IBM

Redpaper

Werner Bauer
Bertrand Dufrasne
Richard Muerke

IBM z/OS Metro/Global Mirror Incremental Resync

This IBM® Redpaper publication describes the Incremental Resynchronization (IR) enhancements to the z/OS® Global Mirror (zGM) copy function when used in conjunction with Metro Mirror in a multiple target three-site solution.

This three-site mirroring solution is often simply called *MzGM*. zGM was previously known as *Extended Remote Copy (XRC)*. The new name is, as the title for this paper indicates, *z/OS Metro/Global Mirror Incremental Resync* or, for short, *RMZ Resync*.

Note: In the remainder of this paper, we refer to z/OS Metro/Global Mirror Incremental Resync as RMZ Resync.

The paper is divided into the following sections:

- Three-site copy services configurations

  We briefly revisit the three-site solutions that are supported by the IBM DS8000™ in an IBM z/OS environment. These solutions are respectively designated a Metro/Global Mirror and a z/OS Metro/Global Mirror.

- IBM HyperSwap™ technology

  We briefly describe the differences between HyperSwap and Basic HyperSwap.

- z/OS Metro/Global Mirror with incremental resync support

  This section, the core of the paper, outlines the improvements brought to z/OS Metro/Global Mirror in the context of Metro Mirror and HyperSwap.

Note: The RMZ Resync function is available only under GDPS/RMZ Resync. It is not available with any TPC for Replication (TPC-R) Basic HyperSwap function as of the writing this paper. It is also *not* sufficient to implement GDPS/PPRC HyperSwap only and add an XRC configuration.

Three-site Copy Services configuration

Many companies and businesses require their applications to be continuously available and cannot tolerate any service interruption. High availability requirements have become as important as the need to quickly recover after a disaster affecting a data processing center and its services.

Over the past years, IT service providers, either internal or external to a company, exploited the capabilities of new developments in disk storage subsystems. Solutions have evolved from a two-site approach to include a third site, thus transforming the relationship between the primary and secondary site into a high availability configuration.

IBM addresses these needs and provides the option to exploit the unique capabilities of the DS8000 with a firmware-based three-site solution, Metro/Global Mirror. Metro/Global Mirror is a cascaded solution. All Copy Services functions are handled within the DS8000 firmware. Metro/Global Mirror also provides incremental resync support when the site located in between fails, as discussed in "Metro/Global Mirror configuration" on page 2.

In contrast to Metro/Global Mirror, Metro Mirror/z/OS Global Mirror or MzGM is a combination of DS8000 firmware and z/OS server software. Conceptually MzGM is a multiple target configuration with two copy relationships off the same primary or source device. More details follow in "Metro/zOS Global Mirror" on page 3.

Being entirely driven by the DS8000 firmware, Metro/Global Mirror offers an advantage over MzGM in that it is independent from application hosts and volume specifics such as CKD volumes or FB volumes. Metro/Global Mirror handles all volumes without any knowledge of their host connectivity. MzGM through its zGM component is dependent on z/OS. Consequently, it can handle only CKD volumes and requires time-stamped write I/Os.

The benefits of zGM when compared to GM include the requirement for less disk space (zGM requires only a single copy of the database at the remote site, compared to two copies with GM) and greater scalability (zGM can handle much larger configurations than GM).

The "three-site configuration" terminology can also mean simply three copies of the data volumes. The differentiation between "site" and "copy" basically refers to three copies, and each copy might reside at a distinct site. Metro/Global Mirror or MzGM often have both Metro Mirror volumes located at the same site, and only the third copy is located at a second site.

Metro/Global Mirror configuration

Metro/Global Mirror is a cascaded approach, and the data is replicated in a row from one volume to the next. Figure 1 on page 3 shows the Metro/Global Mirror cascaded approach. The replication from Volume *A* to Volume *B* is synchronous, while the replication from Volume *B* to Volume *C* is asynchronous.

Volume *B* is the cascaded volume, and it has the following two roles:

1. Volume *B* is the secondary or target volume for volume *A*.
2. At the same time, volume *B* is also the primary or source volume for volume *C*.

Combining Metro Mirror and continue with Global Mirror places the middle volume in a configuration that is known as *cascaded*.

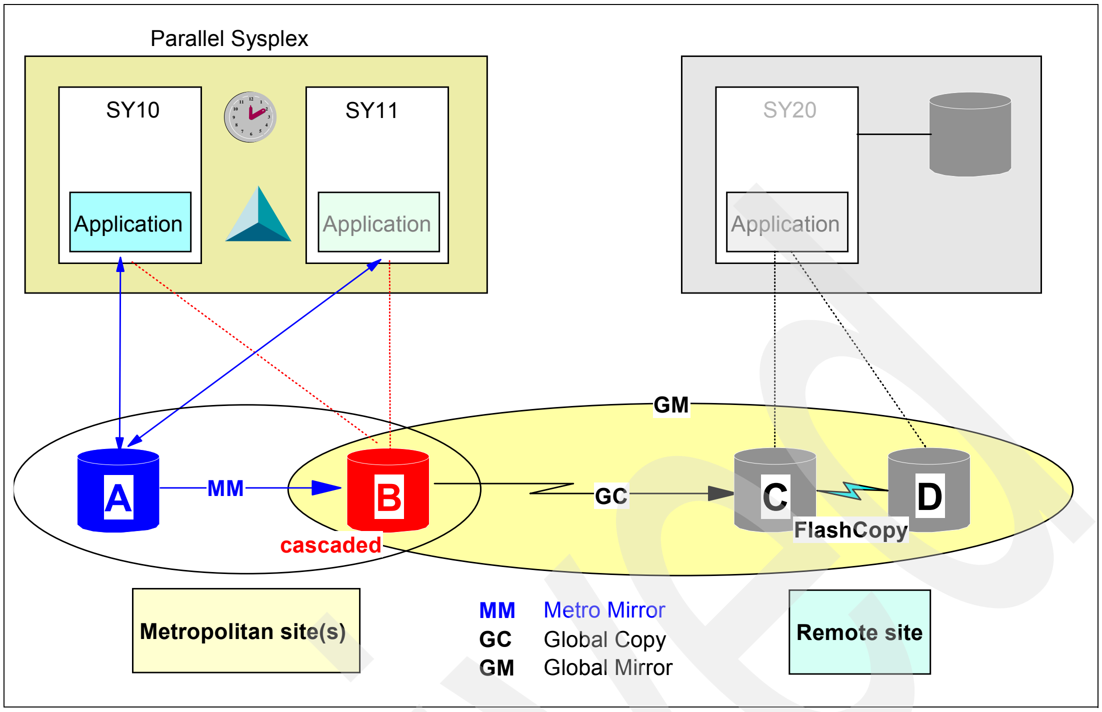


Figure 1 Three-site cascaded configuration

The Metro/Global Mirror function is provided and driven solely by the storage subsystem firmware. During normal operations, no software is involved in driving this configuration. This situation applies especially to the Global Mirror relationship where the firmware autonomically manages the formation of consistency groups and thus guarantees consistent data volumes at all times at the remote site. However, to handle exceptions and failure situations, a (software) framework must be wrapped around a Metro/Global Mirror configuration. The following alternatives manage such a cascaded Metro/Global Mirror configuration, including planned and unplanned failovers:

- GDPS/PPRC and GDPS/GM
- TPC-R
- DSCLI and REXX™ scripts

This software framework is required to handle planned or unplanned events in either site.

IBM GDPS® also manages an incremental resync when site B fails and A needs to connect incrementally to C/D and to continue with Global Mirror (GM) from A to C/D.

Metro/Global Mirror management through GDPS also includes implementing HyperSwap when A fails or a planned swap is scheduled to swap all I/Os from A to B. The swap is transparent to the application I/Os, and GM continues to replicate the data from B to C/D.

We do not discuss this Metro/Global Mirror configuration any further in this paper. For more information, refer to *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374.

Metro/zOS Global Mirror

Metro/Global Mirror is applicable to all volume types in the DS8000, including Fixed Block (FB) for all open systems attached volumes as well as for Count-Key-Data (CKD) volumes attached to System z® operating systems such as z/OS or z/VM®. The HyperSwap function is applicable to CKD volumes managed by z/OS. In addition to z/OS volumes, GDPS controlled HyperSwap activities serve Linux® volumes under native Linux or Linux as z/VM guest.

MzGM is a three-site multiple target configuration in a System z environment. MzGM combines firmware in the DS8000 with software in z/OS.[1] Software support is provided in z/OS through the System Data Mover (SDM) component, which is supported by current z/OS release levels.

Figure 2 shows a MzGM configuration.

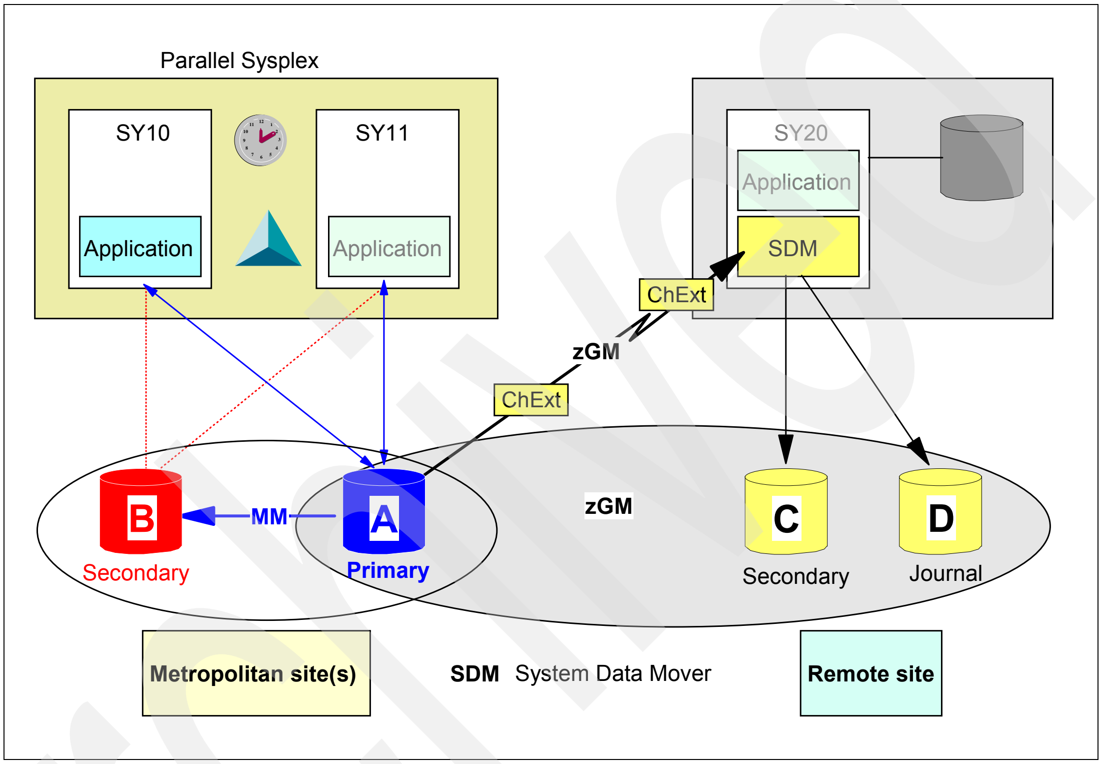


Figure 2 Three-site multiple target configuration, MzGM

In Figure 2 the A volume or primary volume is located in the middle position of the A, B, C, and D volume set.

