Redpaper

Patric Becker
Frank Neumann

IBM DB2 Analytics Accelerator High Availability and Disaster Recovery

Introduction

With the introduction of IBM® DB2® Analytics Accelerator, IBM enhanced DB2 for z/OS® capabilities to efficiently process long-running, analytical queries. Consequentially, the need arose to integrate the accelerator into existing High Availability (HA) architectures and Disaster Recovery (DR) processes. This paper focuses on different integration aspects of the IBM DB2 Analytics Accelerator into existing HA and DR environments and shares best practices to provide wanted Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO).

HA systems are usually a requirement in business critical environments and can be implemented by redundant, independent components. A failure of one of these components is detected automatically and their tasks are taken over by another component. Depending on business requirements, a system can be implemented in a way that users do not notice outages (continuous availability), or in a major disaster, users notice an outage and systems resume services after a defined period, potentially with loss of data from previous work.

System z® was strong for decades regarding HA and DR. By design, storage and operating systems are implemented in a way to support enhanced availability requirements. Parallel Sysplex® and Globally Dispersed Parallel Sysplex (GDPS®) offer a unique architecture to support various degrees of automated failover and availability concepts.

This IBM Redpaper® publication shows how IBM DB2 Analytics Accelerator can easily complement existing System z topologies for HA and DR.

Business continuity and disaster recovery

Business continuity and DR act at different levels in an organization. Business continuity is the strategy at the enterprise level; DR is the solution at the IT level. The concept of DR focuses on recovering IT systems from an unplanned outage to provide business continuity.

When we describe HA concepts in this document, we imply that these concepts provide IT services to achieve business continuity.

Business continuity solutions are based on the concept of data duplication between two or more data center sites, which are often called *primary site* and *secondary site*. If a primary site that processes production workload is struck by a disaster and IT systems are unavailable, IT operations are provided by the secondary site. Without a second site (or *recovery site*) providing the same IT services, business continuity is compromised. Depending on the chosen implementation of the data duplication between primary and secondary sites, businesses might encounter outages until operations are restored at the secondary site. Also, depending on the chosen implementation, transactions can be preserved or lost in a disaster.

Recovery time objective

RTO refers to how long your business can afford to wait for IT services to be resumed following a disaster. Regarding the IBM DB2 Analytics Accelerator, we refer to this term to resume operations not in DB2 for z/OS only, but with an accelerator being ready to process query workloads.

Recovery point objective

RPO refers to how much data your company is willing to re-create following a disaster. Regarding the IBM DB2 Analytics Accelerator, we refer to this term in relation to the data latency within the accelerator when query workloads can be processed.

Built in HA features within IBM DB2 Analytics Accelerator

The IBM DB2 Analytics Accelerator consists of multiple components that contribute to HA inside the physical machine. The following components are inherited from the underlying IBM PureData™ System for Analytics architecture:

- Netezza® Performance Server® hosts
- Redundant S-Blades
- Redundant network configuration
- Redundant array of independent disks

Netezza performance server hosts

Each IBM DB2 Analytics Accelerator appliance is equipped with two Netezza Performance Server (NPS®) hosts, which act as the interface to IBM System z. One NPS host is always active while the other host is in stand-by mode to take over if the first NPS host fails unexpectedly. All query requests are routed from DB2 for z/OS to the active NPS host via a 10-Gigabit Ethernet connection and are processed within the accelerator while being orchestrated from the active NPS host. In case the active NPS host fails, the stand-by NPS host takes over existing workload and processing continues as normal. Queries that ran during the failover of NPS hosts receive a negative SQLCODE and must be resubmitted by the requesting application.

Redundant S-Blades

Depending on the size of the physical IBM DB2 Analytics Accelerator appliance, a different number of Snippet-Blades or S-Blades is installed within a single appliance. Depending on the machine size, a certain number of spare S-Blades with no active role are installed in each physical rack. Each active S-Blade is connected to multiple disks and consists of multiple Field Programmable Gate Arrays (FPGAs) and multiple CPU cores to process query requests. If an active S-Blade fails, a spare blade trades roles and becomes a new active S-Blade. Disks that were assigned to the failing blade are reassigned to the new active S-Blade.

If no spare S-Blade is available when an active S-Blade fails, disks are reassigned to the remaining active S-Blades. This reassignment of disks without a spare S-Blade can degrade performance of all workload that is processed by the IBM DB2 Analytics Accelerator.

Redundant network configuration

Each NPS host in a PureData for Analytics server is equipped with dual-port 10-Gigabit Ethernet (10 GbE) interfaces. If one port becomes inactive, the card automatically uses the second port of the network interface. If the entire card fails unexpectedly, the same failover process takes places as for a failing NPS host that is described in “Netezza performance server hosts” on page 3.

Simple configurations with a single System z server and a single switch that is attached to an accelerator often connect to only one port per NPS host. More advanced configurations use two switches (again for availability, but also to connect multiple System z servers); this is where both ports in the 10 GbE card are used and TCP/IP traffic between DB2 and the accelerator have multiple redundant ways so that outage of one of the network components can fully be compensated.

For more information about HA network setups, see this website:

http://www-01.ibm.com/support/docview.wss?uid=swg27028171

Redundant array of independent disks

Redundant array of independent disks (RAID) technology allows for data mirroring across multiple disks in the same array. Therefore, if one disk fails, the data is preserved.

Each accelerator is equipped with multiple disk drives that use RAID technology. Each accelerator also has spare drives available that are not used during normal operation. Disk drives consist of three partitions: a data partition, a partition that contains mirrored data from another disk's data partition and a partition for temporary work data. If a disk fails, its data is regenerated from the corresponding mirrored data partition on a non-failing disk to a spare disk. The spare disks are automatically reconfigured so that the accelerator can continue to operate seamlessly after a disk failure.

Introducing HA concepts for IBM DB2 Analytics Accelerator

In addition to "built-in" capabilities of the appliance, there are software components and features to build even more extensive HA and DR solutions.

In general, HA and DR solutions are implemented by adding redundant components. This section describes how multiple accelerators or multiple DB2 subsystems can be configured and managed. The following concepts that are described act as the foundation to build advanced configurations and integrate them into existing HA/DR System z environments:

- Workload balancing
- Data maintenance and synchronization with multiple accelerators
- HA setup for incremental update
- High-performance storage saver and multiple accelerators

Workload balancing

A key feature in IBM DB2 Analytics Accelerator is the capability to automatically balance query routing from a DB2 for z/OS subsystem to multiple attached accelerators. With the option to share one or more accelerators between multiple DB2 for z/OS subsystems, this provides options for a flexible deployment pattern.

Figure 1 on page 5 shows a setup with one DB2 for z/OS subsystem (or member of a data sharing group) that contains data in tables T1, T2, and T3.

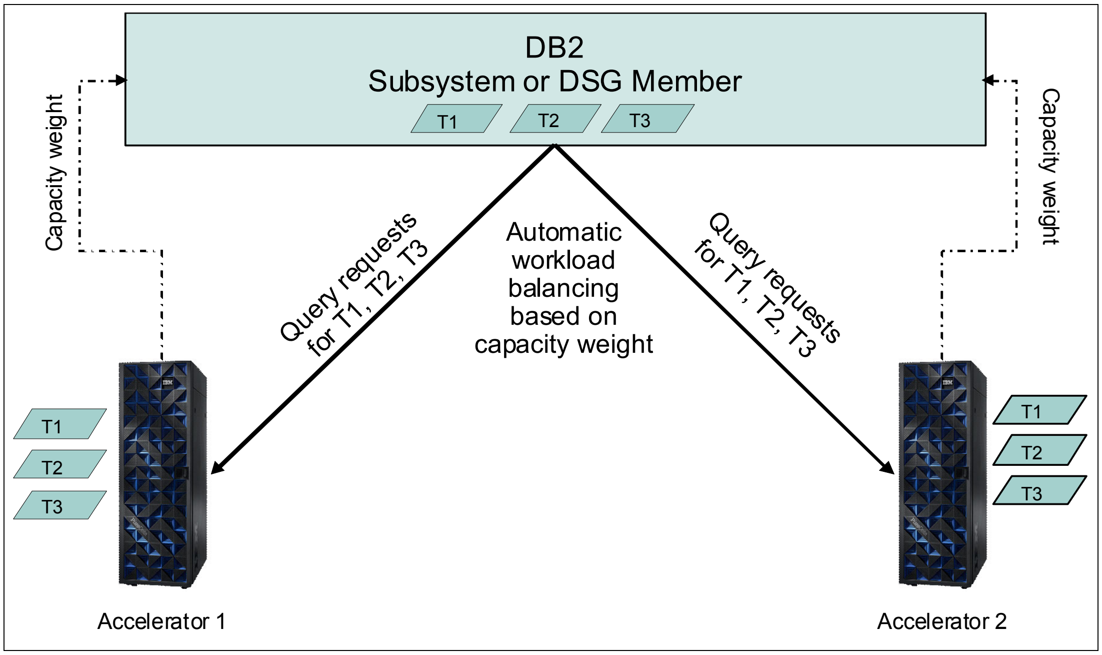
DB2
Subsystem or DSG Member
T1
T2
T3
Capacity weight
Capacity weight
Query requests for T1, T2, T3
Automatic workload balancing based on capacity weight
Query requests for T1, T2, T3
T1
T2
T3
T1
T2
T3
Accelerator 1
Accelerator 2

Figure 1 Workload balancing with multiple active accelerators

Two accelerators (Accelerator 1 and Accelerator 2) are defined for this DB2 for z/OS subsystem and both have tables T1, T2, and T3 loaded and active. For query processing and query routing, the DB2 for z/OS optimizer now must decide as to which accelerator to use if a query is eligible for acceleration. Each accelerator reports its current *capacity weight*, along with other monitoring parameters to each attached DB2 for z/OS subsystem. This capacity weight value includes information about the number of active execution nodes and the current node usage. The DB2 optimizer uses this value to implement automated workload balancing between these two accelerators: the system with a lower capacity weight value receives more queries for execution than the other system.

Because this workload balancing is applied to each query that is run on an accelerator, this feature is useful for adding capacity and for HA scenarios; if one accelerator fails, query requests automatically are routed to the remaining accelerator without any other administration intervention.

Figure 2 on page 6 shows a situation in which Accelerator 1 fails; the DB2 subsystem recognizes (based on heartbeat information) that the accelerator is not accessible and routes subsequent query requests to Accelerator 2 only.

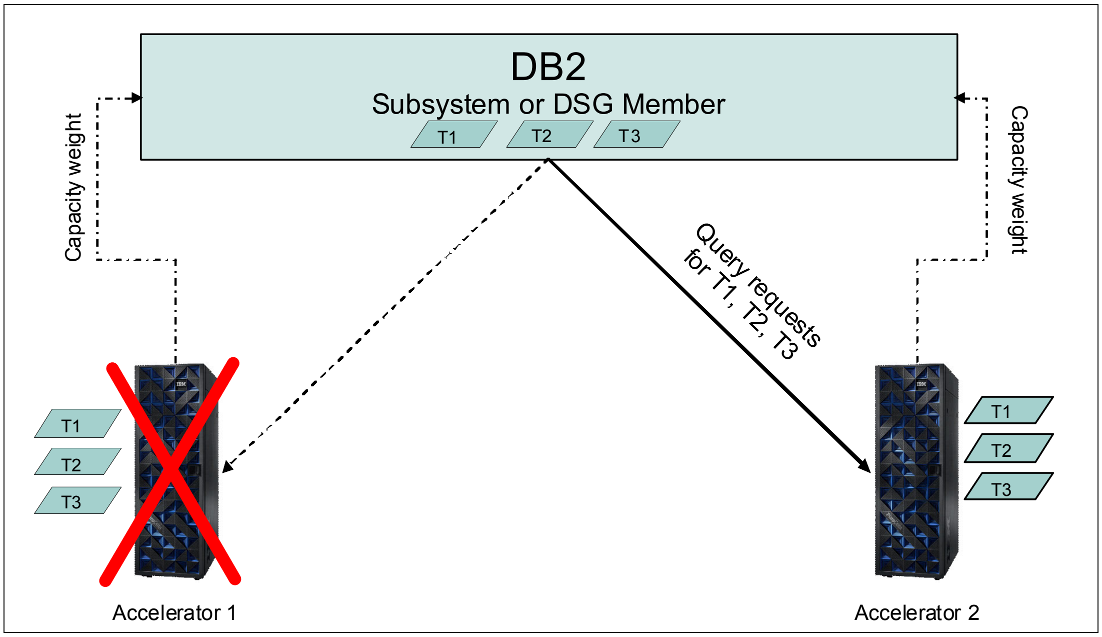

Figure 2 Query routing after failure of an active accelerator

Data maintenance and synchronization with multiple accelerators

All concepts that are in this document assume that data can be loaded consistently into multiple accelerators. For initial loads, this means that selected tables must be unloaded in DB2 for z/OS and loaded into each attached accelerator.

For incremental update, changes must be propagated to all accelerators. Shared log scraping can be used to reduce the overhead of log capturing. Data propagation and management effort is nevertheless higher.

Loading into multiple accelerators

Multiple options exist to keep data current in an IBM DB2 Analytics Accelerator. Those options include a full table reload into an accelerator, reload of one or more identified partitions (manually or by using the automatic change detection feature) or by using incremental update processing. For more information about incremental update processing, see “HA setup for incremental update” on page 10.

Data is loaded into IBM DB2 Analytics Accelerator by using stored procedure SYSPROC.ACCEL_LOAD_TABLES. To ensure the same results for all incoming queries (regardless which accelerator processes a query) and to optimally use the workload balancing capabilities of IBM DB2 Analytics Accelerator, it is recommended to load data into all available accelerators that are participating in a HA setup, as shown in Figure 3 on page 7.

