action: this command will resume HyperSwap replication, and copy across any regions that are different between the two copies to resynchronize as fast as possible. Both copies keep a bitmap of volume regions at a 256kB granularity, used to record writes to that copy that haven’t yet been replicated to the other copy.

On this resynchronization, we use both sets of information to undo writes only applied to the old up-to-date copy, and also to copy across additional writes made to the stale copy during the disaster recovery. As the disaster recovery only happened because the copies were resynchronizing before the up-to-date copy went offline, all differences from that interrupted resynchronization process will be reverted on the old up-to-date copy now as well.

The active-active relationship goes into an inconsistent_copying state, and as copying continues, the progress increases towards 100. At that point, the relationship goes into a consistent_synchronized state, showing that both copies are up-to-date, and high-availability is restored:

```
IBM_2145:dizzy:superuser>lsrcrelationship 0
id 0
name rcrel0
master_cluster_id 0000020076805032
master_cluster_name dizzy
master_vdisk_id 0
master_vdisk_name vdisk0
aux_cluster_id 0000020076805032
aux_cluster_name dizzy
aux_vdisk_id 6
aux_vdisk_name vdisk6
primary aux
consistency_group_id 0
consistency_group_name rccstgrp0
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type active_active
cycle_period_seconds 300
cycling_mode
master_change_vdisk_id 1
master_change_vdisk_name vdisk1
aux_change_vdisk_id 7
aux_change_vdisk_name vdisk7
```

4.2 Go back to the up-to-date copy; discard the stale copy used for disaster recovery

The last section looked at what to do if you want the stale copy used for disaster recovery to be synchronized to the other copy and remain the data used in the long term. But what if you just want to go back to the up-to-date copy, the one that held the latest data before the disaster recovery? Maybe the stale data turned out to not hold useful data, or the outage of the up-to-date copy was shorter than you expected.

This scenario is different to the last one, as the image visible by hosts is going to change again. Just as enabling access to the stale copy required hosts to have no cached data from the volume (and ideally they should be fully shut down), the same is true of reverting back to the up-to-date copy. So before going further, make sure no hosts are going to be surprised by the data changing, and have no stale data that they might corrupt the up-to-date copy with.

Run


```
startrcrelationship -primary <current_secondary> -force <relationship>
```

where <current_secondary> is the copy other than the current primary value of the active-active relationship, and will be “master” or “aux”. In other words, if the primary field says “master”, user “aux” here, and vice versa. As before, you can’t get back to the other set of data once you’ve run this command, and the “-force” flag is there to acknowledge this. In our example, that would be

```
startrcrelationship -primary master -force 0
```

The image visible to hosts instantly reverts back to the up-to-date copy, so as soon as you’ve run this command, feel free to bring your hosts back online and start using this volume again.

As with the other scenario, the active-active relationship will be in an inconsistent_copying state while resynchronization, and once more this resynchronization uses the bitmaps of writes to each copy to accelerate this resynchronization process. Once the copies are fully synchronized, the relationship will go back to a consistent_synchronized state as high availability is restored for the volume:

```
IBM_2145:dizzy:superuser>lsrcrelationship 0
id 0
name rcrel0
master_cluster_id 0000020076805032
master_cluster_name dizzy
master_vdisk_id 0
master_vdisk_name vdisk0
aux_cluster_id 0000020076805032
aux_cluster_name dizzy
aux_vdisk_id 6
aux_vdisk_name vdisk6
primary master
consistency_group_id 0
consistency_group_name rccstgrp0
state consistent_synchronized
bg_copy_priority 50
progress
freeze_time
status online
sync
copy_type active_active
cycle_period_seconds 300
cycling_mode
master_change_vdisk_id 1
master_change_vdisk_name vdisk1
aux_change_vdisk_id 7
aux_change_vdisk_name vdisk7
```

4.3 Disaster Recovery with Volume Groups

All the descriptions above about enabling access to a stale copy of a HyperSwap volume also apply to HyperSwap volume groups, i.e. multiple HyperSwap volumes where the active-active relationships are contained in a single consistency group.