A is the active application volume with the following two copy services relationships:

1. Metro Mirror relationship between A and B

   Volumes A and B are in a synchronous copy services relationship and are the base for potentially swapping both devices through HyperSwap. This topic is discussed later in "z/OS Metro/Global Mirror" on page 17.

2. zGM relationship between A and C

   This relationship is managed by the SDM. An associated zGM session contains all A and C volumes, with all A volumes that are also in a Metro Mirror relationship with B.

[1] IBM DS8000 firmware level (LIC) V3.1 or later

The following two points describe how GDPS/MzGM must be used to establish and manage both sessions:

- GDPS/PPRC for the Metro Mirror configuration between *A* and *B*
- GDPS/XRC for zGM and all *A* and *C* volumes.

  The journal or *D* volumes are used by SDM only.

The number of *D* volumes depends on the number of zGM volume pairs and the amount of data that is replicated from *A* to *C* because all replicated data is written twice in the remote location:

1. Writes to SDM journal data sets on these *D* volumes. We recommend you dedicate the *D* volumes to SDM journals only. SDM Journal volumes usually number 4 to 16, but with the HyperPAV support by SDM, these journals might be consolidated onto fewer *D* volumes. You must ensure that sufficient alias devices are available. Allocate journals as striped data sets across two devices. For details on zGM setup and configuration, refer to *z/OS V1R10.0 DFSMS Advanced Copy Services*, SC35-0428.
2. Apply all writes on the secondary in the same sequence as they occur on the primary volumes.

With the advent of HyperSwap with the Metro Mirror configuration between Volume *A* and *B* as shown in Figure 2 on page 4, zGM loses its active primary volume (volume *A*) because of the transparent swap (HyperSwap). Thus, you must create a new zGM session and perform a full initial copy between volume *A* and volume *C*.

This process has been enhanced with MzGM/IR support and is described in the section “z/OS Metro/Global Mirror” on page 17.

HyperSwap technology

Because z/OS Metro/Global Mirror Incremental Resync relies on z/OS HyperSwap services, this section includes an overview of the z/OS HyperSwap function. It covers two currently available alternatives for interfacing with the HyperSwap service that z/OS provides.

The following HyperSwap interfaces alternatives are available:

- GDPS/PPRC HyperSwap Manager or the full GDPS/PPRC HyperSwap function

  HyperSwap Manager provides only the disk management function with single sites or multiple site configurations. The full function GDPS/PPRC HyperSwap solution also includes the server, workload, and coordinated network switching management.
- TPC for Replication (TPC-R) provides Basic HyperSwap. Basic HyperSwap and some implementation details are covered in a separate paper, *DS8000 and z/OS Basic Hyperswap*, REDP-4441.

Another goal of this section is to clarify the current purpose, capabilities, and limitations of these two management frameworks, which utilize the z/OS Basic HyperSwap services within their individual functions. This section also can help you understand the difference between what commonly is understood as *HyperSwap*, which relates to GDPS, and *Basic HyperSwap*, which is provided by TPC-R.

For details on TPC-R under z/OS and Basic HyperSwap refer to the following sources:

- *IBM TotalStorage Productivity Center for Replication for Series z*, SG24-7563
- *DS8000 and z/OS Basic Hyperswap*, REDP-4441

For more information about the HyperSwap features that GDPS provides, refer to *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374.

Peer-to-peer dynamic address switching (P/DAS)

In the mid-1990s, a first attempt was made to swap disk volumes, more or less transparently to application I/Os, from a PPRC primary volume to a PPRC secondary volume. This process was called *peer-to-peer dynamic address switching* (P/DAS). The process was single threaded for each single device swap and was serially executed for each single PPRC pair on each single z/OS image within a Parallel Sysplex®. It also included a process that halted all I/O first to the impacted primary device and included additional activities before issuing the actual swap command. After the swap successfully ended in all systems belonging to the concerned Sysplex, the stopped I/O was released again. From that point on, all I/O was redirected to the former secondary device.

This first attempt was intended to provide continuous application availability for planned swaps. P/DAS was not intended for unplanned events when the application volume failed. Therefore, P/DAS contributes to continuous availability and is not related to disaster recovery. P/DAS is not suited for large configurations, and the amount of time required even for a limited set of Metro Mirror volume pairs to swap is prohibitive by today's standards.

Details about P/DAS are still publicly available in a patent database under the *United States Patent US6973586*. However, the patent description is not 100% reflected in the actual P/DAS implementation. The P/DAS details can be found on the following Web site:

http://www.freepatentsonline.com/6973586.pdf

HyperSwap basically is a follow-on development of the P/DAS idea and an elaboration of the P/DAS process.

Figure 3 shows an abstract view of a Metro Mirror volume pair and its representation within z/OS by the volumes' Unit Control Blocks (UCB).

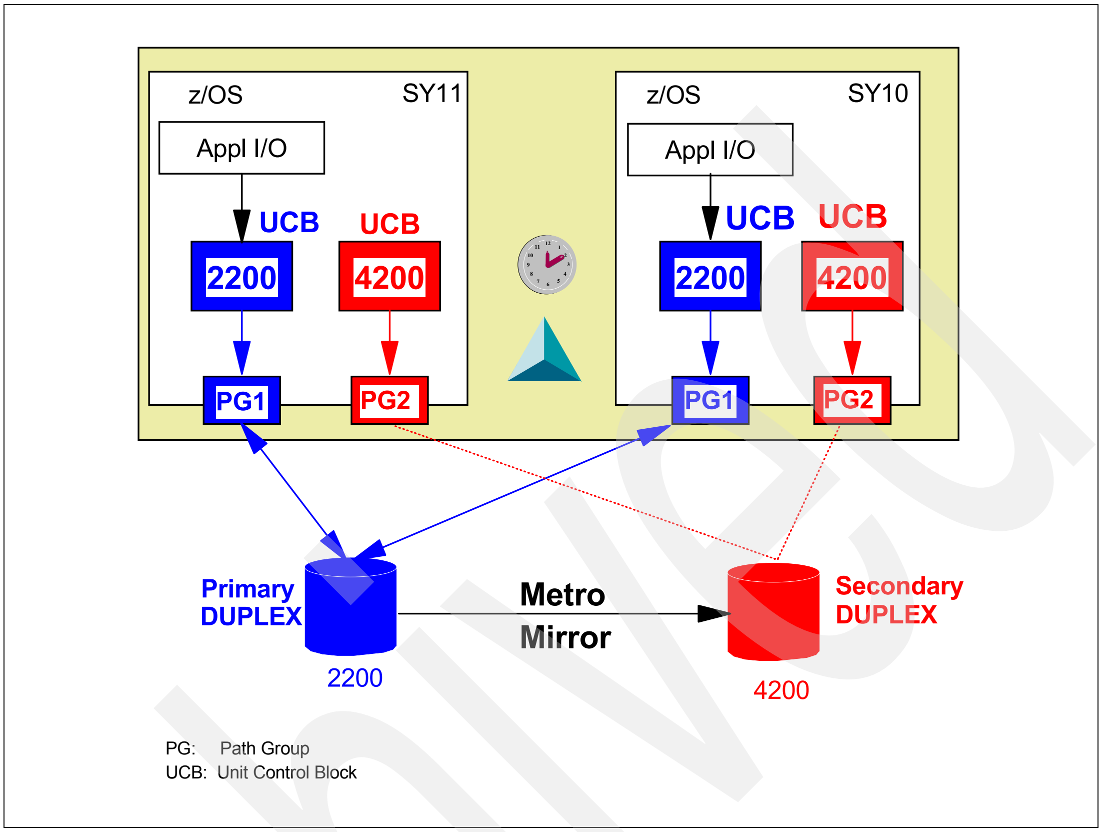


Figure 3 Basic PPRC configuration

UCBs as depicted in Figure 3 are the z/OS software representation of z/OS volumes. Among other state and condition information, the UCB also contains a link to the path or group of paths needed to connect to the device itself. An application I/O is performed and controlled through this UCB. The simplified structure depicted in Figure 3 implies that an application I/O always serializes first on the UCB and then continues to select a path from the path group, which is associated with that UCB. The UCB is then used for the actual I/O operation itself.

The Metro Mirror secondary device does not have validated paths and cannot be accessed through a conventional application I/O.[2]

[2] TSO PPRC commands might still execute their particular I/O to the secondary device. Other exceptions are the PPRC-related commands through ICKDSF.

Start swap process and quiesce all I/Os to concerned primary devices

Figure 4 shows the first step before the occurrence of the actual device swap operation, which can be accomplished only in a planned fashion.