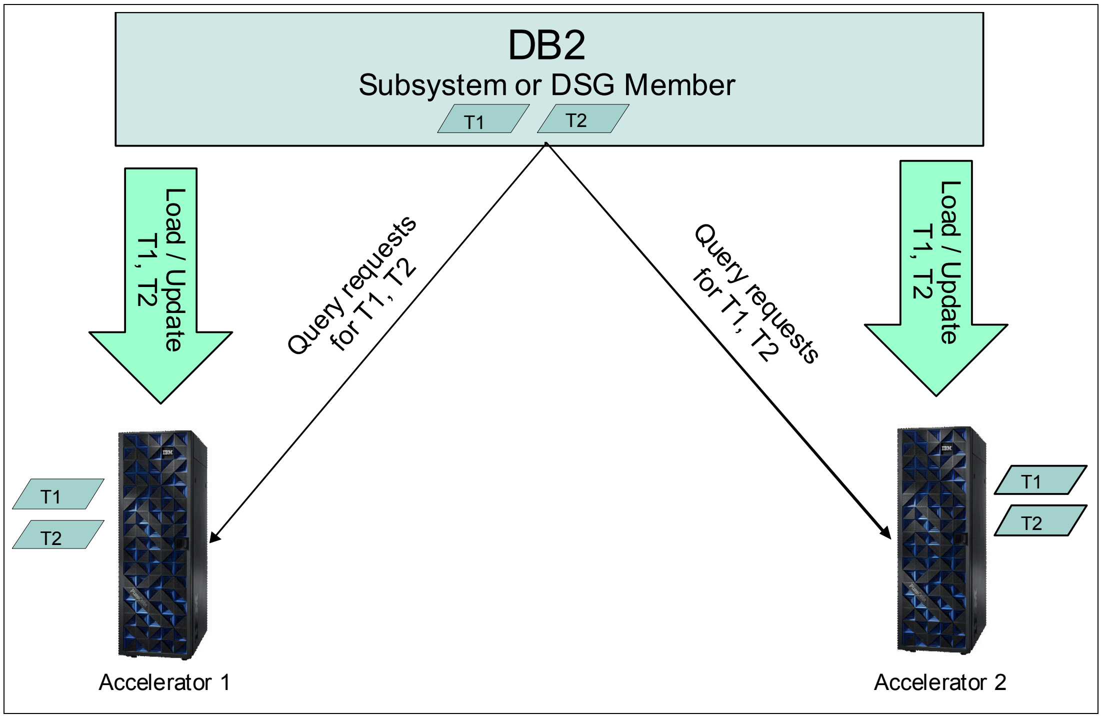

Figure 3 Maintaining two active accelerators with the same data

If one accelerator becomes unavailable, an accelerator that contains the same data as the failing one can provide RTO = 0. However, performance might be degraded if incoming query requests can be processed only by a single accelerator instead of two or more accelerators.

Note: To minimize the amount of data that is loaded into multiple accelerators, use the change detection feature of the stored procedure SYSPROC.ACCEL_LOAD_TABLES to load changed data only. For more information about keyword “detectChanges”, see *Stored Procedure Reference*, SH12-7039 (section “SYSPROC.ACCEL_LOAD_TABLES”).

Data consistency across multiple accelerators

Under certain situations, data in multiple accelerators can get out of sync, as shown in the following examples:

- During loading, if some active accelerators are loaded with new data while others are still waiting to be loaded and still have old data.
- If DB2 table data changes while accelerator loading in ongoing; in this case, different accelerators can take different snapshots of the data in DB2.

In both cases, queries might return different results, depending on which accelerator is chosen for query processing.

The first case can be addressed with the option to disable and enable a table for acceleration on an accelerator. The following update sequence for multiple accelerators might be used:

1. Disable table for acceleration on all but one accelerator.
2. Load data and enable tables for acceleration on all accelerators sequentially, starting with the accelerator that has the table that is enabled for acceleration.
3. During this time, query requests are processed by the only accelerator that has the table that is enabled for acceleration.

For the second case, table locks can be used to prevent changes during the load process. Under transactional control, the following sequence can be implemented:

1. Start transaction.
2. Lock table(set).
3. Call ACCEL_LOAD_TABLES for first accelerator.
4. Call ACCEL_LOAD_TABLES for second accelerator.
5. Release table(set) lock.

Another solution for both cases is to use the DB2 Analytics Accelerator Loader product, which is a separately licensed program. By using this tool, you can load data into accelerators from (consistent) image copies (or point-in-time data) so that updates in DB2 tables do not affect the data in the accelerators.

For an incremental update (for more information, see “HA setup for incremental update” on page 10), requirements are different. If data in the referenced DB2 table is continuously changing, applications must deal with continuously changing results as a consequence and a slightly different latency in applying changes on accelerators might not be an issue.

If there is query processing after relevant table changes, the waitForReplication option in the ACCEL_CONTROL_ACCELERATOR stored procedure can be used. This procedure must be started for all attached accelerators that are included in the HA setup (with the same data loaded) to ensure that the latest changes are applied. After these stored procedure calls are returned, query processing can safely be started with the latest changes included.

Loading a subset of data

If multiple fully synchronized accelerators are not an option, a subset of data can be stored in each accelerator. If there is a failure or disaster, the remaining data must be loaded into the remaining accelerator first, which causes RTO > 0 for this data. While this load process occurs, DB2 for z/OS can continue to serve query requests, possibly with higher CPU and elapsed times.

Figure 4 on page 9 shows a workload distribution example in which query requests for table T1 can run only on Accelerator 1 while query requests for table T2 can run only on Accelerator 2.

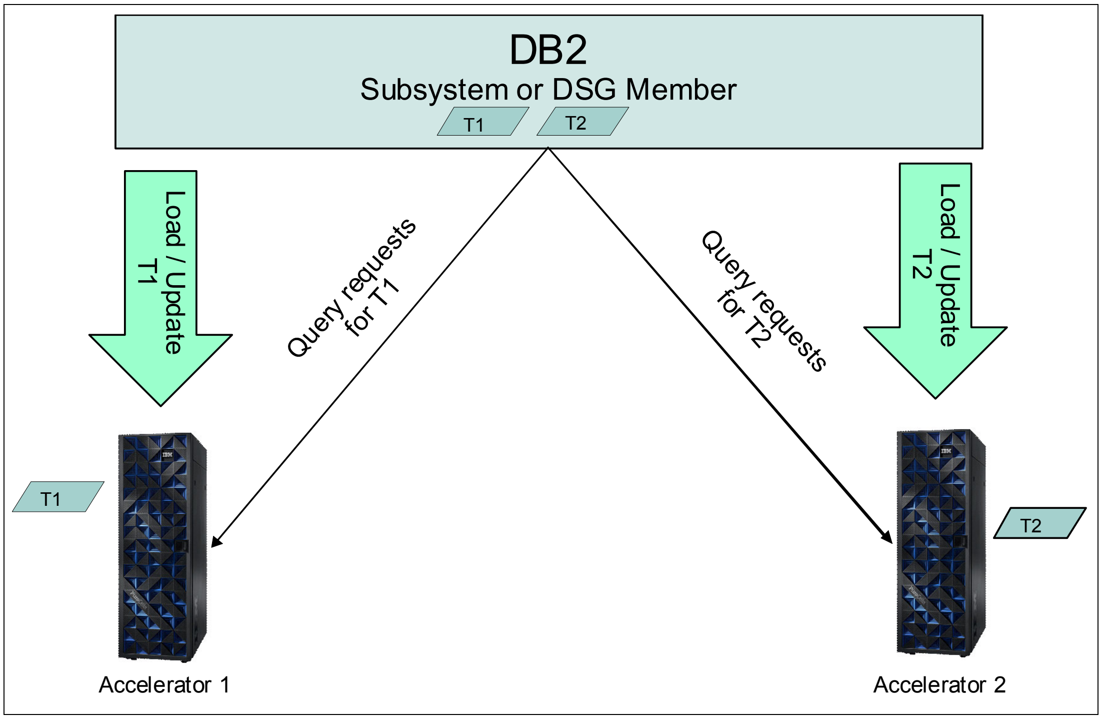

Figure 4 Two accelerators, maintained with a subset of the data

Figure 5 shows the failover processing when a disaster strikes Accelerator 1. Data for table T1 must be loaded into Accelerator 2 to satisfy all incoming query requests for tables T1 and T2 on Accelerator 2.

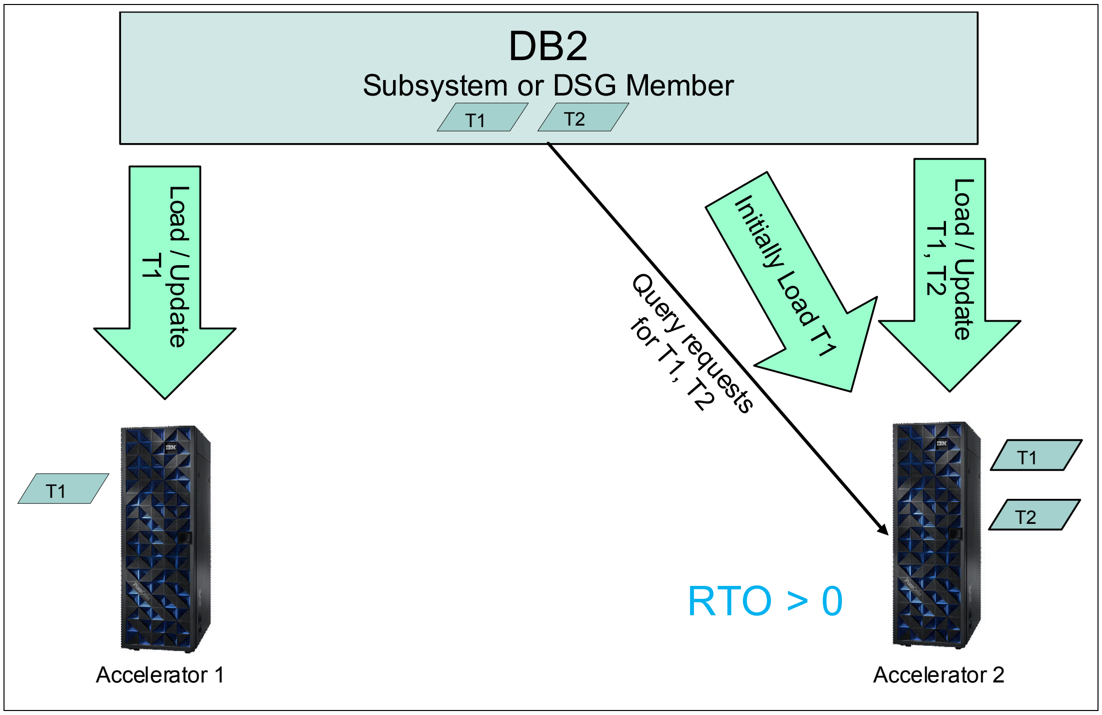

Figure 5 Data maintenance activities after a failure

HA setup for incremental update

Incremental update is a feature of IBM DB2 Analytics Accelerator where data changes within DB2 for z/OS can automatically be propagated to an accelerator by using DB2 for z/OS logs. It is based on InfoSphere® Changed Data Capture (CDC), which is part of the product packaging. This option is useful to track continuous changes in a DB2 for z/OS table without the need to run a full table or partition load into the accelerator. The underlying technology uses a capture agent that reads database logs and propagates changes to one or more apply agents. In the context of IBM DB2 Analytics Accelerator, the capture agent is on z/OS and the apply agent is inside the accelerator component.

Figure 6 shows a DB2 for z/OS data sharing group with two members. There is an active log capture agent on one DB2 for z/OS data sharing group member and it can read and process all logged changes for the entire data sharing group.

If updates should be propagated to multiple accelerators, it is suggested to implement the single log scrape capability that is called log cache so that DB2 log data is read only once and not multiple times for each incremental update target accelerator.

Figure 6 also shows a standby capture agent that is associated with another member of the DB2 data sharing group. If the active agent fails, this standby capture agent gets the active role. Because the components on the accelerator communicate with the active capture agent though a defined TCP/IP address, the active capture agent must always use the same TCP/IP address. This requirement includes the standby capture agent after taking over the active capture agent role. For this to occur, a dynamic virtual IP address is defined and bound to the active capture agent.

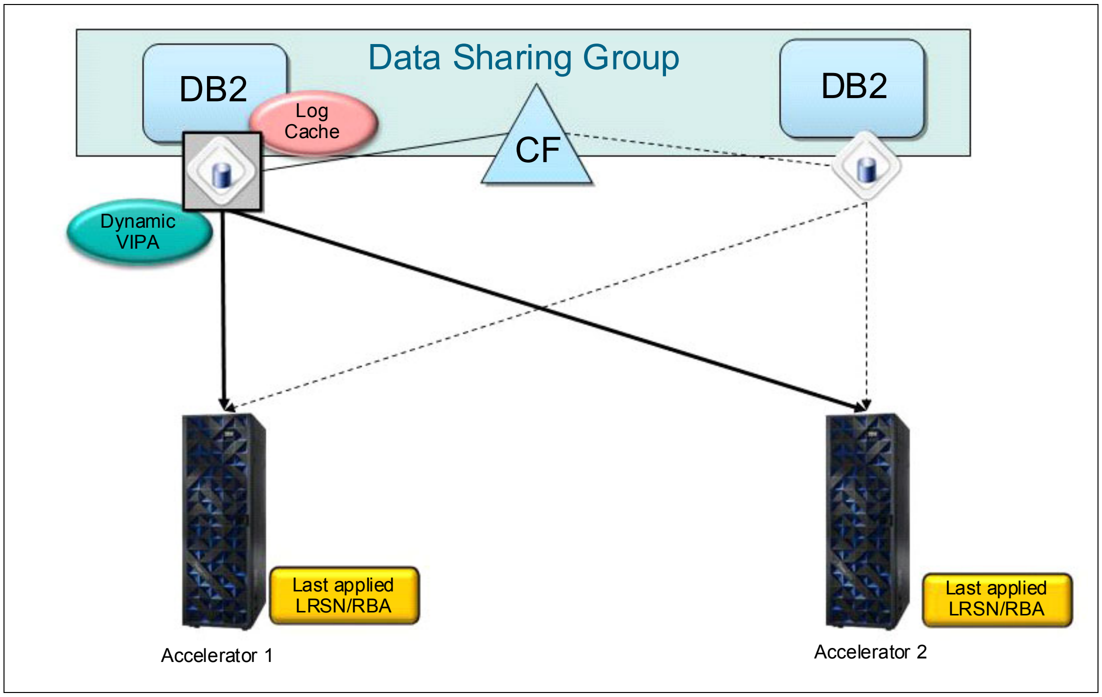

Figure 6   Incremental Update with multiple accelerators and active/standby capture agents

The apply agent component on the accelerator tracks applied changes and notes the last log record sequence number (LRSN) in data sharing or relative byte address (RBA) in non-data sharing, per replicated table. This information is used to continue consistent incremental update processing by requesting past changes from DB2 for z/OS logs if replication was interrupted for some time.