If during resynchronization, any of the up-to-date copies of volumes in a volume group is offline or unavailable (typically all would be offline in a disaster), you can choose to enable access to the stale copy of every volume in the volume group. As the HyperSwap function links replication and failover across HyperSwap volumes in a volume group, it’s guaranteed that during resynchronization, all

copies on one site will have a stale consistent copy of data, captured at an identical point in time, ideal for disaster recovery.

The “consistgrp” version of the commands above is used:

```
stoprcconsistgrp –access <consistency_group>
```

to gain access to the stale copies; then either

```
startrcconsistgrp –primary <current_primary> -force <consistency_group>
```

to retain the stale disaster recovery copies currently visible to hosts, and resume HyperSwap replication; or

```
startrcconsistgrp –primary <current_secondary> -force <consistency_group>
```

to revert to the previous up-to-date copy.

4.4 The overridequorum command

Version 7.2 of the SVC software introduced an overridequorum command, used to override the tiebreaking performed by the system quorum if it left the system in an unusable state.

One scenario where this could be useful is if a rolling disaster first broke the link between between the two sites, resulting in the quorum deciding which site’s nodes should be allowed to continue. Next, the rolling disaster impacts the chosen site’s nodes, taking them offline. The entire system is unusable at this point, because of how the tiebreak was resolved.

When the overridequorum command is issued

```
satask overridequorum -force
```

on a node showing a 551 or 921 error, that site’s nodes use their cluster state to form a new cluster (with a new cluster ID) based on the system state at the point that the tiebreak stopped that site’s nodes from taking part in the cluster.

Other than the new cluster ID, this gives the appearance of reverting the system state to the point in time of that tiebreak, and because the restored nodes have system cache and local volume copies matching that point in time, the volume state is reverted to that point in time as well. In typical scenarios where this command is used, little data and configuration will have changed since the tiebreak, so little will be lost by reverting the state.

There’s no specific interaction between HyperSwap volumes and the overridequorum command: if a volume copy local to the site brought online by the overridequorum command was up-to-date at the time of the lost tiebreak, it will immediately come online after the overridequorum command is run. On the other hand, if the copy was stale at the time of the lost tiebreak, access will need to be enabled to it with the stoprcrelationship command.

This command also removes the nodes in the other site. This means that volume copies on that site will need to be deleted and recreated. For this initial release of the HyperSwap function, this is done by deleting the active-active relationship, and whichever out of the vdisks are in the I/O groups that now have no nodes. The next section has more details about deconfiguring HyperSwap.

Once the HyperSwap volumes are converted to single-copy volumes on the online site, and the nodes in the other site have been re-added to the system, you can then convert the single-copy volumes back to HyperSwap volumes as in section 2.2.

4.5 HyperSwap Failure scenarios

Failure scenario	HyperSwap system behavior	Server and application impact
Single switch failure.	System continues to operate by using an alternate path in the same failure domain to the same node.	None.
Slow read or write performance to a copy (giving greater than 30 seconds response time)	System temporarily stops replicating to slow copy, and resynchronizes after 5 minutes.	None.
Single data storage failure.	System continues to operate by using the other data copy.	None.
Single quorum storage failure on site 3.	System continues to operate using alternative storage at site 3.	None.
Failure of either site 1 or 2	System continues to operate on the remaining site	Servers without high availability (HA) functions in the failed site stop. Servers in the other site continue to operate. Servers with HA software functions are restarted from the HA software. The same disks are seen with the same UIDs in the surviving site, and continue to offer similar read and write performance as before the disaster.
Failure of site 3, containing the active quorum disk	System continues to operate on both sites 1 and 2, selecting a quorum disk from sites 1 and 2 to allow I/O processing to continue	None.
Access loss between sites 1 and 2	System continues to operate the site that wins the quorum race. The cluster continues with operation, while the nodes in the other site stop, waiting for connectivity between sites 1 and 2 to be restored.	Servers without HA functions in the failed site stop. Servers in the other site continue to operate. Servers with HA software functions are restarted from the HA software. The same disks are seen with the same UIDs in the surviving site, and continue to offer similar read and write performance as before the disaster.
Access loss between sites 1 and 2 because of a rolling disaster. One site is down, and the other is still working. Later, the working site also goes down because	System continues to operate the site that wins the quorum race. The system continues with operation until the other site goes down. Even if the first site to go down comes back up, the whole system is considered	The system can restart using just the site that initially lost the quorum race, by using the overidequorum command. The volumes revert to the state they were at then that site lost the quorum race. Servers must be stopped before