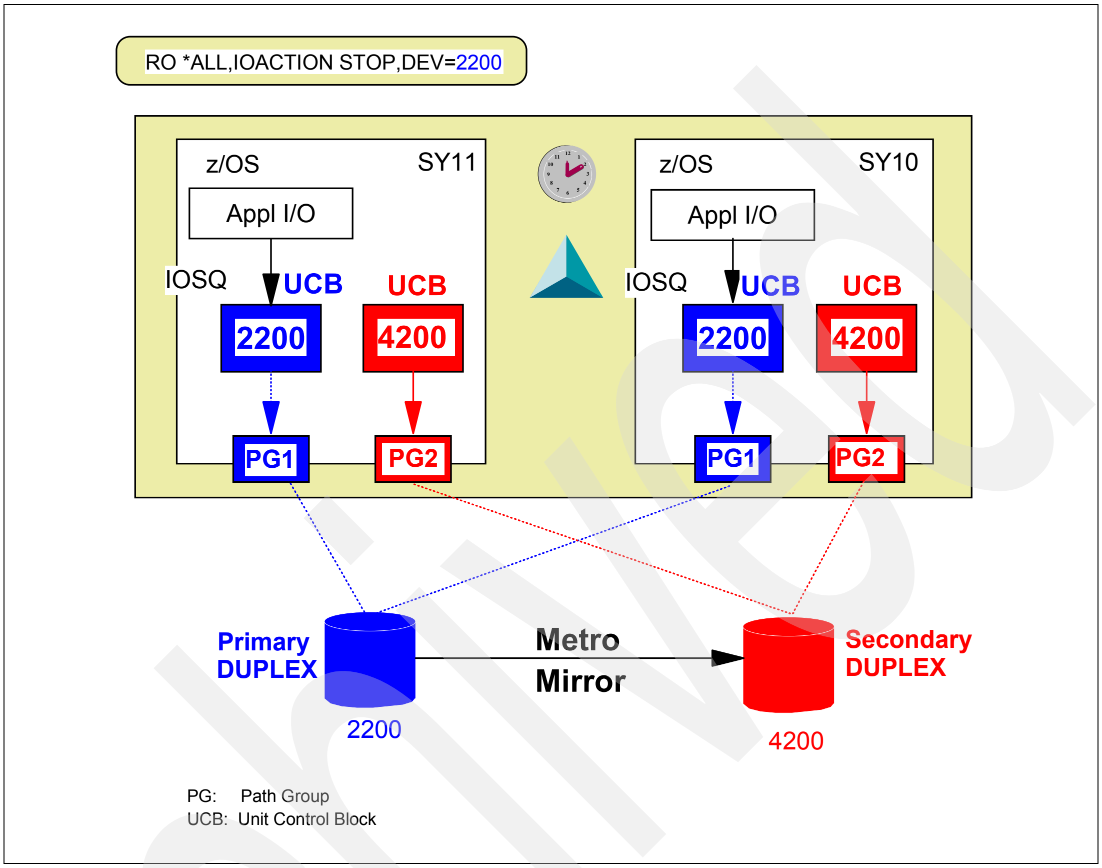


Figure 4 Quiesce I/O to device before swapping

In Figure 4, we first quiesce all application I/Os to the Metro Mirror primary device in all attached system images. This process does not end an active I/O. The I/O simply is quiesced at the UCB level, and the time is counted as IOS queuing time.

This quiescence operation is performed by the user. The user has to verify that the I/O is successfully quiesced. IOACTION,STOP returns a message to the SYSLOG when the I/O for the primary device has been stopped. The user also can verify that the I/O is successfully quiesced with the *D* IOS,STOP command.

Current HyperSwap support through GDPS and TPC-R incorporates this I/O quiesce step into its management function stream, which is transparent to the user.

When all attached systems have received the message that I/O has been stopped to the concerned Metro Mirror primary device, you continue with the actual SWAP command.

Actual swap process

Figure 5 shows the next step in a P/DAS operation, which is the actual device swap operation. No I/O takes places due to the execution of the previous I/O quiesce command, the IOACTION command, which is indicated in Figure 5 by the dashed lines between the path groups and the related devices.

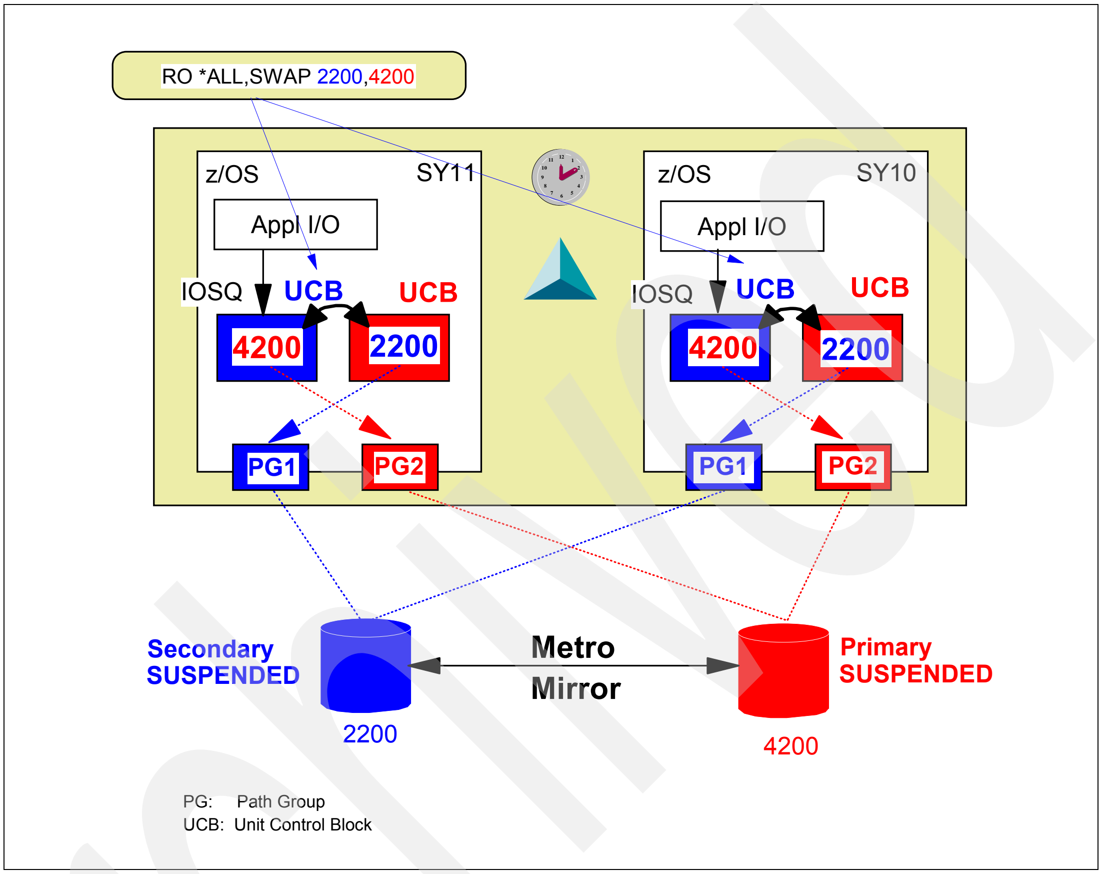


Figure 5 Swap devices exchanging the UCB content of both devices

Without looking into all possible variations of the swap operation, we can assume a planned swap and respond to the SWAP 2200,4200 triggered WTOR with switch pair and swap. This swap operation leads first to a validation process to determine whether the primary and secondary PPRC (Metro Mirror) volumes are in a state that guarantees that the actual swap is going to be successful and does not cause any integrity exposure to the data volumes. Therefore, both volumes must be in DUPLEX state. This validation process is performed through IOS and does not change the functions that Basic HyperSwap provides.

The swap operation then literally swaps the content of the UCBs between UCB for device 2200 and UCB for device 4200. This action automatically leads to a new route to the device when the application I/O continues after the swap operation is finished.

The swap operation then continues with PPRC (Metro Mirror) related commands and terminates the PPRC (Metro Mirror) pair. This action is followed by establishing the PPRC (Metro Mirror) pair with NOCOPY in the reverse direction. In our example in Figure 4 on page 8, the reverse direction is from device 4200 to device 2200. We assume that the PPRC path exists in either direction, which is required for the swap operation to succeed.

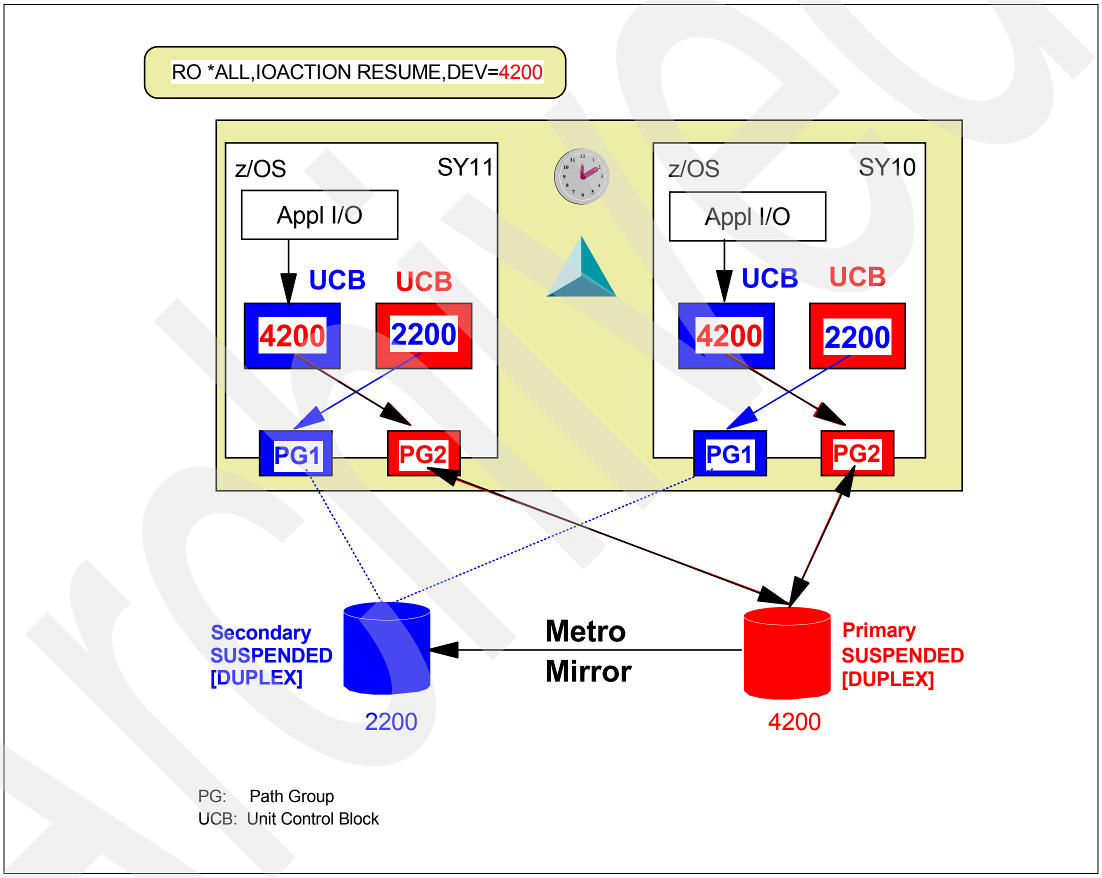
The next operation is a PPRC suspend of the volume pair. A PPRC primary SUSPENDED volume has a change recording bitmap that records all potentially changed tracks for a later resync operation. Changed tracks are only masked in this bitmap and are not copied. However, the I/O is still halted. In the next step, the quiesced I/O is released again.