Figure 7 shows a failure or outage of the active capture agent log (as seen on the left side). This can be because of a problem with the capture agent software component, the DB2 subsystem, or the entire site.

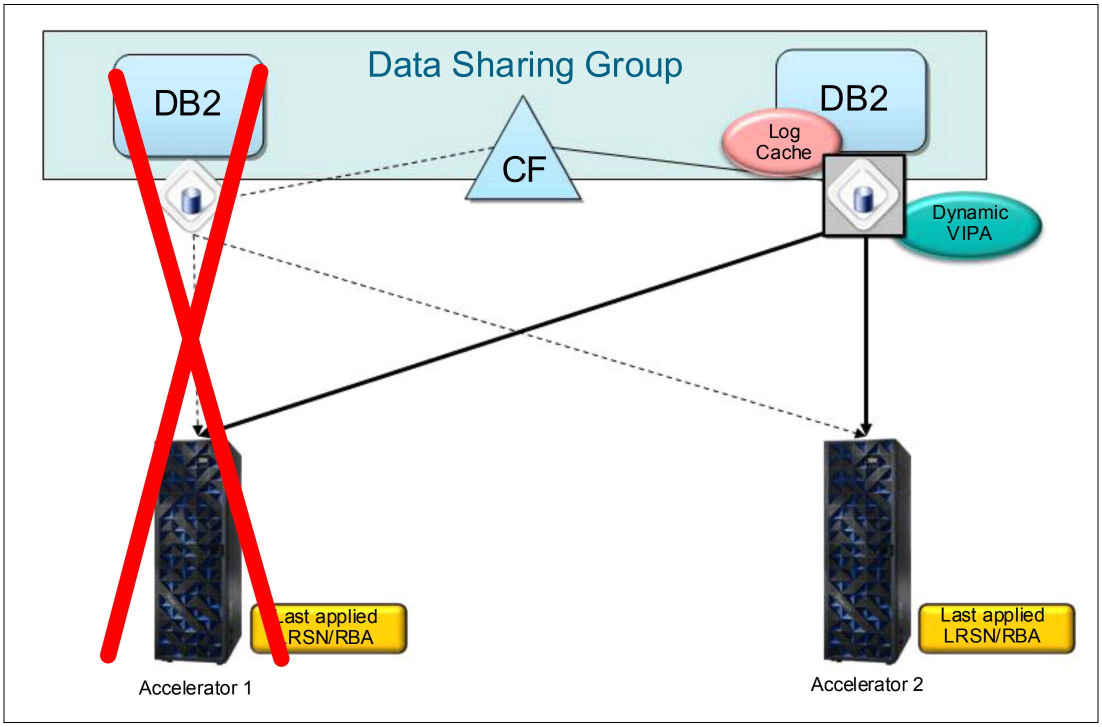

Figure 7 Incremental update after the active capture agent failed

The standby capture agent (as seen on the right side) is attached to another DB2 for z/OS member and running in another LPAR. This standby capture agent regularly monitors the active capture agent and takes over log capturing automatically if the active capture agent fails.

It also gets the dynamic virtual IP address so that the new active capture agent is accessible from the accelerator via the same IP address as before the failure.

Information regarding the last applied changes (LRSN/RBA) is used to fully and consistently be compensated by applying the changes since the last committed change on the accelerator.

Built-in consistency checks raise alerts if LRSN/RBA values in the accelerator no longer match with those in DB2 (for more information, see “Data maintenance using incremental update” on page 22).

For more information about the configuration of this setup, see the document at this website:

http://www-01.ibm.com/support/docview.wss?uid=swg27037912

The document also shows how dynamic virtual IP addresses can be set up in a way that the active capture agent continues to be accessible from the accelerator with the same IP address.

High-performance storage saver and multiple accelerators

IBM DB2 Analytics Accelerator can store historical data, which is static data that is not changed. If such data is stored in DB2 for z/OS, it occupies disk storage in your storage system that is attached to z/OS. By archiving this data to IBM DB2 Analytics Accelerator and querying data only on an accelerator, it is no longer required for this data (including indexes) to be stored on z/OS storage devices. The saved space on z/OS storage then can be reused for other purposes while data access to historical (*archived*) data is still enabled through the accelerator.

In the context of HA, it is necessary to implement a data strategy so that this archived data can still be accessed in a failure or disaster.

Since IBM DB2 Analytics Accelerator version 4, archived data can be loaded and accessed in multiple accelerators. Because archived data is only accessible for DB2 for z/OS native processing via image copies that are taken during archiving, it is essential to archive data on multiple accelerators to minimize *unplanned* data unavailability. With multiple accelerators present, after the initial archive process to a first accelerator, a subsequent archiving step to other attached accelerators creates a copy of this archived data in the other accelerators.

Image copies remain the primary backup vehicle and as such, should also be available and accessible on a disaster recovery site.

Figure 8 shows a table with six partitions. Partitions 1 and 2 have active data, which remains on DB2 for z/OS storage and may be changed. Partitions 3 - 6 have historic (static) data, which does not change and is archived to the accelerators.

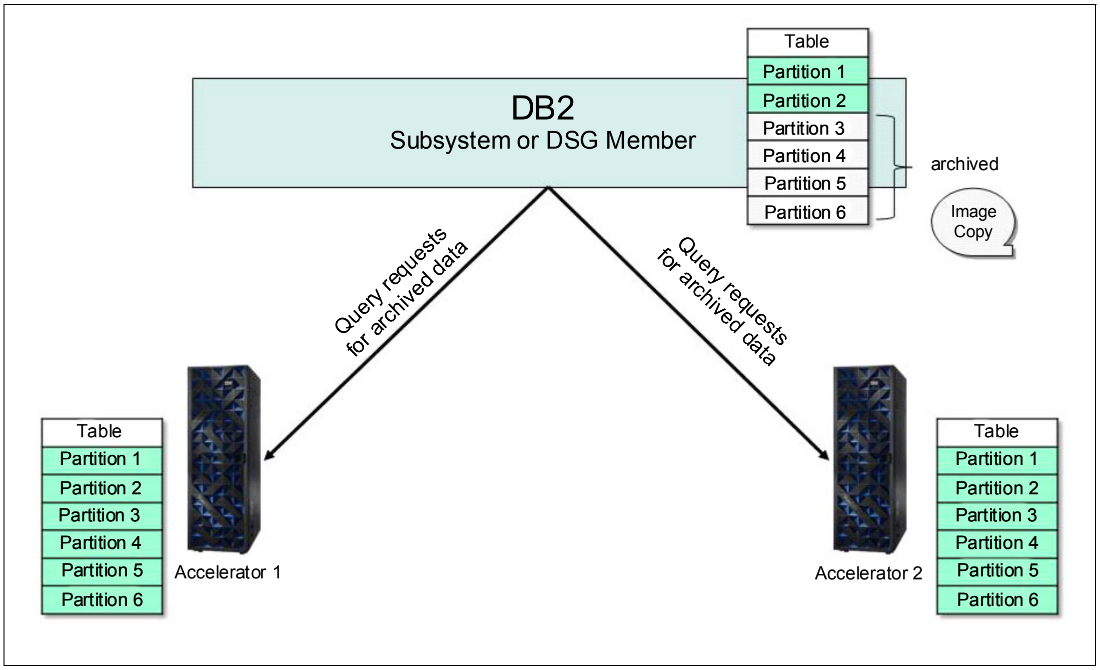

Figure 8 High-performance storage server with multiple accelerators

In a first step, an image copy of the archived data (one per partition) is created and the data is stored on Accelerator 1. Functionality to archive data on an accelerator is encapsulated in stored procedure ACCEL_ARCHIVE_TABLES. In a second step, by calling stored procedure ACCEL_ARCHIVE_TABLES again, the same archived data is also stored on Accelerator 2. Because the data is not available in DB2 for z/OS for unloading, a subsequent invocation of the same stored procedure unloads the data from the image copy that was created during the first invocation of the same stored procedure for the same object.

With this setup, query requests for archived data can be satisfied by both active accelerators. If Accelerator 1 fails, archived data is still accessible though Accelerator 2.

However, to automatically restore archived data by using IBM DB2 Analytics Accelerator's stored procedure (ACCEL_RESTORE_ARCHIVE_TABLES) from an accelerator to DB2 for z/OS, all accelerators that contain these archived partitions must be available. If data must be restored without all required accelerators being available, the following manual steps are needed to recover data into DB2 for z/OS that is based on existing image copies that were taken at archiving time:

1. Reset PRO state from archived partitions in DB2 for z/OS.
2. Recover to image copy taken at archiving time.
3. Rebuild affected indexes.

After a failing accelerator becomes available again, archived data must be removed from this accelerator to ensure all catalog-entries are synchronized.

> **Note:** Even if maintaining all data on another accelerator is not feasible because of required CPU costs and elapsed times, you should consider archiving data on all available accelerators. This configuration helps avoid manual intervention to bring archived data to another accelerator if there is a failover.

Shared usage of a second accelerator

Multiple DB2 for z/OS subsystems can be connected to a single accelerator. An accelerator provides workload management capabilities to an extent that minimum and maximum resource allocations within an accelerator can be dedicated to connected DB2 for z/OS subsystems. One example is sharing a single accelerator between development and production DB2 for z/OS subsystems where both subsystems are entitled to use 50% of the accelerator resources, as shown in Figure 9 on page 14.

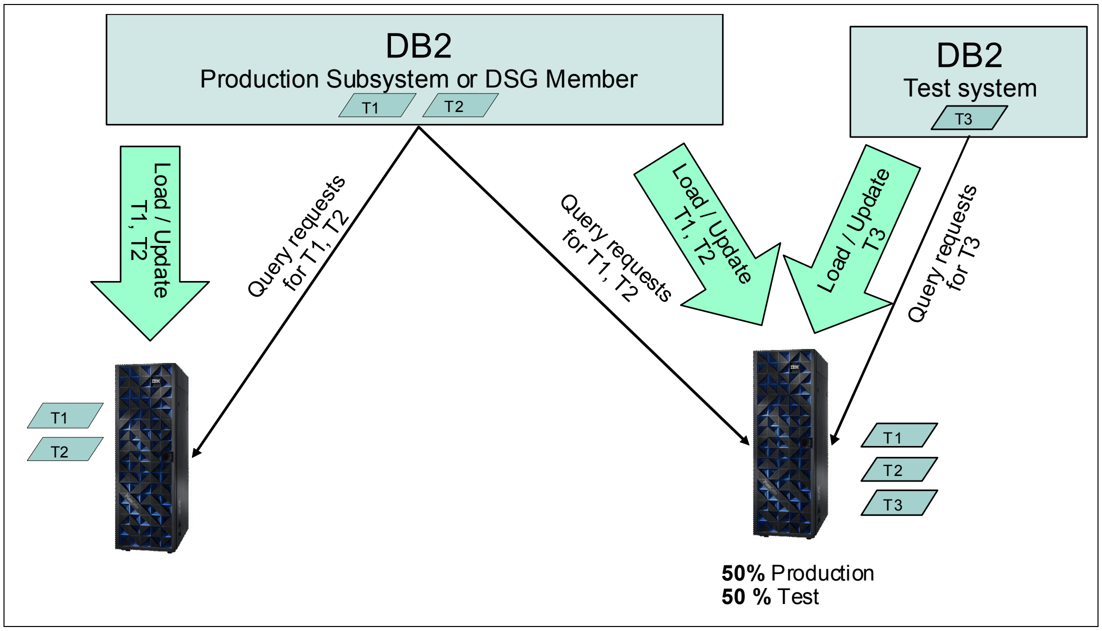

Figure 9 Mixed set up with a shared accelerator

If a disaster strikes Accelerator 1 that is solely used for production workload, Accelerator 2 can be reconfigured to dedicate a higher percentage of its resources to query workloads from the production DB2 for z/OS subsystem. Dedicated resources for the development of the DB2 for z/OS subsystem are reduced or fully eliminated, depending on individual requirements. Reconfiguration of resource assignments is not an automated process; it requires manual interaction if there is a failover.

Figure 10 on page 15 shows the scenario after resource assignments of a shared accelerator between production and development workloads are changed. In our example, no resources (0%) are left for ongoing development or test workloads and this can be achieved by disabling query acceleration for the test system. Reducing the previous allocation of 50% for the test system (for example, to 5%) can be implemented by changes through the DB2 Analytics Accelerator Console.

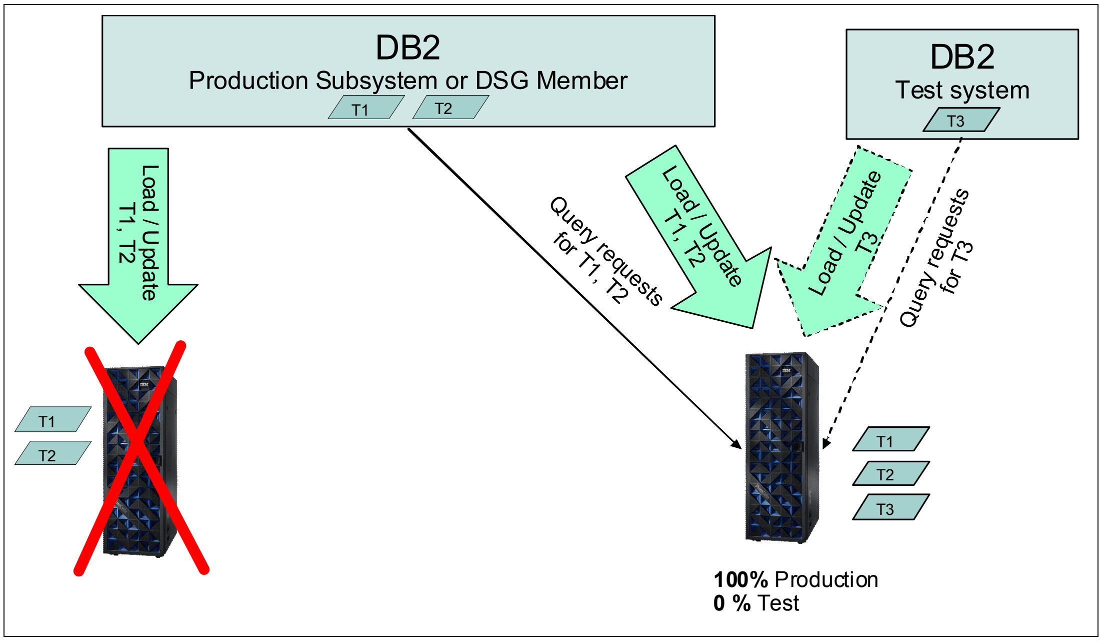

Figure 10 Changes for a shared accelerator after failure

Stand-by accelerators

For most scenarios that are described in "Planning for HA and DR with IBM DB2 Analytics Accelerator", implementing multiple active accelerators with in-sync data is essential. The benefit is optimal use of accelerator resources and optimal RPO and RTO if there is an unplanned outage. These benefits include other costs because data must be maintained in multiple accelerators.