of the rolling disaster	offline until the site that won the quorum race comes back up.	issuing this command, and restarted with the reverted state. Full read and write performance is given.

5 Deconfiguring HyperSwap

If you want to get rid of HyperSwap, there are a number of options available, depending on what your goal.

5.1 Removing HyperSwap volumes completely

If you don't need any data on a HyperSwap volume, and you just want to delete all objects related to it, simply delete all four vdisks associated with the HyperSwap volume: the master vdisk, the auxiliary vdisk, and the two change volumes. As the vdisks are deleted, the active-active relationship and any host maps will be deleted automatically.

```
rmvdisk hsVol0Mas
rmvdisk hsVol0Aux
rmvdisk hsVol0MasCV
rmvdisk hsVol0AuxCV
```

This can be done on as many or as few volumes as you want.

5.2 Converting to single-copy volumes, retaining access through the master vdisk

If you want to go back to using single-copy volumes, you need to decide which copy should be retained. If the master vdisk holds the copy to be retained, this conversion process is simple.

Delete the auxiliary vdisk and the two change volumes. The active-active relationship will be automatically deleted, and you'll just be left with a single-copy volume. From outside the system, there will be no visible change, as hosts will still have access to their data.

```
rmvdisk hsVol0Aux
rmvdisk hsVol0MasCV
rmvdisk hsVol0AuxCV
```

This can be done on as many or as few volumes as you want. Remember that if the deleted volume was part of a volume group, the remaining HyperSwap volumes in the volume group will not consider this single-copy volume in their failover and synchronization. This means that you would not normally do this conversion for a subset of the volumes supporting a specific application.

This will give you access to whatever data is currently on the master vdisk, so it should normally only be done if the master copy is up-to-date. This can be seen by the active-active relationship either having a primary value of "master", or a state value of "consistent_synchronized".

5.3 Converting to single-copy volumes, retaining access through the auxiliary vdisk

If you need to retain the auxiliary vdisk, there are two possible procedures.

First, you must quiesce any host using this volume. By deleting the master vdisk and the two change volumes, the active-active relationship will be automatically deleted, and you'll just be left with a

single-copy volume. Host maps will be deleted as the master vdisk is deleted, and new host maps must be created from the remaining auxiliary vdisk.

```
rmvdisk hsVol0Mas
rmvdisk hsVol0MasCV
rmvdisk hsVol0AuxCV
```

As with the previous case, this can be done on as many or as few volumes as you want.

This will give you access to whatever data is currently on the auxiliary vdisk, so it should normally only be done if the master copy is up-to-date. This can be seen by the active-active relationship either having a primary value of “master”, or a state value of “consistent_synchronized”.

You will then have to remap these auxiliary vdisks to the host systems, as the existing volume host maps were deleted with the master vdisks. Finally you can redetect volumes on the host systems, reconfigure them to use the auxiliary vdisks for I/O, then resume host I/O.

An alternative procedure is to delete the master vdisk with the command

```
rmvdisk –keepaux <mastervdisk>.
```

which may be run with host I/O running. This deletes the master vdisk’s storage, and replaces it with the auxiliary vdisk’s storage, preserving the master vdisk ID, the master vdisk host maps, and the auxiliary vdisk storage. This also deletes the active-active relationship. Finally, delete the change volumes, which are not deleted as part of the previous step.

```
rmvdisk hsVol0MasCV
rmvdisk hsVol0AuxCV
```

This allows a clean-up of failed master storage without impacting host I/O access, potentially as part of replacing the master vdisk’s storage.