These PPRC-related commands are controlled by IOS.

HyperSwap uses different Copy Services commands to transparently fail over to the Metro Mirror secondary volumes, which then change to primary SUSPENDED. A subsequent failback operation transparently resynchronizes the Metro Mirror relationship from 4200 → 2200. Another planned HyperSwap establishes the previous Metro Mirror relationship from 2200 → 4200.

Resume I/O after swap completes

Figure 6 illustrates the IOACTION RESUME command, which is directed to the former secondary device, which, in our example, is device number 4200.



```
RO *ALL,IOACTION RESUME,DEV=4200
```

Figure 6 Resume I/O after successfully swapping devices

The user issues this IOACTION command for all concerned members within the Sysplex.

From this point on, the application I/O continues to be executed and is redirected to the former secondary device, which is due to the swap of the UCB content between the primary and secondary devices. P/DAS then copies the potentially changed tracks in 4200 back to the former primary device, device number 2200, and this PPRC pair changes its status to DUPLEX.

GDPS with HyperSwap support or TPC-R Basic HyperSwap make this process totally transparent to the user. The cumbersome command sequence and lengthy execution time that P/DAS formerly imposed is no longer required.

P/DAS considerations and follow-on device swap development

The entire P/DAS process, as used in the past, was somewhat complex, and it did not always successfully complete, especially when a large number of PPRC volume pairs were swapped at once. An even more severe problem was the time the process itself required to complete. A rough estimate formerly was in the range of 10+ seconds per PPRC volume pair.[3] This amount of time was required because you also had to account for the operator to respond to a WTOR caused by the SWAP command and to issue these IOACTION commands before and after the SWAP command.

Therefore, P/DAS was used only by software vendors such as Softek (now part of IBM) and its Transparent Data Migration Facility (TDMF™), performing a transparent device swap to the application I/O at the end of a volume migration process.

Later, the P/DAS process was enhanced and improved to execute the swap of many PPRC volumes within a few seconds. This enhanced swap process was called *HyperSwap* but only in context with GDPS because GDPS was the first interface to utilize this new swap capability. The new swap capability was introduced with GDPS/PPRC with HyperSwap in 2002. Depending on the number of involved Sysplex members, GDPS/PPRC with HyperSwap was able to swap as many as 5000 PPRC pairs within approximately 15 seconds with five z/OS images involved (swapping one PPRC pair through P/DAS in such an environment would take 15 seconds).

With the advent of TPC-R and its ability to trigger the same swapping function, Basic HyperSwap became possible. We call it *Basic* because TPC-R does not provide anything beyond a planned swap trigger and avoids the entire swap process once a HyperSwap is triggered in an unplanned fashion. The HyperSwap is then handled by the z/OS IOS component, a process detailed in “Basic HyperSwap,” which follows.

In contrast, GDPS provides a controlling supervisor through its controlling LPAR(s), which manage not only the result of a device swap operation but also offers complete Parallel Sysplex server management for planned and unplanned events. This feature includes all automation capabilities based on NetView® and System Automation, including starting and stopping applications, networks, and entire systems.

Basic HyperSwap

Basic HyperSwap is a z/OS enhancement that contributes to high availability. Basic HyperSwap is not a disaster recovery function.

Basic HyperSwap is a storage-related activity that takes place at the System z server within z/OS. The management interface for Basic HyperSwap is TPC-R. The reader must not misunderstand Basic HyperSwap as a TPC-R function. TPC-R only offers a new session type called *a HyperSwap session*. This TPC-R-related HyperSwap session simply contains an underlying Metro Mirror session and is essentially managed just like a TPC-R Metro Mirror session.

TPC-R manages the underlying Metro Mirror session through a user-friendly interface. TPC-R management includes loading the configuration into z/OS when all Metro Mirror pairs have reached FULL DUPLEX state. The process also includes adding new volumes or triggering a planned HyperSwap.

[3] This number is optimistic and assumes some automation support.

TPC-R server within z/OS

Figure 7 shows a simplified structure of TPC-R when set up in a z/OS system.

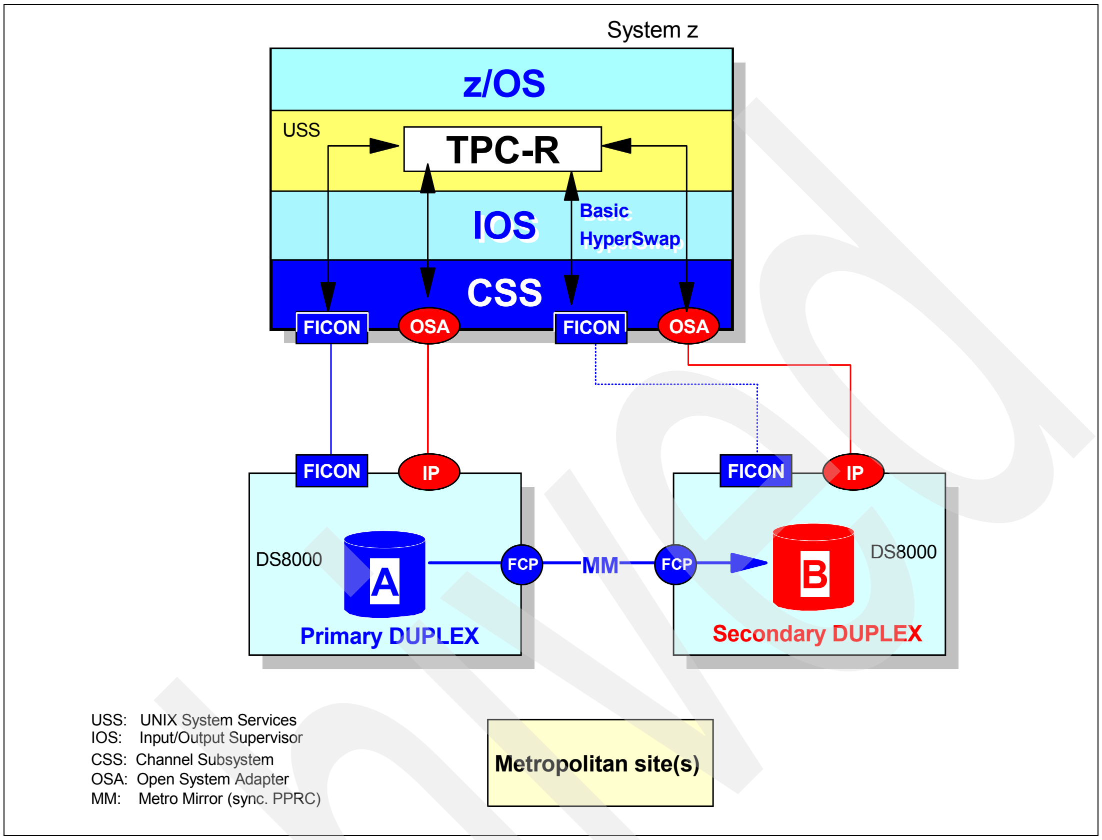


Figure 7 TPC-R connectivity in a System z environment

The TPC-R server code runs in an z/OS address space and utilizes underlying Unix System Services within z/OS. TPC-R manages the following two routes to the disk storage subsystems:

1. IBM FICON® channels that are shared with regular data I/Os from applications. FICON I/O is utilized by TPC-R to send certain Metro Mirror commands to the disk storage subsystem.
2. An Ethernet route to the disk storage subsystem. At the time of writing this paper, this Ethernet network connects to dedicated Ethernet ports within the System p® servers in the DS8000. A second possibility is connecting the TPC-R server with the DS8000, which is then the Ethernet port of the DS8000 HMC from IBM DS8000 Licensed Machine Code Release V4.1 and later. This Ethernet network is typically used to query and manage the DS8000 from the TPC-R server and can also support a heartbeat function between the disk storage server and the TPC-R server.[4]

> **Note:** In a TPC-R environment where the TPC-R server is not hosted in System z, all communications and controls between TPC-R and the DS8000 are performed through the Ethernet-based network.

[4] TPC-R can also manage Copy Services configurations with IBM DS6000™ and IBM ESS 800 in addition to the DS8000 in z/OS configurations. Of course TPC-R can also manage Copy Services configurations on IBM System Storage™ SAN Volume Controllers from any supported TPC-R server platforms, including z/OS.

TPC-R and Basic HyperSwap

Figure 8 shows a Basic HyperSwap-enabled environment. TPC-R must run on z/OS to manage a Basic HyperSwap session. For Basic HyperSwap support, TPC-R cannot run on any other of the platforms where it is usually supported (Microsoft® Windows® 2003, AIX®, or Linux).

> **Note:** Figure 8 shows only components that are relevant to HyperSwap.