Stand-by systems can be implemented for DB2 for the following reasons:

- One reason is related to latency between sites. Longer distances between sysplex members affect transaction times because I/O operations must wait until they are committed on the remote site.
- The other reason is related to licensing. A dormant DB2 for z/OS subsystem causes fewer MLC costs than an active one.

Both reasons do not apply to accelerators. Network latency often is not an issue for analytical query processing and the price model for an accelerator is not MLC-based.

Consider a stand-by accelerator for the following reasons:

- Lack of network bandwidth between primary and remote site to keep data in-sync between accelerators.
- Non-critical application where a longer recovery time objective (hours to days) is acceptable so that loading the accelerator after a failure is feasible.

Planning for HA and DR with IBM DB2 Analytics Accelerator

We observe two paradigms when dealing with unplanned outages. The first paradigm is to plan for continuous availability of IT operations if disaster strikes the data center. A typical implementation for continuous availability is the GDPS/Peer-to-Peer Remote Copy (PPRC) architecture, which uses an active-active configuration. The second paradigm is to plan for minimal downtime and a low (but acceptable) data loss to resume normal operations. The related architectures with IBM DB2 Analytics Accelerator we describe are GDPS/PPRC that uses an active-standby configuration and GDPS/Extended Remote Copy (XRC).

In the following sections, we describe the most common GDPS configurations and how IBM DB2 Analytics Accelerator can be integrated into them by applying established concepts from previous sections of this paper.

Latency and bandwidth requirements

Network latency and bandwidth requirements for IBM DB2 Analytics Accelerator differ from requirements in a Parallel Sysplex and a data sharing group. Because GDPS/PPRC implements a guaranteed and synchronous commit protocol for primary and secondary data, network latency affects transaction response times. Practically, this limits HA environments to a distance of less than 20 km. It is important to understand that an accelerator does not participate in any kind of two-phase commit protocol and, therefore, network latency does not have a significant effect on its operation. A query routing request from DB2 for z/OS to an accelerator might take milliseconds longer, but with query response times in the range of seconds, minutes, or even hours, this usually is not considered to be significant.

However, it is important to provide significant network bandwidth between DB2 for z/OS and the accelerator to allow for data loading and large query result list processing in a reasonable time. That is why a 10 Gbps connection is a standard prerequisite to operate an accelerator.

Definition of accelerators in DB2 catalog tables

If a DB2 for z/OS data sharing group was implemented to provide HA capabilities, it is important to understand how and where accelerator definitions are stored. During the so-called pairing of a DB2 for z/OS subsystem with an accelerator, all configuration parameters are stored in the communication database (CDB) within DB2 for z/OS catalog tables and other pseudo catalog tables. This includes IP addresses of accelerators and an authentication token, which grants access to the data of a paired DB2 for z/OS subsystem or a data sharing group. Because all members of a data sharing group share these catalog entries, all of them automatically have access to a shared accelerator and the data from the respective data sharing group. Assuming that proper installations are completed on all members (for example, stored procedure setup), no other configuration is required if another member (active or standby) wants to access data on an accelerator.

For a disaster recovery setup, GDPS can be used to replicate information of an entire DB2 for z/OS subsystem to a remote site. This usually includes DB2 for z/OS setup information and active catalog table content.

If such a dormant DB2 for z/OS subsystem is brought up, it uses definitions in the catalog table to reach out to any connected accelerator, which must be considered.

GDPS/PPRC active-active configuration

In a GDPS/PPRC active-active configuration (as shown in Figure 11), production systems are running in both sites. Both sites have access to the same data that is mirrored at the storage disk level (DASD volumes), from primary to secondary DASD volumes. Updates that are made on the primary DASD volumes are synchronously shadowed onto the secondary DASD volumes. The DB2 for z/OS write operation is only completed when the data is secure on the secondary site's storage system. Required serialization mechanisms are implemented by using one or more data sharing groups that span both sites and include DB2 for z/OS members from both sites.

Figure 11 also shows a setup with two sites ("Site 1" and "Site 2") where primary disks are on Site 1 and secondary disks are on Site 2. Both sides are active and process workloads. If there is an unplanned outage on Site 1, data is accessed from the secondary disks and capacity on Site 2 is used to process the entire workload.

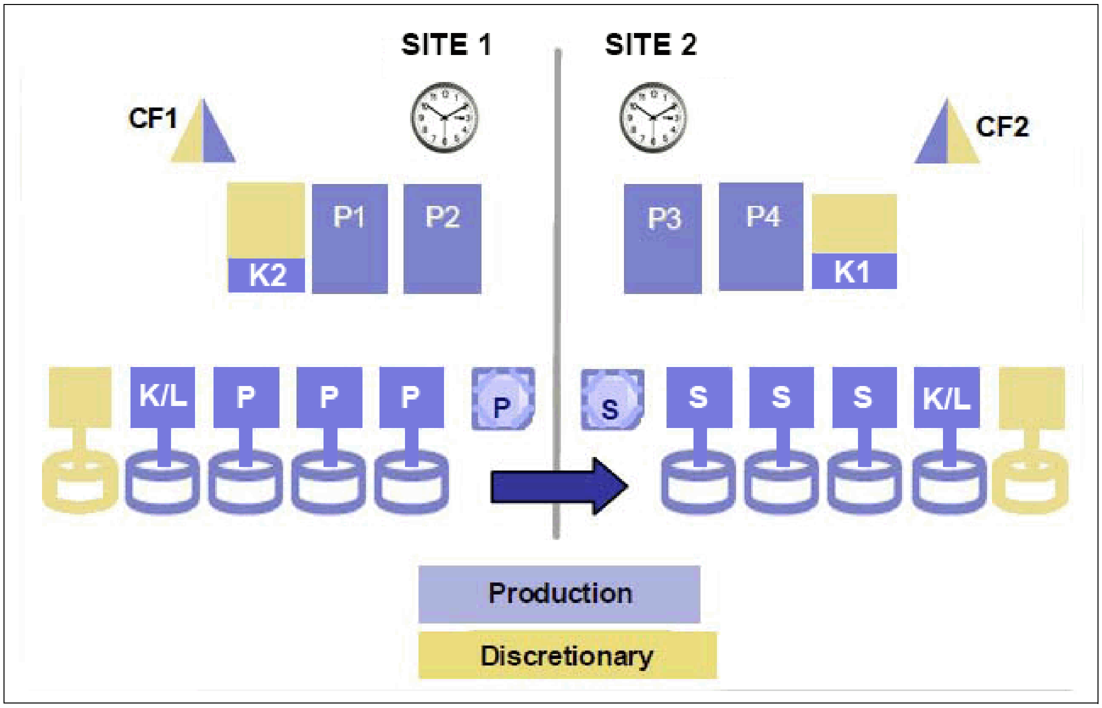

Figure 11 GDPS/PPRC active-active configuration

Because in this example we have production systems in both sites, we must provide the capability to recover from a failure in either site. Therefore, in this case, there also is a GDPS controlling system (K1 and K2) that is available to decide how to react to that failure and what recovery actions are to be taken.

GDPS/PPRC active-active configuration with accelerator

In a GDPS/PPRC active-active configuration, one or more accelerators are connected both to a System z server at the primary and secondary site. By using the concept of data sharing groups, all accelerators are visible to all DB2 for z/OS members on the primary and secondary site, as shown in Figure 12 on page 18.

An example of a data sharing group spanning two sites (Site 1 and Site 2) with DB2 members in each site is shown in Figure 12 on page 18. Primary disks are in Site 1 and secondary disks in Site 2. The suggested setup includes two accelerators, one for each site. Both of the sites are connected via switches to the DB2 members.

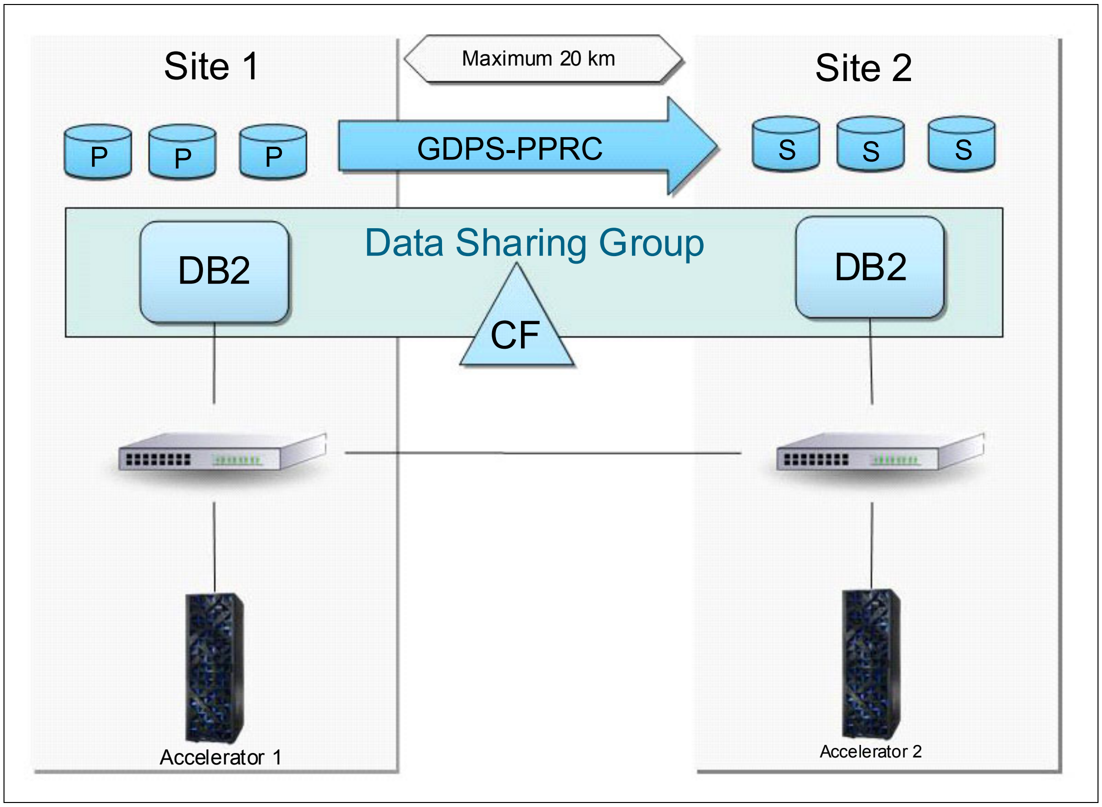

Figure 12   GDPS/PPRC active-active configuration with accelerators

As described in “Data maintenance and synchronization with multiple accelerators” on page 6, the preferred option is to actively use all available accelerators for query workloads in both sites if resources are available to maintain data in both accelerators.

Figure 13 on page 19 shows how query workload flows with such a setup. The active-active configuration allows the acceptance of requests in both sides. The respective DB2 for z/OS subsystems then chose dynamically which accelerator is best-suited to process the query request. The workload balancing feature that is described in “Workload balancing” on page 4 ensures that workload is distributed among the connected accelerators.

Remember that the distance between the sites often does not impose a problem for query processing. In practice, users do not notice a difference if a query was processed on Accelerator 1 or Accelerator 2.

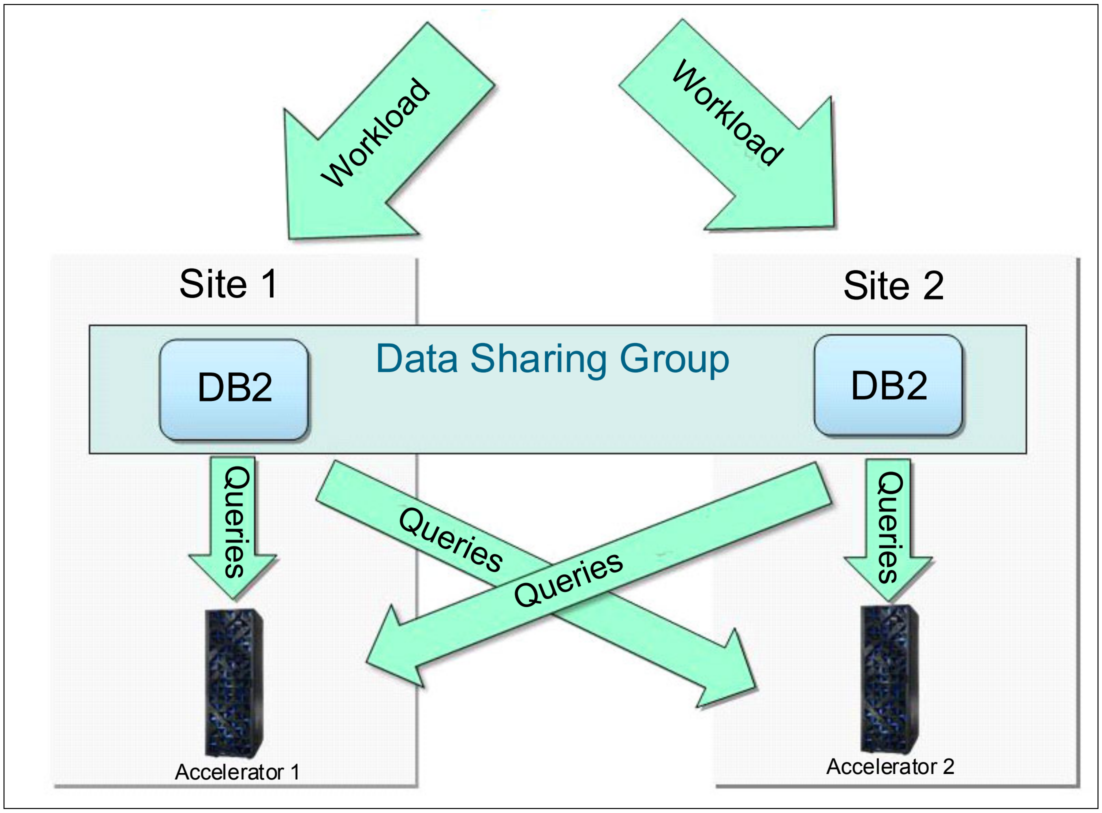

Figure 13 Workload flow with GDPS/PPRC active-active configuration and accelerators

Assume a failure at Site 1 can affect the entire site or certain components. Figure 14 on page 20 shows the scenario where the entire Site 1 is unavailable. The components in Site 2 take over, the secondary disks become active, and the DB2 member (or members) in Site 2 no longer use Accelerator 1.