Note: the `mkvdisk –keepaux` command may not be available on the initial release of the HyperSwap function.

5.4 Converting to system topologies other than “hyperswap”

Once all active-active relationships have been deleted, the system topology can be changed to “standard”. Aspects of the system locked down in the HyperSwap system topology, for example node and controller sites, can then be changed.

If the goal is to ultimately get the system to the “stretched” system topology, you’ll need to reconfigure the nodes while in the standard topology.

6 Summary of interesting object states for HyperSwap

6.1 lsvdisk

- **status** shown for HyperSwap volume master vdisk in lsvdisk shows whether hosts are able to access data, i.e. whether whole HyperSwap object has access to up-to-date data or not, not whether master vdisk itself is actually online. status for auxiliary vdisk will always be offline.

Note that running lsvdisk on a specific vdisk to get detailed information will also show the vdisk copy status value (see the next section)

- **RC_id** and **RC_name** attributes for both master and auxiliary vdisks will show the active-active relationship supporting the HyperSwap volume

6.2 lsvdiskcopy

- **status** for vdisk copies supporting HyperSwap volumes will show whether the underlying storage is online or not.

6.3 lsrcrelationship or lsrcconsistgrp

These values are for stand-alone relationships seen with lsrcrelationship. Relationships in a consistency group must all share the same state, primary, and freeze_time field values, so will change value based on the condition of all the relationships in that consistency group. The consistency group itself will show the same values when queried through lsrcconsistgrp.

- **state** value is the key attribute that tells you what copying the HyperSwap volume is doing:
  - **inconsistent_stopped** – this HyperSwap volume only has useful data on the master vdisk, and the relationship's change volumes are not both configured yet
  - **consistent_stopped** – this HyperSwap volume only has useful data on the master vdisk, the relationship's change volumes are not both configured yet, but the relationship was created with –sync, limiting needed copying to only the data written to on the master volume since the active-active relationship was created
  - **inconsistent_copying** – this HyperSwap volume only has useful data on the master vdisk, but it is correctly configured and is performing initial synchronization
  - **consistent_synchronized** – this HyperSwap volume is correctly configured, has up-to-date data on both vdisks, and is highly available to hosts (if addvdiskaccess has been run correctly)
  - **consistent_copying** – this HyperSwap volume is correctly configured, has had or is currently having a period of inaccessibility of one vdisk, leaving that vdisk consistent but stale (the freeze_time attribute will show when that stale data dates from); access to that data can be provided in a disaster with the stoprcrelationship –access command (not recommended in beta); resynchronization automatically will take place when possible
  - **idling** – this HyperSwap volume is correctly configured, and has had access enabled to a stale but consistent copy by running stoprcrelationship –access when the active-acitve relationship was in a state of consistent_copying (this is not recommended in beta); synchronization is paused, and can be resumed by running startrcrelationship –primary (master|aux) according to direction relationship should resynchronize in
- **primary** will be "master" or "auxiliary", and tells you which copy is acting as the primary at the moment, and thus which I/O group is primarily processing IOs for this HyperSwap volume
- **status** will show "online" if all needed vdisks are online and are able to synchronize, or the reason why synchronization is not possible: "primary_offline", "secondary_offline", "primary_change_offline", or "secondary_change_offline" if one of the vdisks of the

HyperSwap volume is offline; or “change_volumes_needed” if the HyperSwap volume does not have both change volumes configured

- **progress** shows how similar the two copies are as a percentage, rounded down to the nearest percent: during resynchronization this will count up to 100 as the HyperSwap volume nears being synchronized
- **freeze_time** shows at what point the data is frozen on a stale but consistent copy when the relationship has a state of consistent_copying, allowing the user to decide if there is value in using the data (with the stoprcrelationship –access command) if the up-to-date copy goes offline