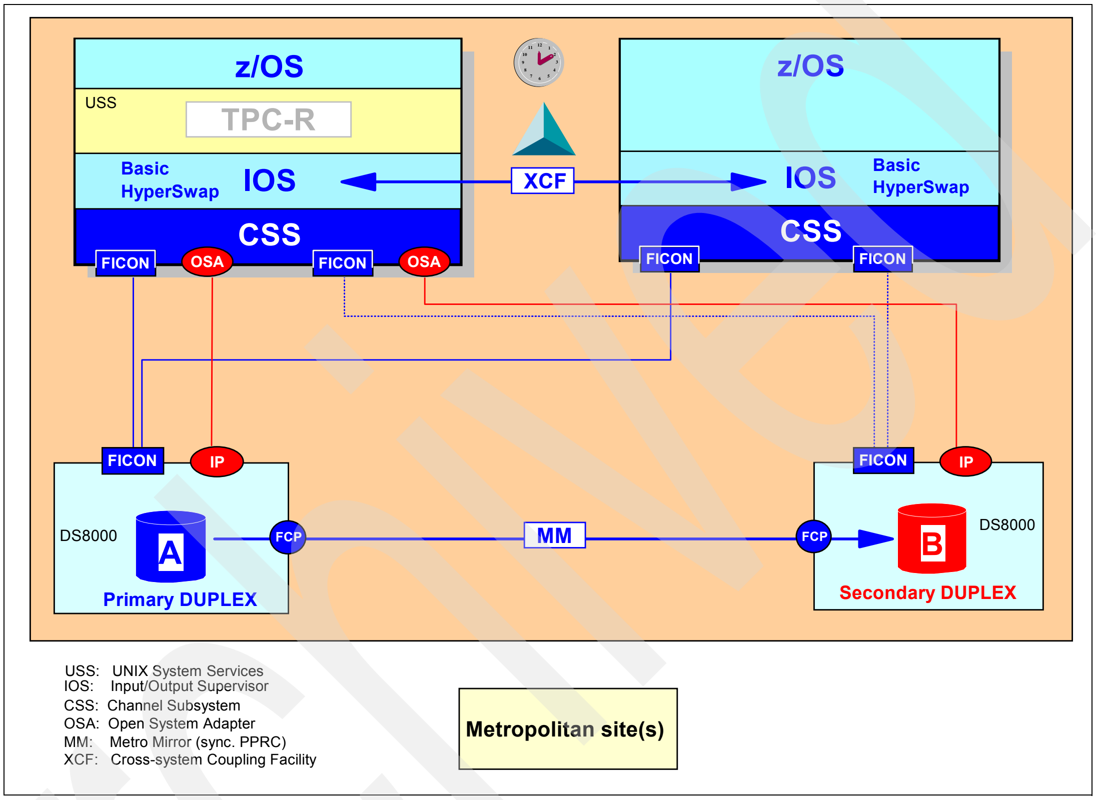


Figure 8 Basic HyperSwap-enabled environment

HyperSwap enabled means that all volume pairs within the TPC-R HyperSwap session are in a proper state, which is PRIMARY DUPLEX or SECONDARY DUPLEX. If a pair within the TPC-R HyperSwap session is not full DUPLEX, the session is not HyperSwap ready or HyperSwap enabled.

> **Important:** Remember that TPC/R does not play an active role when the HyperSwap itself occurs, which is the reason why Figure 8 shows the TPC/R address space grayed out. HyperSwap itself is carried out and managed by IOS, a z/OS component responsible for driving all I/Os.

HyperSwaps can occur in a planned or an unplanned fashion. Planned HyperSwap is triggered through TPC-R even though TPC-R does not continue to play a role after issuing the planned trigger event to IOS. IOS then handles and manages all subsequent activities following the swap of all device pairs defined within the TPC-R HyperSwap session, which includes a freeze and failover in the context of Metro Mirror management.[5] The freeze

[5] The difference between P/DAS, which did the PPRC swap through delete pair, and establish pair with NOCOPY and HyperSwap, which uses the failover function that the Copy Services of the DS8000 firmware provides.

guarantees an I/O quiesce to all concerned primary devices. IOS performs the actual swap activity.

In the present discussion, we are still referring to Basic HyperSwap, even when describing the process of incrementally swapping back to the original configuration. This swap is possible because the underlying Metro Mirror session within the TPC-R HyperSwap session provides all the Metro Mirror-related functions.

HyperSwap management by IOS also includes the management of the volume alias relationships when using Parallel Access Volumes (PAV) such as in Dynamic PAV or HyperPAV. In this context we recommend you configure the implicated disk storage subsystems to be identical as possible in regard to base volumes and alias volumes. For example, a Metro Mirrored LSS/LCU in *A* with base devices 00-BF and alias devices C0-FF ought to have a matching LSS/LCU in *B*. Following this recommendation simplifies the handling of the configuration. For more hints and tips, refer to *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374 and to *DS8000 and z/OS Basic Hyperswap*, REDP-4441.

GDPS/HyperSwap Manager

Since its inception, GDPS was the only application able to use the HyperSwap service provided by z/OS. Figure 9 shows a schematic view of a GDPS-controlled configuration. Note the GDPS-controlling LPAR, which is a stand-alone image with its own z/OS system volumes. In Figure 9, this system, SY1P. SY1P, runs the GDPS code as well as all managed application-based images such as SY10 and SY11.

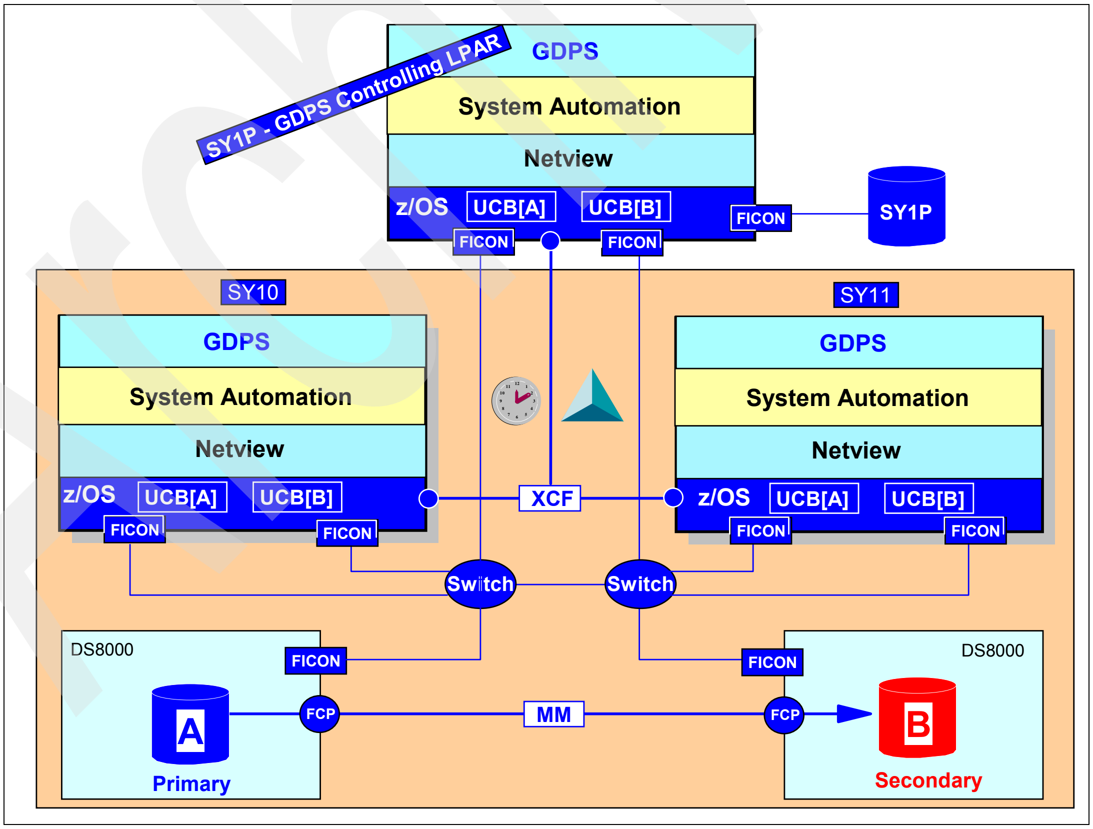


Figure 9 Schematic GDPS-controlled configuration

GDPS requires NetView and System Automation and communicates through XCF with all attached IBM Parallel Sysplex members.

> **Note:** Figure 9 does not depict where SY1P is located nor does it show other configuration-critical details such as a redundant FICON infrastructure and the IP connections from the GDPS controlling LPAR to the concerned System z HMCs (required to reconfigure LPARs, which might be necessary in a failover). For details refer to *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374, an IBM Redbooks® publication that was developed and written by a team of GDPS experts.

GDPS/PPRC with HyperSwap and GDPS/HM have used this z/OS-related device-swapping capability since 2002. GDPS with its concept of a controlling LPAR at the remote site or two controlling LPARs, one LPAR at either site, can act as an independent supervisor on a Parallel Sysplex configuration. In addition to handling all Copy Services functions, it also manages all server-related aspects, including handling couple data sets during a planned or unplanned site switch.

It is important to understand the difference between what GDPS/PPRC with HyperSwap provides as opposed to what GDPS/HM or Basic HyperSwap through TPC-R provides.

As discussed earlier, Basic HyperSwap, when handled and managed through a TPC-R HyperSwap session, provides full swapping capabilities as GDPS does. The difference between TPC-R-managed Basic HyperSwap and GDPS-managed HyperSwap is that TPC-R does not handle the actual swap and does not provide any sort of action when the swap itself does not successfully complete. IOS handles the swap activities and provides some error recovery when the swap cannot successfully finish. However, TPC-R provides Copy Services management when implementing a fee-based, full-function TPC-R environment. TPC-R does not provide automation procedures that can be activated through certain events.

In contrast to TPC-R and IOS, GDPS provides some post-processing capabilities after a failed swap attempt. For example, when a swap operation does not successfully finish within a z/OS image, GDPS ensures that this system does not continue its operations with a partially swapped configuration. In this case, GDPS resets the impacted LPAR to guarantee data integrity. GDPS provides additional functionality such as Parallel Sysplex management and allows user-specific and predetermined scripts to run under GDPS control.

Table 1 provides an overview and side-by-side comparison of TPC-R and its Basic HyperSwap with GDPS-managed HyperSwap. However, this table is not a complete list of all the relevant points of comparison.