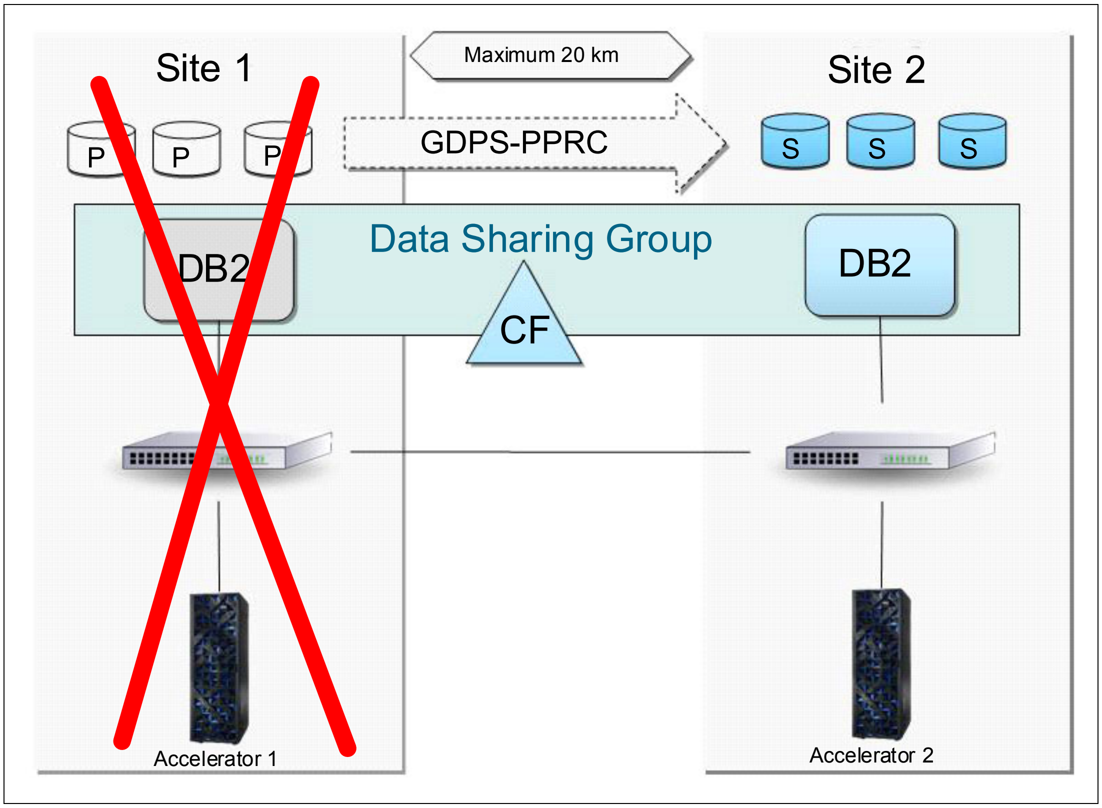

Figure 14 GDPS/PPRC active-active configuration with accelerator after failure

The query workload (as shown in Figure 15 on page 21) go entirely to Site 2. If Accelerator 1 also is affected by the outage, the DB2 subsystem in Site 2 automatically detects that Accelerator 1 is unavailable and continues with the remaining query workload only on Accelerator 2.

If Accelerator 1 is available and only the DB2 subsystem on Site 1 is down, the picture changes and both accelerators are used.

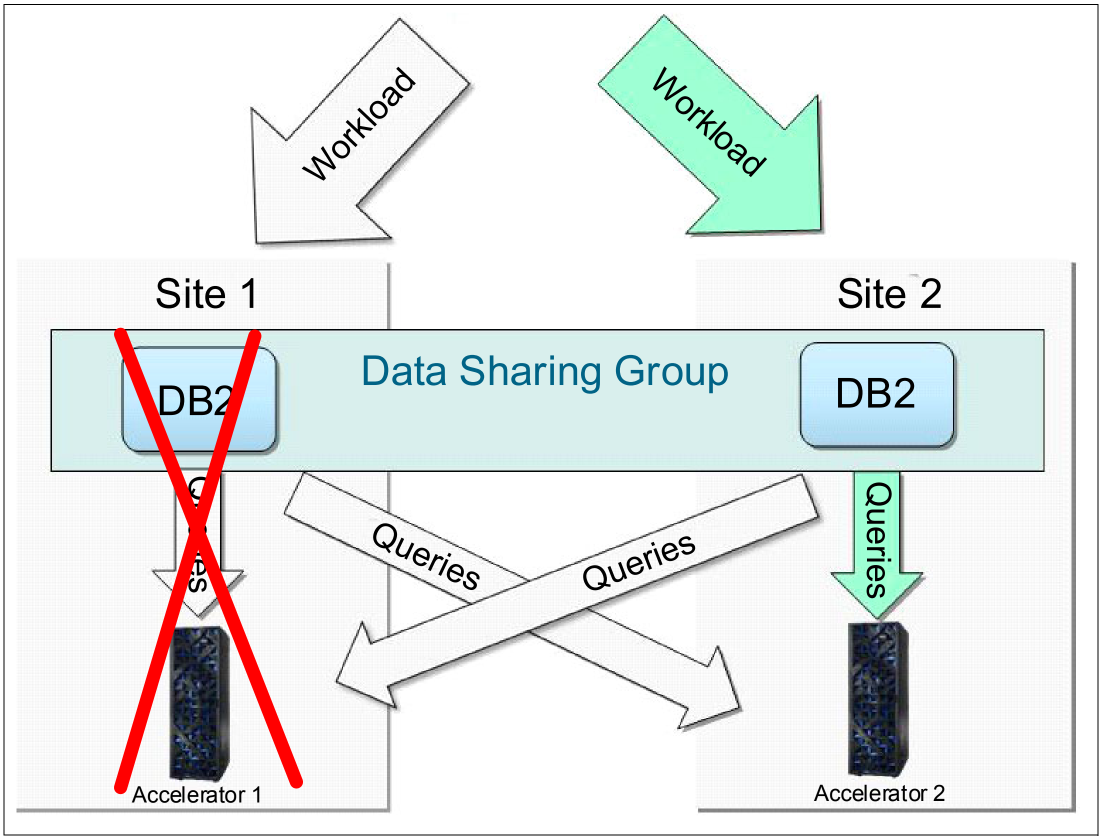

Figure 15 Workload flow with GDPS/PPRC active-active configuration and accelerators after failure

Data maintenance using regular load cycles

By applying the same data maintenance processes to all connected accelerators, the same data is stored on all connected accelerators. This configuration allows DB2 for z/OS to balance incoming query requests between accelerators, which results in a balanced usage of all accelerators and best possible response times by even workload distribution to existing appliances. However, if the stored procedure (ACCEL_LOAD_TABLES) is called multiple times to load data into more than one accelerator, it is likely that all invocations complete at a different time. This means that the newly loaded data is available first on one accelerator, then on the other. The time difference between completing both stored procedure calls shows queries to obtain different result sets. There is no automation in place to prevent this from happening. If different results cannot be tolerated for the time while data is loaded into one but not into another accelerator, a potential mechanism to work around this issue is described in “Data consistency across multiple accelerators” on page 7.

Because all connected accelerators can be reached from each site and all accelerators contain the same data for regularly loaded tables, query acceleration remains active even if one site is not available.

This concept allows for RPO = 0 including query acceleration and maintains the same RTO as it is achieved without an accelerator in the same configuration.

Data maintenance using incremental update

The suggested option to keep both accelerators in sync also applies to the concept of incremental update. As described in “Data maintenance and synchronization with multiple accelerators” on page 6, capture agents must be installed in both sites. One site has the active capture agent, while the other site runs a standby capture agent. The active capture agent propagates changes to all active accelerators (Accelerator 1 and Accelerator 2 in the example). If there is a failure, the standby capture agent in Site 2 becomes active and continues with incremental update processing.

This concept is not limited to a single data sharing group. Multiple DB2 data sharing groups or stand-alone DB2 for z/OS subsystems (with their own capture agent instance) can propagate their changes to multiple accelerators.

High-performance storage saver

For a GDPS/PPRC active-active configuration, all archived data should be stored in all active accelerators, as described in “High-performance storage saver and multiple accelerators” on page 12. This configuration allows for continuous processing and for requests that access archived data.

GDPS and the FREEZE and GO option

If an RPO that is not necessarily zero is acceptable, you might decide to allow the production systems to continue to operate after the secondary DASD volumes are protected by the Freeze. In this case, you use a FREEZE and GO policy.

For more information about the different FREEZE options, see *GDPS Family An Introduction to Concepts and Facilities*, SG24-6374-08.

For DB2 Analytics Accelerator, a FREEZE and GO implementation with data loss means that changes might not be made to the secondary DASD volumes. If there is a disaster at the primary site, the synchronization status between the surviving state and the accelerator is unclear. Although it puts another challenge on top of the data loss, a reload of the data in the accelerator must be considered in this situation. As described in “HA setup for incremental update” on page 10, this situation can be discovered and corrective actions can be put in place.

GDPS/PPRC active-standby configuration

In a GDPS/PPRC active-standby configuration (as shown in Figure 16 on page 23), production systems are running in the primary site. Data from the primary site is mirrored at the DASD volume level. Updates that are made on the primary DASD volumes are synchronously shadowed onto the secondary DASD volumes. The DB2 for z/OS write operation is completed only when the data is secure on the secondary site’s storage system (DASD volumes). The DB2 for z/OS subsystem on the secondary site is a clone of the DB2 for z/OS subsystem on the primary site.

If there is an outage in Site 1, systems are brought up in Site 2 and continue processing requests.

The LPARs in blue in Figure 16 on page 23 (P1, P2, P3, and K1) are in the production sysplex, as are the Coupling Facilities CF1 and CF2. The primary disks (DASD volumes) are all in Site 1, with the secondaries in Site 2. All of the production systems are running in Site 1, with the GDPS controlling system (K1) running in Site 2.

System K1's disks (those marked K) are also in Site 2. The unlabeled boxes represent work that can be displaced, such as development or test systems.

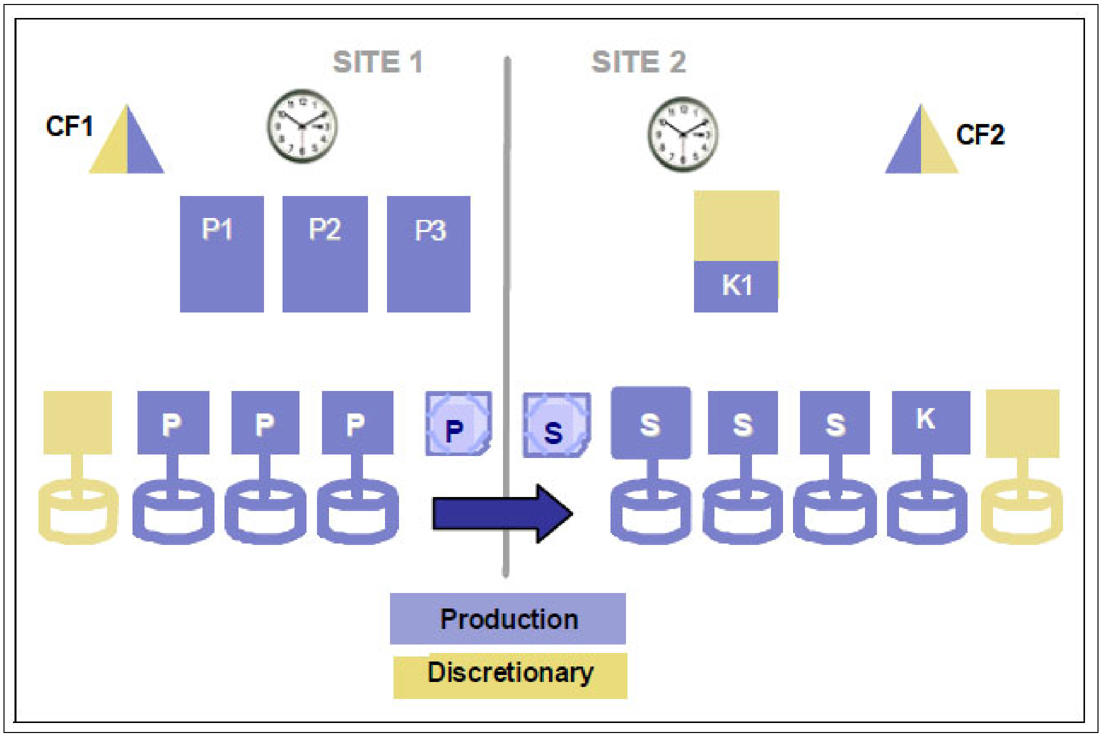

Figure 16 GDPS/PPRC active-standby configuration

GDPS/PPRC active-standby configuration with accelerator

In a GDPS/PPRC active-standby configuration, one or more accelerators are connected to a System z server at the primary and secondary sites. All accelerators are visible to all DB2 for z/OS members on the primary and secondary sites, as shown in Figure 17 on page 24.