Table 1 TPC-R versus GDPS for HyperSwap management

Topic or Function	TPC-R	GDPS/HM or GDPS/PPRC
Minimum prerequisites	z/OS V1.9+ with PTF, SSRE Internal Repository through Derby Cloudscape® provided when TPC-R server is installed	GDPS code, NetView, System Automation, IBM service engagement
Multiple site support	No	Yes
Manage Copy Services configurations based on DS8000, DS6000, ESS 800	Yes	Yes
Manage Copy Services configurations based on SVC	Yes	No

Topic or Function	TPC-R	GDPS/HM or GDPS/PPRC
Manage Copy Services in other vendor high-end storage subsystems	No	Yes, provided that they follow Copy Services standards (for example, ESS 800, DS6000, and DS8000)
Provide transparent device swap when configuration is HyperSwap ready	Yes	Yes
Allow user-specific automation procedures	No	Yes
Provide Parallel Sysplex Server management	No	Yes
Provide error recovery when swap operation fails	IOS (some recovery provided)	Error recovery by IOS and by GDPS
Manage planned disk failure outages	Yes	Yes
Manage unplanned disk failures	No	Yes
HyperSwap Linux on System z data	No	Yes
Manage Couple Data Sets	No	Yes

The following points summarize our discussion in this section:

- TPC-R is a Copy Services management framework that handles Copy Services configurations based on ESS 800, DS6000, DS8000, and SVC.
- GDPS, besides managing Copy Services configurations, provides a complete automation framework to manage Parallel Sysplex and automation capabilities based on NetView and System Automation. In addition to the IBM ESS 800, DS6000, and DS8000, GDPS can also manage other vendors' high-end disk storage subsystems as long as these systems follow IBM Copy Services standards as provided in ESS 800, DS6000, and DS8000.
- z/OS provides the HyperSwap capability, which contributes to high availability and continuous availability. Both GDPS and TPC-R are similar in their capability to exploit the z/OS-based HyperSwap services.

Basic HyperSwap trigger

At present, no published API triggers Basic HyperSwap. Unplanned HyperSwap triggers are recognized within IOS and handled by IOS. For its HyperSwap session, TPC-R can trigger only a planned HyperSwap. Currently TPC-R allows a HyperSwap session only within a single Parallel Sysplex configuration.

IOS also contains a comprehensive list of trigger messages that cause IOS to start a swap operation for Metro Mirror volume pairs within a TPC-R HyperSwap session. This list is derived from observation and analysis over time of situations without HyperSwap support that had the potential of Parallel Sysplex outages or that actually caused an outage.

HyperSwap through GDPS

GDPS takes a similar approach and knows all trigger messages that can lead to a swap operation for Metro Mirror volume pairs within the scope of a GDPS configuration. Whether or not the first indication shows up in an application-based system or in the GDPS-controlling LPAR does not matter. The actual swap operation is triggered and controlled by the GDPS-controlling LPAR.

z/OS Metro/Global Mirror Incremental Resync

In this section, we first describe how to configure a z/OS Metro/Global Mirror configuration. Then, we discuss a z/OS Metro/Global Mirror configuration with incremental resynchronization support. Finally, we describe the enhancements made to RMZ Resync in conjunction with HyperSwap.

Important: RMZ Resync is supported by GDPS exclusively. It is not supported through Basic HyperSwap and its related TPC-R HyperSwap session.

z/OS Metro/Global Mirror

z/OS Metro/Global Mirror is a multiple target three-site solution. As the name implies, it consists of a Metro Mirror relationship and a zGM relationship.

Both relationships copy from the same primary or source volume but to different secondary or target volumes.

Note: *zGM* is the new term for what was previously known as *Extended Remote Copy*.

Figure 10 shows a possible MzGM configuration.

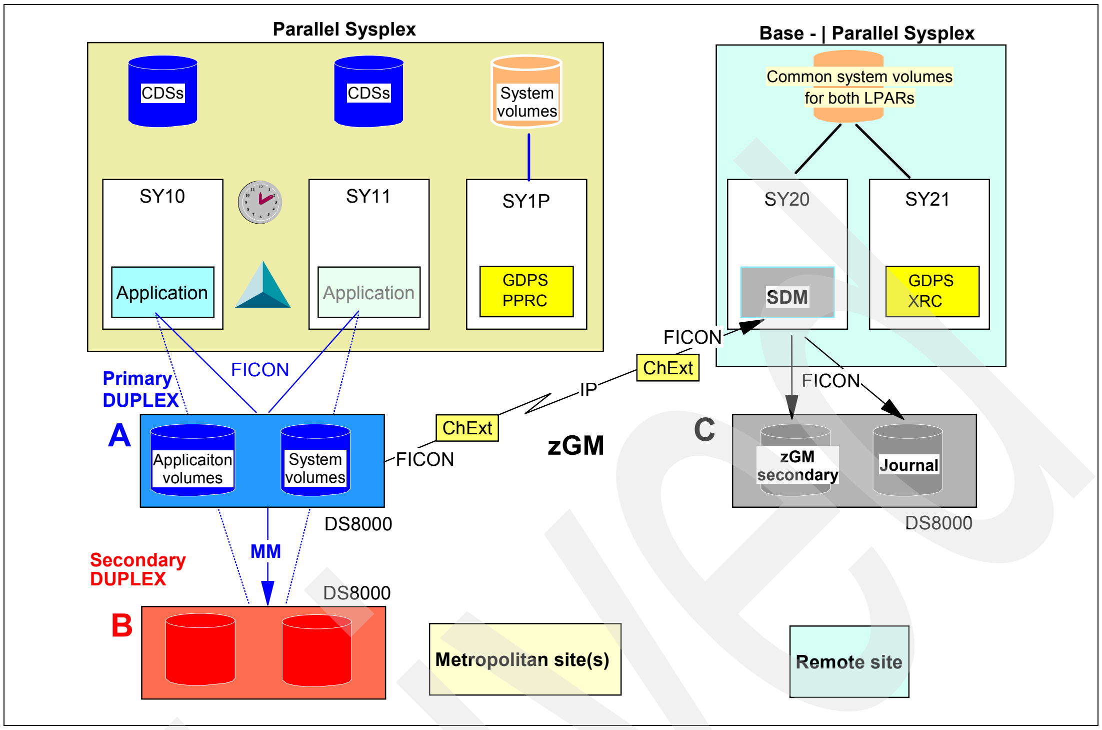


Figure 10 Multiple target three-site configuration, MzGM

In Figure 10, Volume A has the following two Copy Services relationships:

1. Volume A is primary to the Metro Mirror relationship between A and B. Both disk storage subsystems, A and B, are connected through PPRC FCP links.

   Volume A as the primary or application volume is accessed by the application LPARs, which are represented in Figure 10 by the two images, SY10 and SY11.

   The HyperSwap function is made possible through the Metro Mirror relationship.

2. At the same time, volume A is also primary to the zGM session, with volume C as the corresponding zGM secondary volume.

In Figure 10, System SY1P is the GDPS-controlling system and is also located in a metropolitan area, that is, within the distance limits imposed by Parallel Sysplex specifications. In our discussion, we leave open as to whether SY10 and SY11 are in the same or in different sites and whether Figure 10 depicts a single or multiple site workload configuration. For more information about different configurations and their GDPS-based management, refer to *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374.

Note: No communication interlocking exists between the GDPS instances, GDPS/PPRC in SY1P and GDPS/XRC in SY21.

The purpose of a three-site configuration is to provide high availability and, at the same time, disaster recovery capability, using a remote disaster recovery site, if necessary, with limited impact on application response time.[6] The following two capabilities are provided:

- High availability is achieved through the z/OS based HyperSwap service in the context of the Metro Mirror configuration and the potentially transparent volume swap managed by GDPS/PPRC or GDPS/HM. In this configuration, the following differences exist between these two management approaches:
  - GDPS/HM handles only disk subsystem failures within the same site or across the two local sites.
  - GDPS/PPRC HM includes the full function GDPS and handles all equipment failures including CEC failures. This feature exploits the capabilities of System z servers, VTS PtP or TS7700 Grid Tape Copy, as well as Persistent Sessions with various networking protocols and CF duplexing. In the event of a site A or site 1 failure, a choice can be made to recover at site B (site 2) or site C (site 3) with different RPO and RTO trade offs.
- As a second copy relationship for volume A, zGM provides disaster recovery capability in case of complete site failure at the metropolitan sites and can be managed by GDPS/XRC.

The distance between the metropolitan sites and the remote site is not limited when replicating volumes through zGM. With extended distance FICON, you can even consider omitting any in-between channel extenders.[7]

High availability despite primary site failures

A primary site failure might lead to failing devices in disk storage subsystem A (see Figure 11 on page 20).

Provided that GDPS/PPRC is implemented with HyperSwap support, this primary site failure leads to a HyperSwap trigger, and access to all A volumes is transparently swapped over to B volumes. This action contributes to high availability and even to continuous availability when performing a HyperSwap in a planned fashion (for whatever reasons, and there are plenty of reasons).

[6] Infrastructure costs usually are the biggest inhibitor for long distance replication and disaster recovery.

[7] Extended distance FICON channels are available with IBM System z10™ servers, DS8000 with Release V3.1 Licensed Machine Code, and z/OS V1.7 and later with PTF for APAR OA24218.

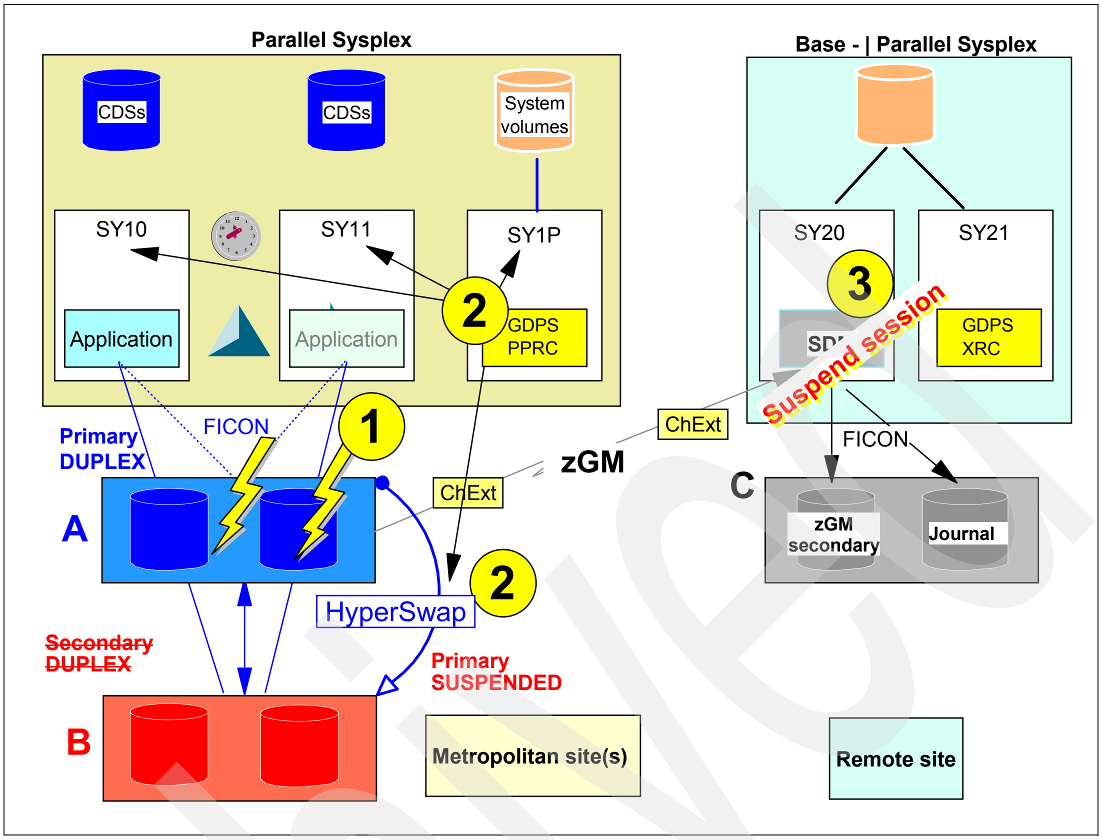


Figure 11 Primary volume failure

A failing I/O against an A volume causes the HyperSwap to start its staged process. The following process is controlled by GDPS/PPRC, out of the GDPS-controlling LPAR SY1P, as shown in Figure 11. (The numbers in the figure correspond to these steps.)

1. A first step is most likely the HyperSwap operation itself. During that time, in the course of reading changed records out of disk subsystem A, SDM cannot access the A volumes due to the freeze operation when the HyperSwap operation starts.
2. Acting independently of the HyperSwap process itself, SDM might eventually discover problems within disk subsystem A and might suspend its SDM session.[8] This action does not affect data volume consistency in C; by definition, all zGM secondary volumes provide a consistent set of volumes at all times.

   After the HyperSwap has completed, application I/O might occur in SY10 and SY11 and consequently update the volume set in disk subsystem B. As a result, zGM stalls and no longer replicates data. The current volume set in C still represents a consistent set of volumes. However, this volume set ages over time, and the possibility of an RPO between B and C steadily increases as long as application I/Os write to B.

   As a result, new zGM session has to be created between B and C.

   Figure 12 on page 21 shows this new zGM session after connectivity is shifted from $A \rightarrow C$ to $B \rightarrow C$. Shifting the connectivity usually is not very difficult, because a FICON-switched infrastructure might exist and the required connectivity might already be configured.

[8] Normally such a SDM session has been established with error-level SESSION, which leads to a suspension of the entire session as soon as a single SDM primary volume fails.

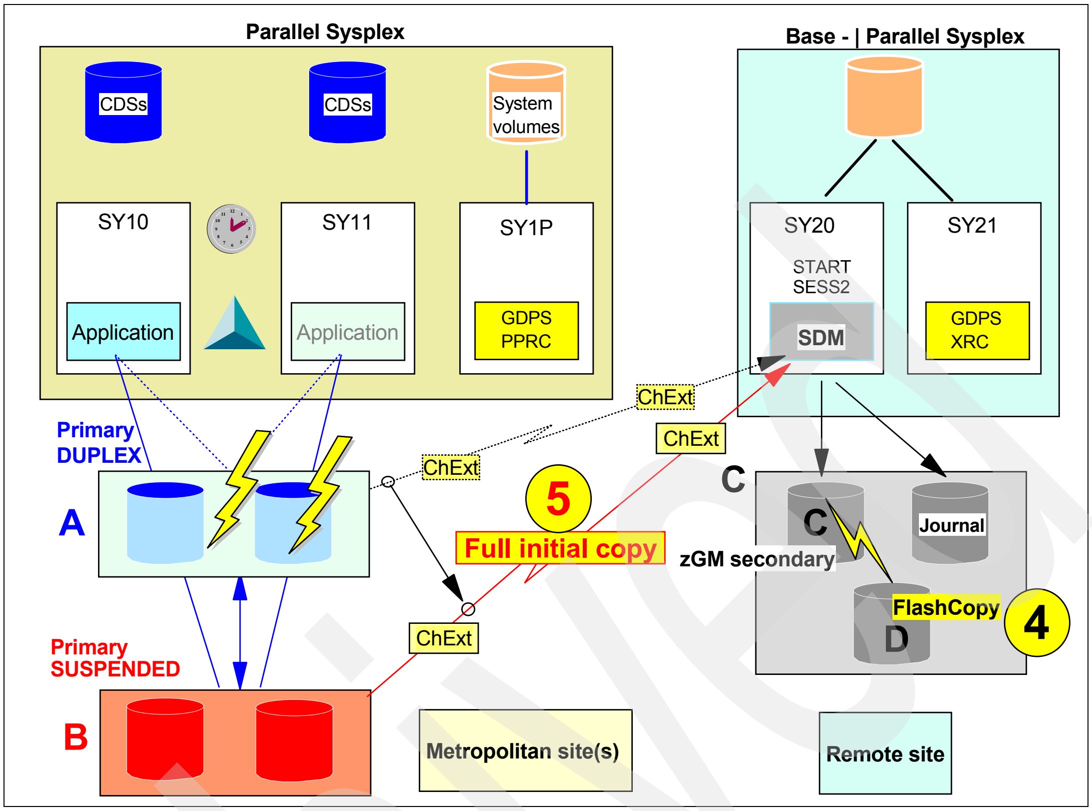


Figure 12 Resume zGM session with new session and full initial copy between B and C

3. Before starting a new full initial copy between B and C, which is required, you might want to consider first saving the set of consistent volumes in D and then starting the initial copy between B and C. This initial copy might run for several hours, and if any error occurs with the B volumes before the initial copy is completed and with all zGM volume pairs in DUPLEX state, the remote site has consistent source data volumes in D to restart. You can exercise GDPS/XRC to quickly FlashCopy® C volumes to D volumes.
4. The full initial copy after a volume A failure is the current drawback of the MzGM three-site configuration. High availability is achieved through HyperSwap between the A and B volumes. However, this is a drawback for the zGM session, because it does not follow the HyperSwap in an incremental fashion, which shifts its primary volumes automatically from A to B. To support an incremental resynchronization, GDPS/XRC V3.5+ is required in addition to GDPS/PPRC.

Failures in volumes B or C

Before moving on to our discussion of the improved RMZ Resync support, we must point out that failure of the B volumes is not too difficult to handle; the Metro Mirror session between volumes A and B simply suspends through a GDPS/PPRC-handled and -managed freeze operation. However, the failure of volume B disables the HyperSwap function. zGM continues to replicate the records from A to C, and application I/Os are not affected.

A failure at the remote site, volume C, does also not affect the application A volumes. The failure causes the $A \rightarrow C$ XRC session to suspend. Depending on what happens to the disk storage subsystems in C, an incremental resync to the zGM session can be handled through the hardware bitmaps maintained in the zGM primary disk subsystem for each A volume.

Incremental resync after site A failure and HyperSwap

With DS8000 Licensed Machine Code Release V3.1 and later and compatible SDM software in z/OS V1.7 and later, RMZ Resync provides incremental resync support to zGM (after Volume *A* fails and HyperSwap swaps automatically to the *B* volume).

Figure 13 is basically the same as Figure 10 on page 18; the only difference is the IP-based connectivity between the two GDPS instances that execute in this environment. GDPS/PPRC is connected with the remotely running GDPS/XRC, and both GDPS instances communicate through NetView over this infrastructure. In a sense, the GDPS instances are coupled together and utilize NetView to interact with one another over this Ethernet network.

> **Note:** The communication between the channel extenders can be any Telco protocol and does not have to be IP based, as shown in Figure 13.

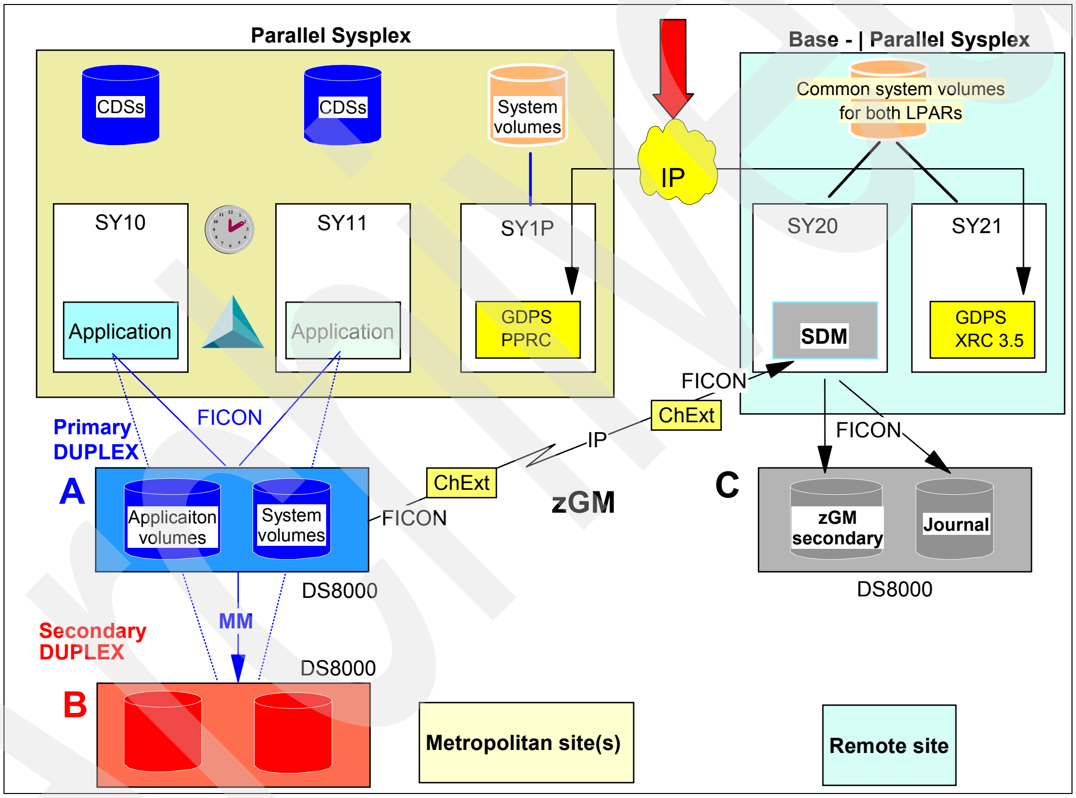


Figure 13 Multiple target three-site configuration, MzGM with "coupled" GDPS instances and IR support

Enhanced zGM and System Data Mover (SDM) code provide the new incremental resynchronization support for zGM. This requires also the DS8000 firmware level (Licensed Machine Code) Release 3.1 or later. Last but not least, GDPS/PPRC and GDPS/XRC needs to be at release 3.5 or later.