The suggested configuration for this setup has two active accelerators. Although the DB2 for z/OS subsystem in Site 2 is in standby mode, Accelerator 2 in Site 2 is active.

To support a quick failover if there is a failure, data on the second accelerator must be kept updated. Except from considerations such as power consumption, it is advantageous to use the second accelerator for regular processing as well.

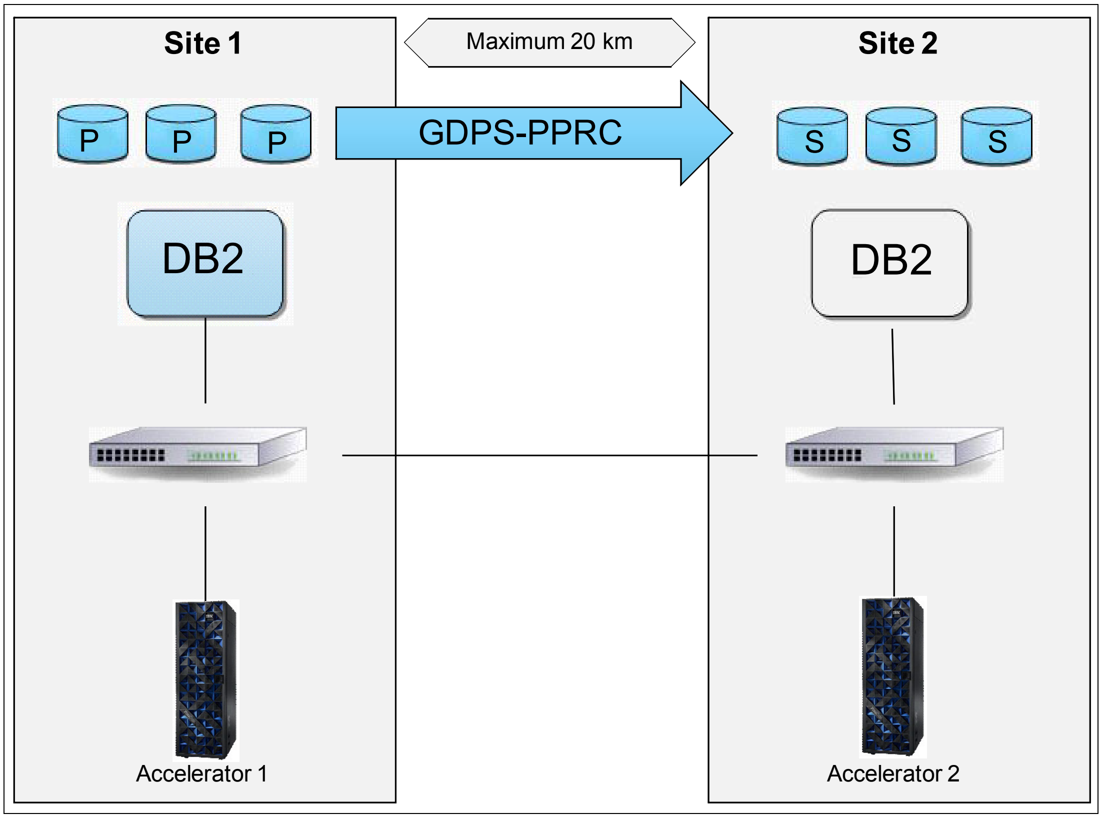

Figure 17   GDPS/PPRC active-standby with accelerators

Figure 18 on page 25 shows how queries are processed with this setup. Only the active DB2 for z/OS subsystem in Site 1 accepts requests. Because Accelerator 1 and Accelerator 2 are updated and active, query routing considers both accelerators.

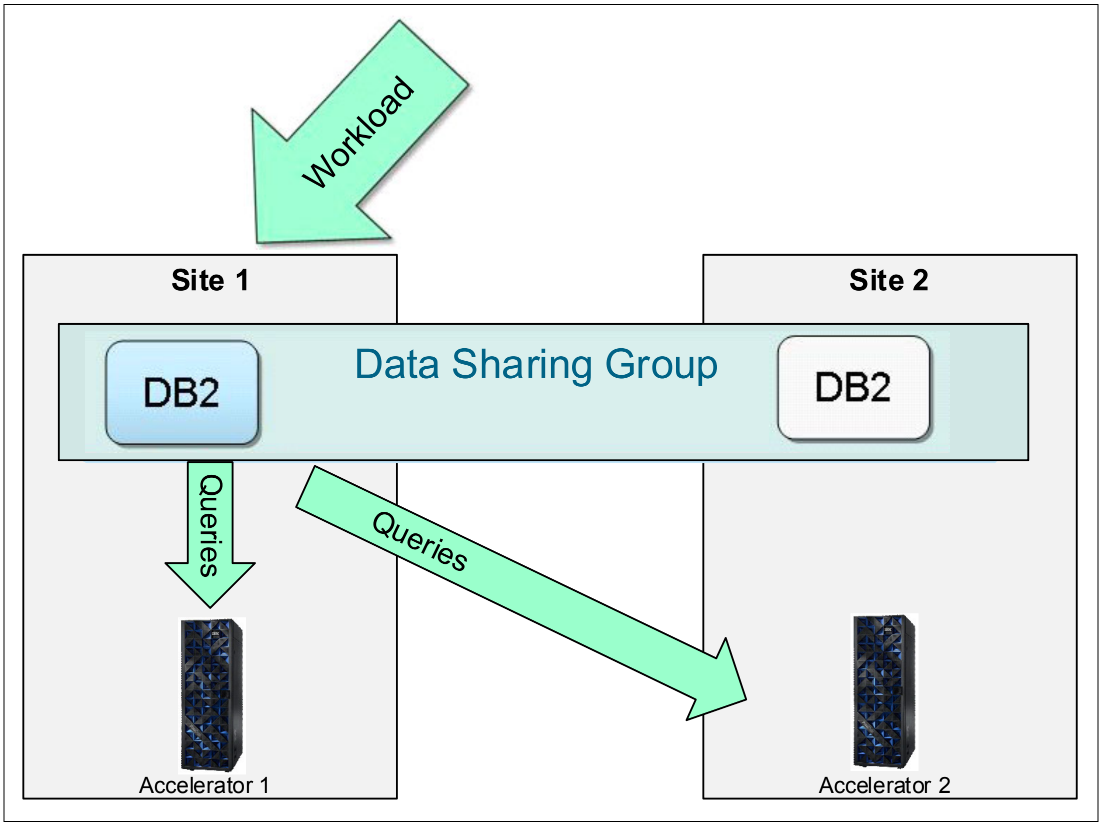

Figure 18 Workload flow with GDPS/PPRC active-standby and accelerators

After a failure of active Site 1, the picture changes, as shown in Figure 19 on page 26. The passive DB2 for z/OS subsystems in Site 2 take over, which access the secondary DASD volumes. The shared accelerator definitions in the DB2 catalog tables ensure that the standby DB2 for z/OS subsystem can seamlessly access the previously configured accelerators.

If the DB2 for z/OS subsystem in Site 2 is a clone of the DB2 for z/OS subsystem in Site 1, all information to access both accelerators also is available after a failover to Site 2. This is because that accelerator access information is stored in DB2 for z/OS catalog and pseudo-catalog tables. It is not mandatory to operate in data sharing mode to benefit from this concept.

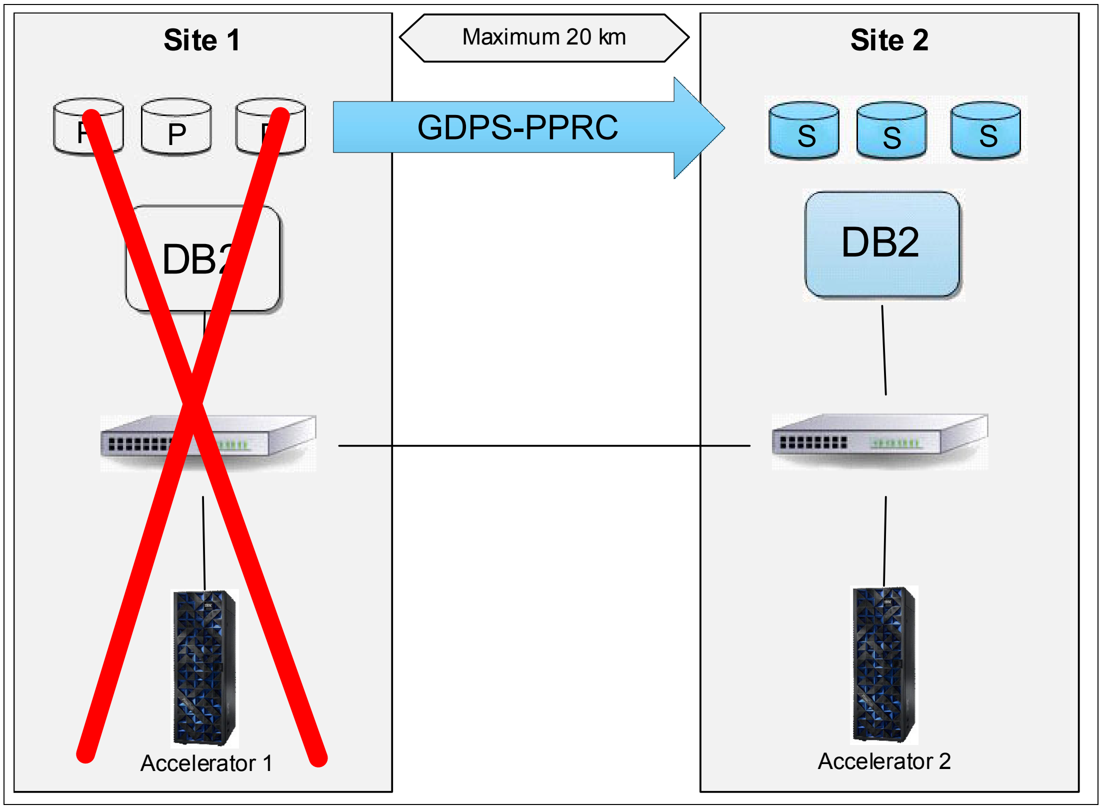

Figure 19 GDPS/PPRC active-standby configuration with accelerators after failure

Query workload (see Figure 20 on page 27) is then accepted by only DB2 for z/OS subsystems in Site 2. The now activated DB2 for z/OS subsystem accesses and uses Accelerator 2. Depending on the scope of the outage in Site 1, it also can use Accelerator 1 (if it is not affected by the outage).

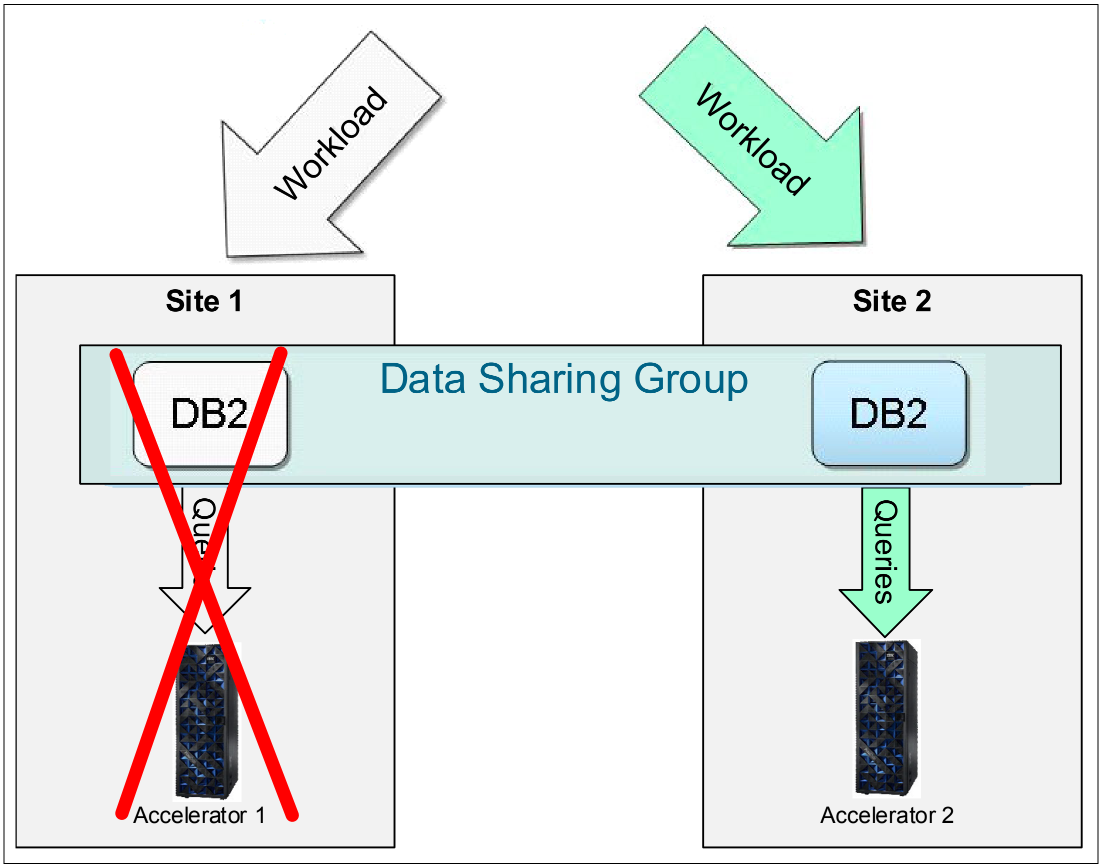

Figure 20 Workload flow with GDPS/PPRC active-standby configuration and accelerators after failure

Accelerator data maintenance for load and incremental update are the same as described with GDPS active-active in sections “Data maintenance using regular load cycles” on page 21 and “Data maintenance using incremental update” on page 22.

The concepts for high-performance storage servers that are described in “High-performance storage saver” on page 22 also apply here in the same way.

GDPR/XRC active-standby configuration

A GDPS/XRC is an active-standby configuration by default, as shown in Figure 21 on page 28. Production workload is processed only in the primary site. The DB2 for z/OS write operations are completed in the primary site and data is then asynchronously duplicated at the DASD volume level in a secondary site. Because of its asynchronous data duplication nature, GDPS/XRC can virtually span unlimited distances between the primary and secondary sites.

In a disaster, the DB2 for z/OS subsystem on the secondary site must be brought up. This leads to an RTO > 0. Because of its asynchronous data mirroring technique, a GDPS/XRC architecture comes with RPO > 0 and some data loss might occur.

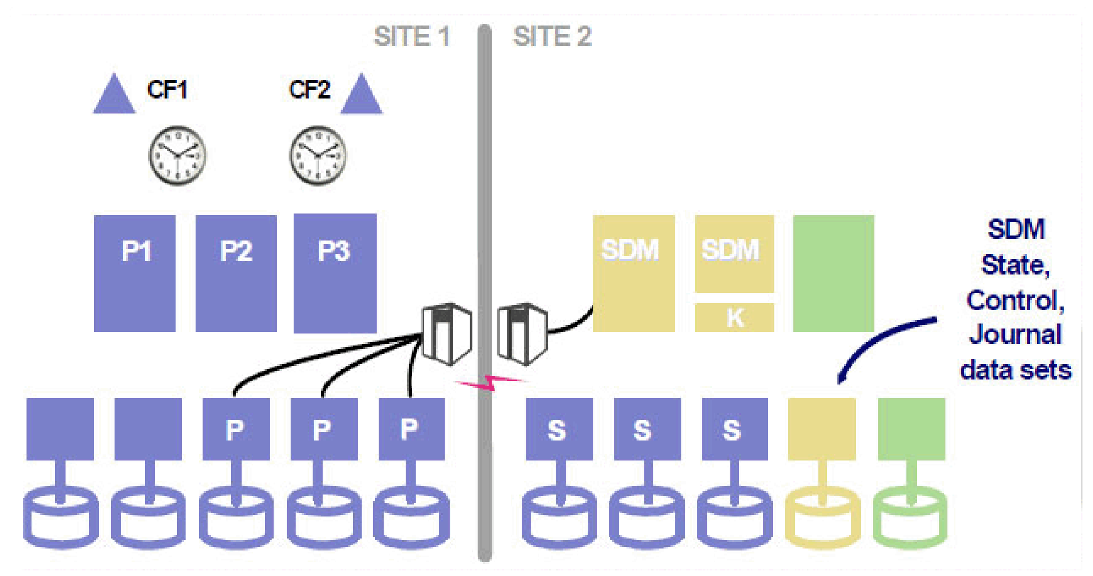

Figure 21 GDPS/XRC configuration

GDPR/XRC active-standby configuration with accelerator

The suggested configuration for this setup depends on available network bandwidth between both sites. In our scenario, we assume that network bandwidth is sufficient and that accelerators on both sites can be operated and maintained as active accelerators. Scenarios in which bandwidth is insufficient and do not provide capabilities to operate an accelerator at the secondary site as part of the primary site's IT services that are described in “GDPS/XRC with limited bandwidth to remote accelerator” on page 32.

Figure 22 on page 29 shows the recommended configuration with two accelerators in both sites that use GDPS/XRC.

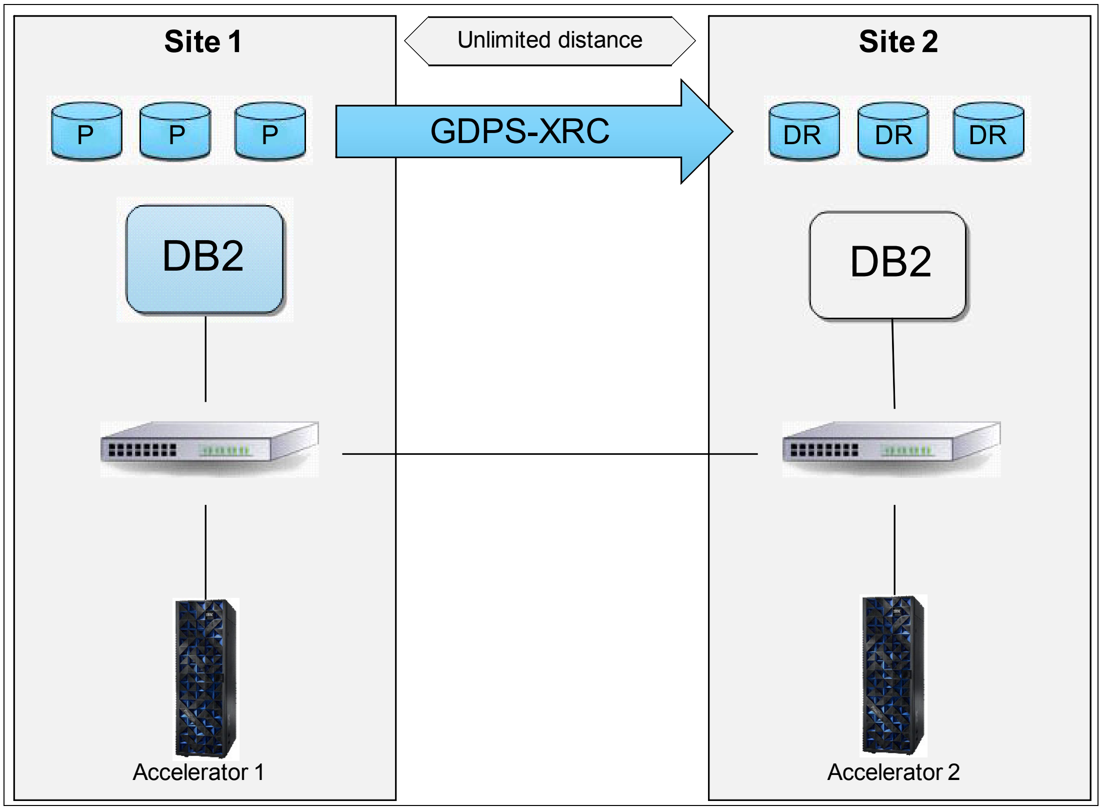

Figure 22 GDPS/XRC with accelerators

Incoming query requests in Site 1 use Accelerator 1 and Accelerator 2 as both accelerators can be reached from both sites. Even if the distance for GDPS/XRC exceeds supported distances for GDPS/PPRC architectures, bandwidth might be enough to allow for query requests and accelerator maintenance over long distances from Site 1. Credentials for both accelerators are stored in catalog and pseudo-catalog tables of the DB2 for z/OS subsystem in Site 1. Because of data duplication to Site 2, the same credentials can be used if there is a failover after the DB2 for z/OS subsystem is started in Site 2. Figure 23 on page 30 shows how workloads are processed in a GDPS/XRC environment.

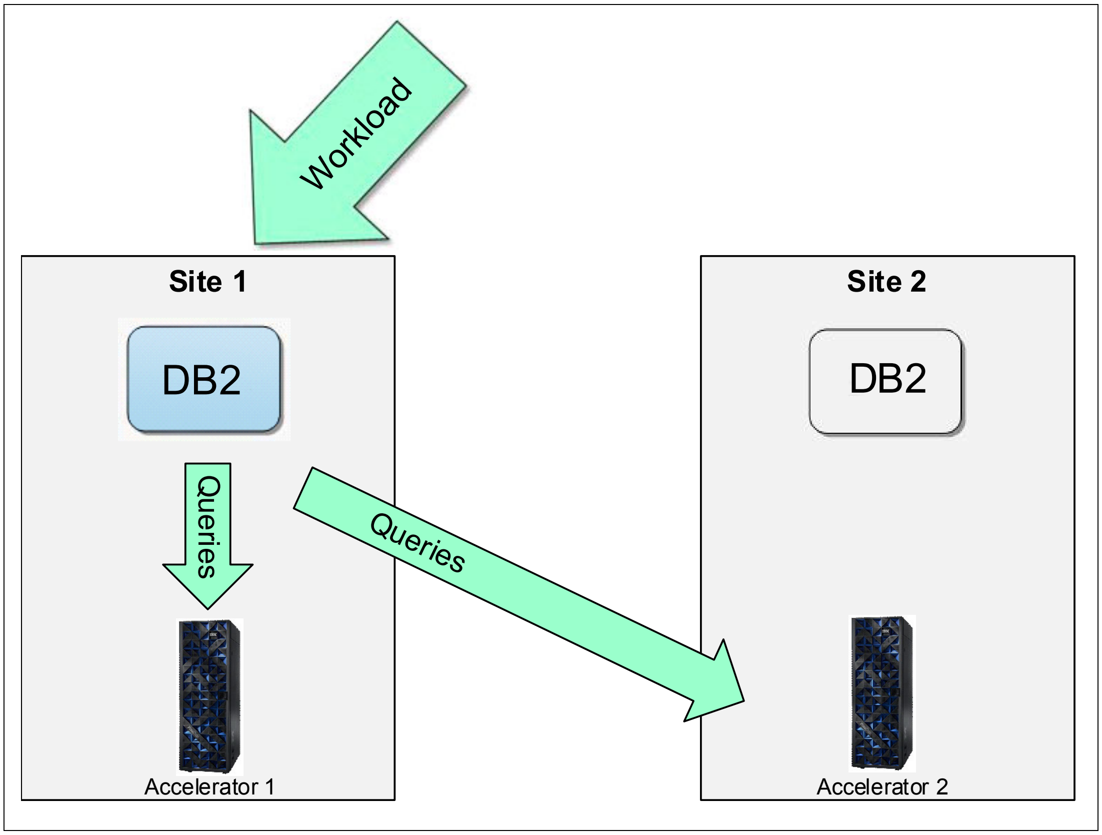

Figure 23 Workload flow with GDPS/XRC and accelerators

If a disaster strikes Site 1, the DB2 for z/OS subsystem in Site 2 is brought up. Data that was not duplicated at DASD volume level between the last data duplication cycle and the disaster is lost. Because catalog and pseudo-catalog tables also are available to the DB2 for z/OS subsystem in Site 2, both accelerators can be used instantly. The scenario of GDPS/XRC with accelerators after a failover is shown in Figure 24 on page 31.

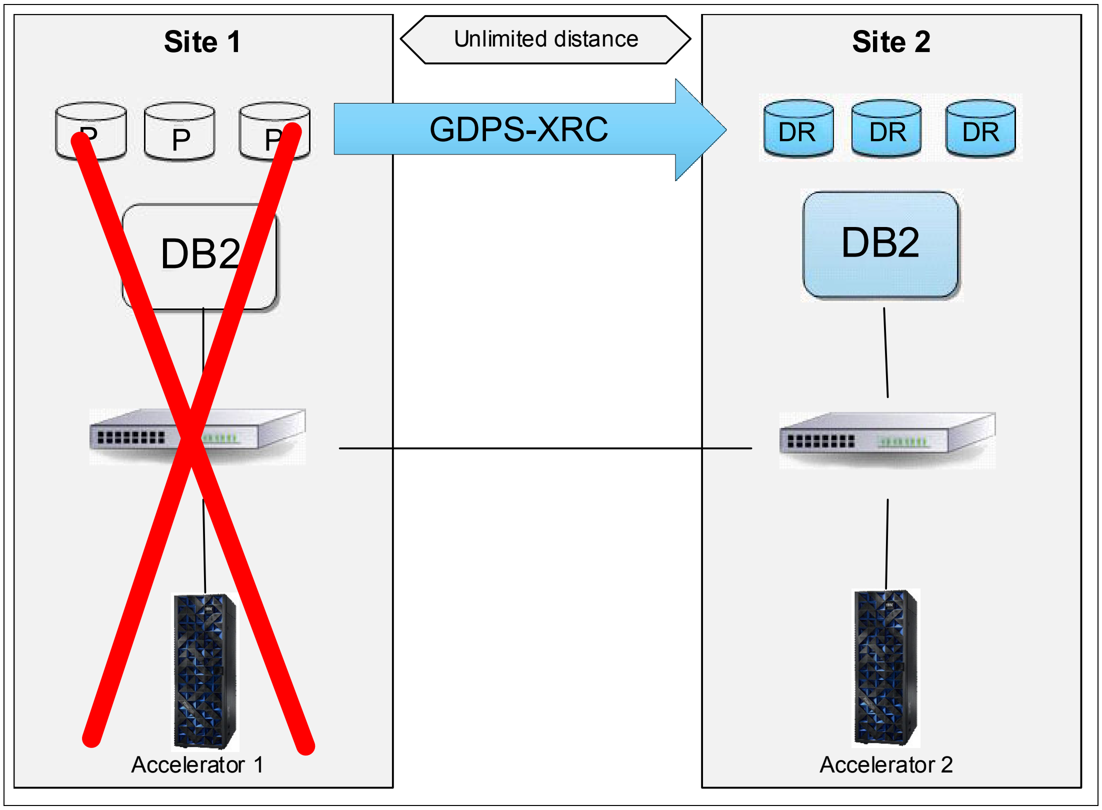

Figure 24 GDPS/XRC with accelerators after failure

Figure 25 on page 32 shows the workload processing after a failover. If the disaster affects only the DB2 subsystem and Accelerator 1 is still accessible, Accelerator 1 continues to serve query requests from Site 2.