Figure 14 is not too different from Figure 12 on page 21, but the figure shows a big difference to the zGM session after a HyperSwap between *A* and *B*. The scenario shown in the figure applies to an unplanned HyperSwap as well as to a planned HyperSwap between volumes *A* and *B*.

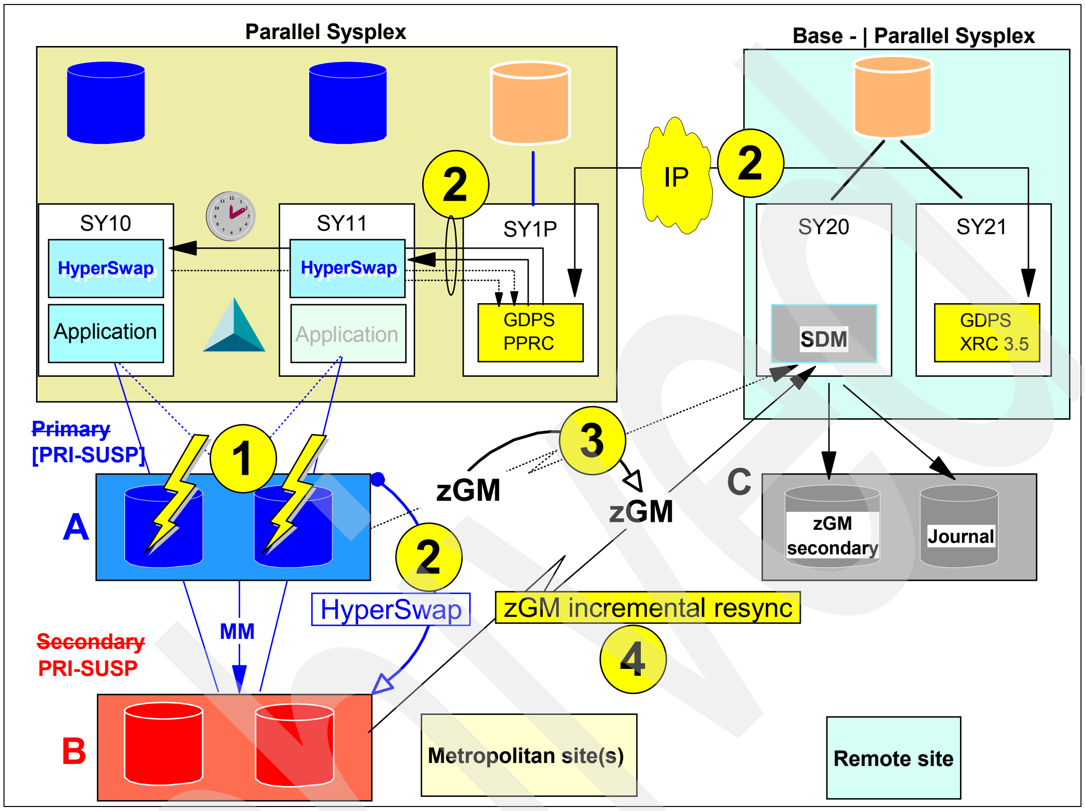


Figure 14 Unplanned HyperSwap in MzGM environment with MzGM IR support controlled by GDPS

The following sequence of activities for an unplanned hyperswap in an MzGM with IR environment is depicted (with some simplification) in Figure 14:

1. In the case of an unplanned HyperSwap caused by a failure in disk subsystem *A*, GDPS/PPRC in SY1P takes control and handles the follow-on activities.
2. GDPS/PPRC handles and controls the HyperSwap process and instructs all members in the Parallel Sysplex to perform the actual HyperSwap. GDPS/PPRC uses XCF communication to submit its commands to the Parallel Sysplex members to manage the HyperSwap process. GDPS/PPRC also instructs GDPS/XRC to end its session between *A* and *C* and swap to a “shadow” session between *B* and *C*. GDPS/XRC creates this shadow session with NOCOPY between *B* and *C* in the process of establishing the session between *A* and *C* for the purpose of doing a full initial copy between *A* and *C* (as need to occur in a RMZ Resync configuration). This communication between GDPS/PPRC and GDPS/XRC happens over the IP network as shown in Figure 14.
3. GDPS/XRC commands SDM to swap to the other session.
4. The SDM reader tasks resume the process, at this point reading record sets from the *B* storage subsystems, but only those record sets accumulated from the time of the HyperSwap and until SDM was reconnected to the *B* volumes. Bit-mapping management between the DS8000 and SDM makes this action possible.

This automated take over from one zGM session to the other zGM session is only possible through GDPS management.

Note that these same considerations, apply to site *2* or *B* volume failures. Depending on the policy defined, which can be FREEZE and GO, the application I/O to *A* volumes is not affected. However, HyperSwap is disabled because the Metro Mirror secondary volumes, the *B* volumes, are no longer available for a potential HyperSwap.

Volume *C* failures affect only the zGM session between *A* and *C*. The zGM session suspends. Later, the session between *A* and *C* can be resynchronized through the zGM hardware bitmaps that are maintained in the primary disk subsystem for each zGM suspended volume.

Adjustments to SDM, GDPS/XRC, and GDPS/PPRC

SDM provides a new mode of operation that is not externalized. This operation cannot be handled through TSO commands or a commonly available API; it can be handled only through GDPS/XRC.

During the HyperSwap process, the GDPS/PPRC system, system SY1P in Figure 14 on page 23, signals over its IP connectivity to the GDPS/XRC session to instruct SDM to stop reading from disk storage subsystem *A*. (Note that SDM might already have suspended its session and stopped reading from disk storage subsystem *A*.) The next step is an SDM internal swap operation to swap its primary volumes from volume *A* to volume *B*.

Because the session was previously suspended, a session restart is possible through a XADDPAIR command with the SUSPENDED keyword. This command starts the actual resynchronization.

Note: All the Metro Mirror volumes have to match all the volumes within the zGM session.

Because the SDM session needs connectivity to all involved volumes (meaning volumes in *A* and *B* as primary volumes and *C* volumes, and perhaps FlashCopy volumes, as secondary volumes), you can quickly reach the limit of a maximum 64,520 volumes within a z/OS system.

Adjustments are also made to GDPS/XRC and GDPS/PPRC and their GEOXPARM and GEOPARM parameters, respectively, to reflect the three-site configuration. For details, refer to *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374.

You can also add XRC fixed utility devices to the GDPS/PPRC configuration so that SDM can find its fixed utility devices after a HyperSwap from *A* to *B*.

Conclusion

RMZ Resync is a high availability and disaster recovery solution based on z/OS HyperSwap technology and DS8000 Copy Services functions. GDPS, a solution available exclusively through the IBM Services group, combines these functions and automatically handles them. This solution enables fully automated and transparent failover and failback management without impacting application I/O for component failures. RMZ Resync offers prompt restart capabilities at the remote site when the primary site totally fails. This premium solution is available for z/OS volumes and any system providing the time-stamp required by zGM.

The team that wrote this IBM Redpaper publication

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), San Jose Center.

Werner Bauer is a Certified Consulting IT Specialist in Germany. He has 26 years of experience in storage software and hardware as well as experience with S/390® and z/OS. His areas of expertise include disaster recovery solutions, ESS, and DS6000 and DS8000. He has written extensively in various IBM Redbooks publications, mainly about the DS6000, Ds8000 and TPC for Replication. He holds a degree in economics from the University of Heidelberg and in mechanical engineering from FH Heilbronn.

Bertrand Dufrasne is an IBM Certified IT Specialist and Project Leader for IBM System Storage disk products at the International Technical Support Organization, San Jose Center. He has worked at IBM in various IT areas. He has written many IBM Redbooks and has developed and taught technical workshops. Before joining the ITSO, he worked for IBM Global Services as an Application Architect. He holds a degree in electrical engineering.

Richard Muerke is a Senior IT Specialist in Germany. He has 25 years of experience with S/390 hardware and software, including 8 years in disk storage hardware. He works in Field Technical Sales Support for storage and SAN environments. His areas of expertise include performance analysis and implementations with FICON, IBM disk storage servers, and SAN directors from multiple vendors.

Thanks to the following people for their contributions to this project:

Bob Kern
William Micka
IBM US

Nick Clayton
IBM UK

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4429-00 was created or updated on January 9, 2009.

Send us your comments in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Redpaper™

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
Cloudscape®
DFSMS™
DS6000™
DS8000™
FICON®
FlashCopy®
GDPS®
HyperSwap™
IBM®
NetView®
Parallel Sysplex®
Redbooks®
Redbooks (logo) ®
REXX™
S/390®
System p®
System Storage™
System z10™
System z®
TDMF™
z/OS®
z/VM®
z10™

The following terms are trademarks of other companies:

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.