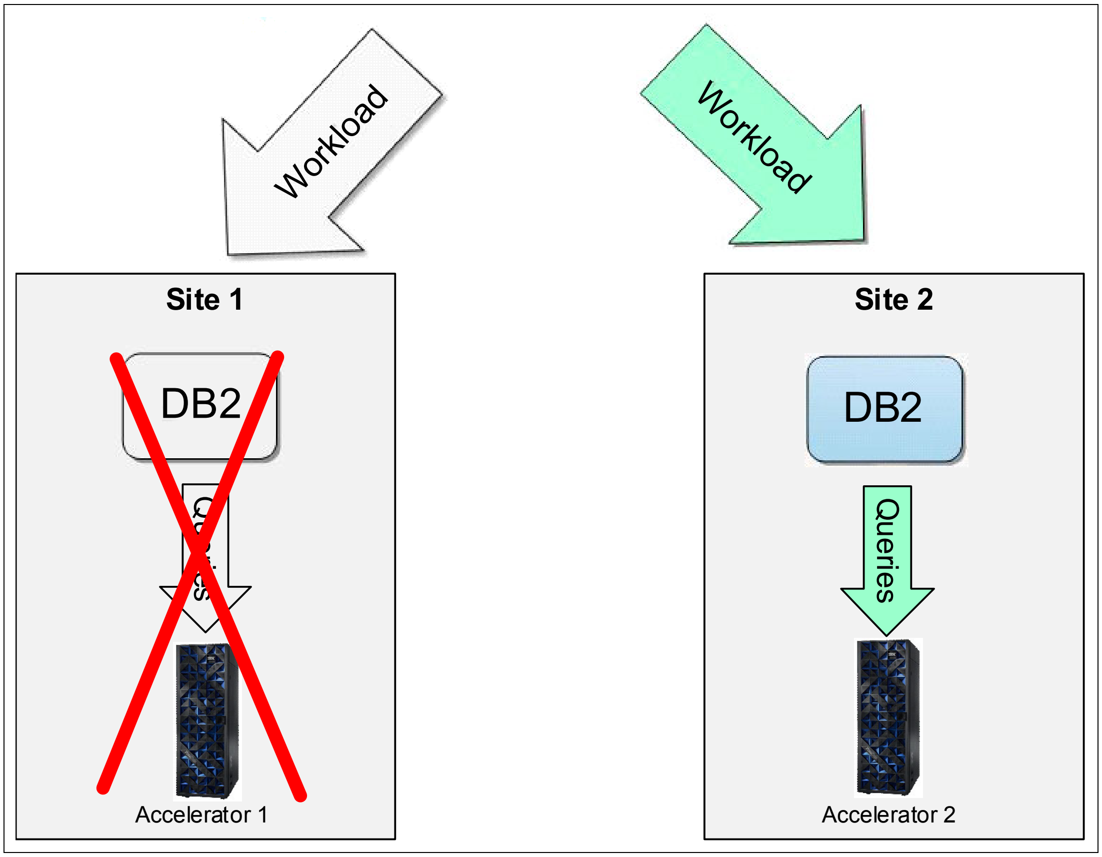

Figure 25 Workload flow with GDPS/XRC and accelerators

Data maintenance using regular load cycles

Consider maintaining data in Accelerator 2, as described in "Data maintenance using regular load cycles" on page 21.

Data maintenance using incremental update

Consider maintaining data in Accelerator 2, as described in "Data maintenance using incremental update" on page 22.

With asynchronous copy, there might be a situation where changes are applied to the accelerator with incremental update, while changes though GDPS-XRC were not yet written to the secondary disk at the disaster recovery site. The recorded LRSN or RBA in the accelerator does not match the records in the DB2 subsystem. In such a situation, the incremental update component recognizes this inconsistency and the corresponding table on the accelerator must be reloaded.

High-performance storage saver

Consider archiving data on both accelerators, as described in "High-performance storage saver and multiple accelerators" on page 12.

GDPS/XRC with limited bandwidth to remote accelerator

If bandwidth is not sufficient to maintain data regularly on the accelerator in Site 2 (see Figure 26 on page 33), query workload is processed only by Accelerator 1 during normal operations.

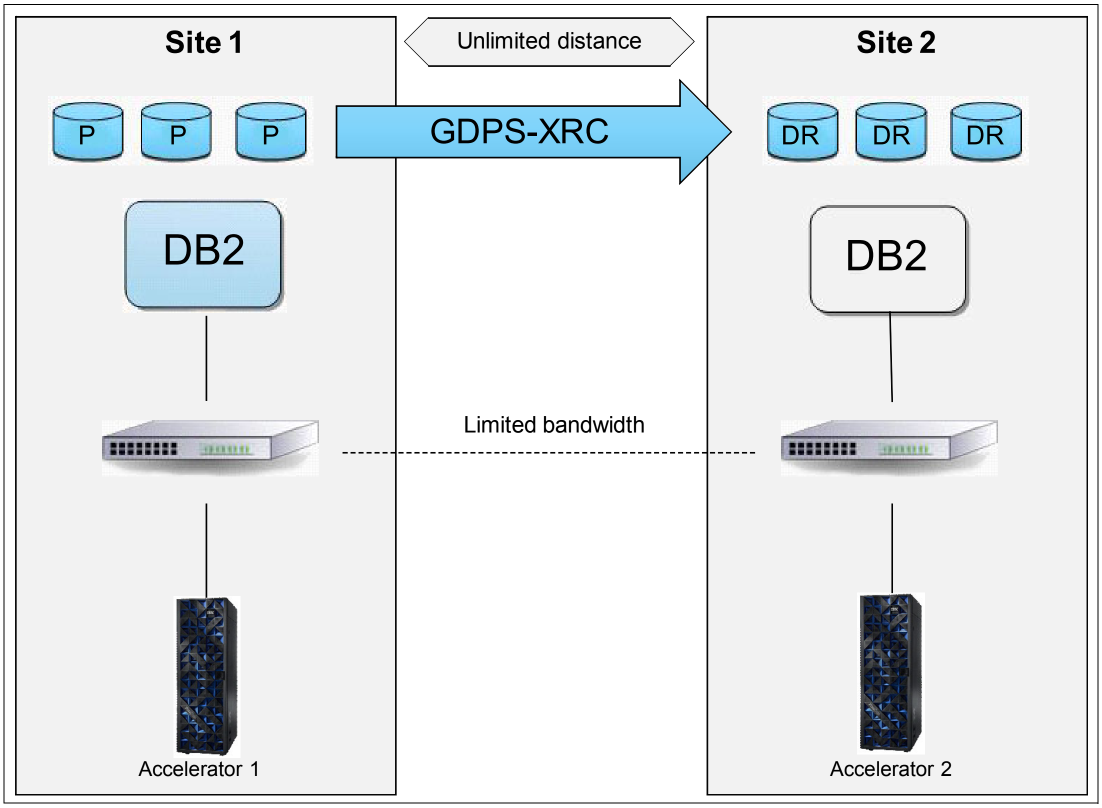

Figure 26 GDPS/XRC with limited bandwidth to accelerator

Even with limited bandwidth, Accelerator 2 should be paired with the DB2 for z/OS subsystem in Site 1. This ensures that if there is a failover, the DB2 for z/OS subsystem in Site 2 can be brought up and is ready to use accelerators in both sides.

If there is no TCP/IP connectivity at all from Site 1 to Accelerator 2, Accelerator 2 must be paired with the DB2 for z/OS subsystem in Site 2 after a failover was started.

If it is not possible to maintain data on Accelerator 2 from the active DB2 for z/OS subsystem in Site 1, Accelerator 2 should be stopped during normal operations. Accelerator 2 must be loaded to satisfy incoming query requests if there is a disaster. This leads to RTO > 0 from an accelerator perspective, as shown in Figure 27 on page 34.

After a failover, Accelerator 1 cannot be used for query workloads from the DB2 for z/OS subsystem in Site 2 because it cannot be maintained. Therefore, Accelerator 1 must be stopped from the DB2 for z/OS subsystem in Site 2 after a failover.

If available bandwidth does not permit maintaining data in Accelerator 2 through incremental update during ongoing operations, Accelerator 2 must undergo an initial load of all required tables if there is a disaster before re-enabling incremental update for those tables.

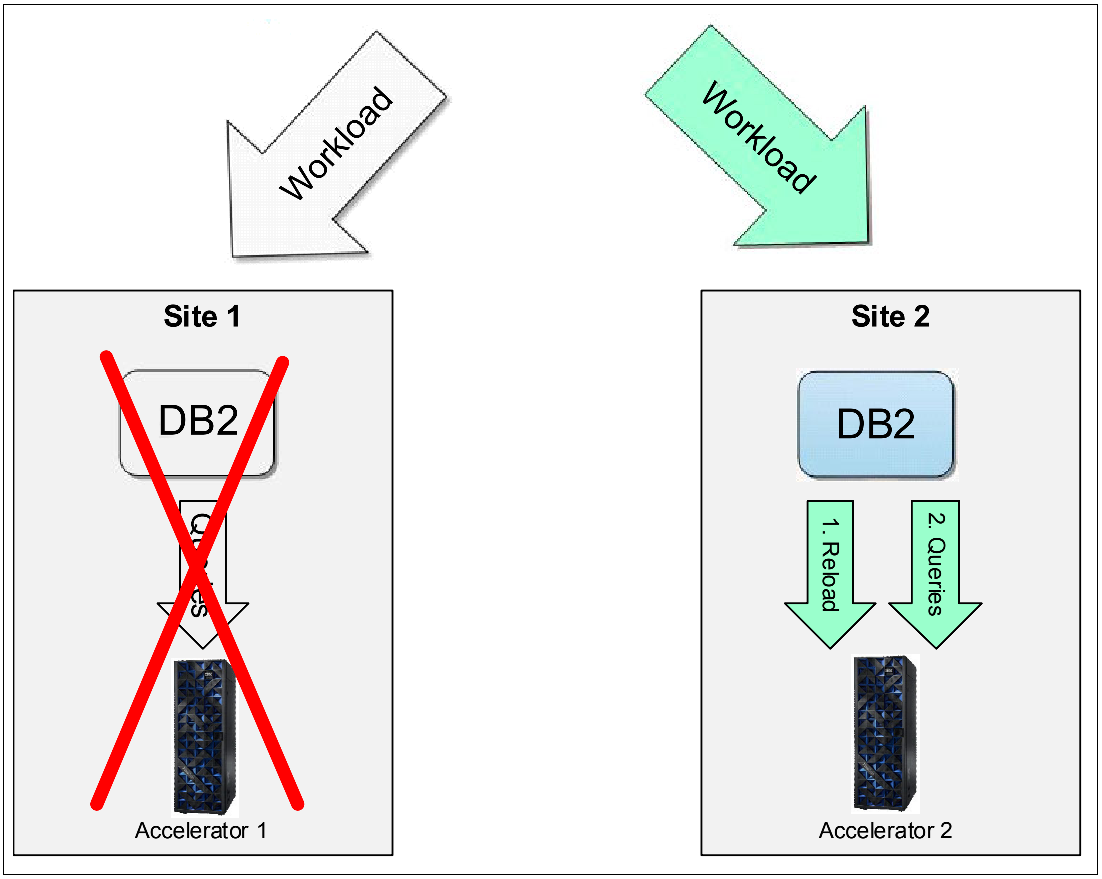

Figure 27 GDPS/XRC with limited bandwidth after disaster

Data that was archived on Accelerator 1 must be manually recovered only into DB2 for z/OS and archived again on Accelerator 2 after a failover.

Image copies that were automatically taken when data was archived on Accelerator 1 were also mirrored in the secondary DASD volume in Site 2. To recover previously archived partitions in DB2 for z/OS, PRO (persistent read only) state must be removed from partitions flagged as archived in DB2 for z/OS. After the PRO state is removed, RECOVER utility restores previously archived data in DB2 for z/OS. After data is restored in DB2 for z/OS, it can be archived again in Accelerator 2.

Combined PPRC and XRC environment with accelerator

The concepts that are described in this document can be combined to build a suitable configuration for even more complex setups.

Figure 28 on page 35 shows an example in which GDPS/PPRC is used to implement an active-active solution with two systems on a campus (closely located to each other) and another remote site to cover a disaster recovery scenario.

The suggested setup is to have multiple accelerators, at least one per site to address the same outage scenarios as with the DB2 subsystems. Again, all accelerators can be maintained and active to serve query requests.

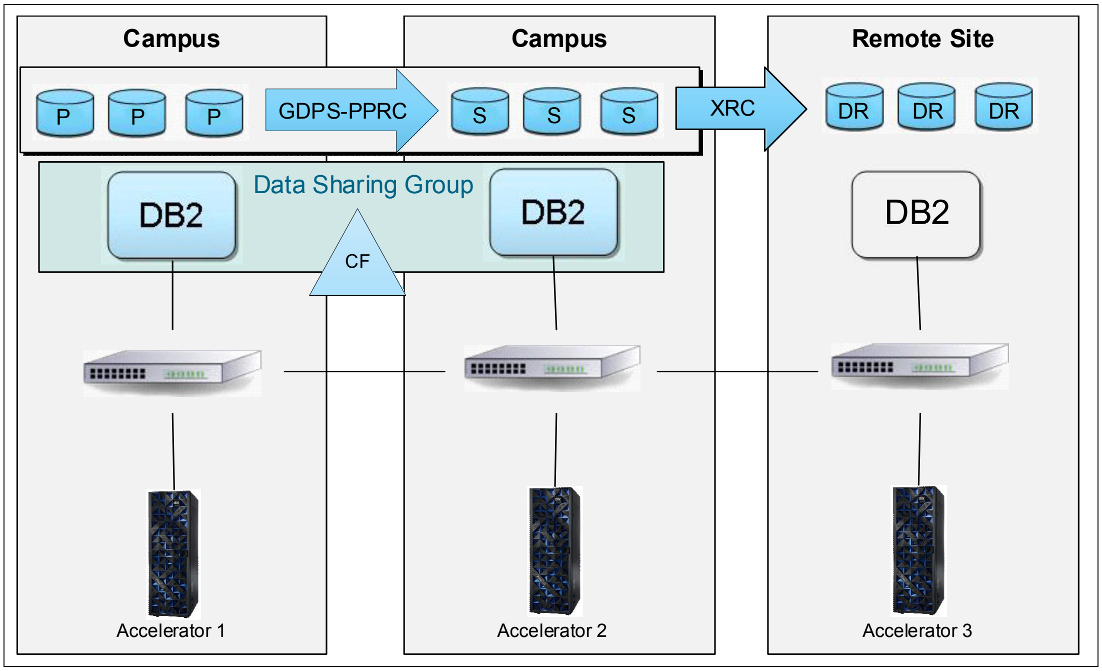

Figure 28   GDPS/PPRC and GDPS/XRC with accelerators

Summary

IBM DB2 Analytics Accelerator can integrate well into existing HA and DR environments and extend the strength of System z servers by supporting complex queries that are accessing large amounts of data efficiently. Table 1 summarizes the suggested options with characteristics.

Table 1   Summary of options

GDPS option DB2 operation	**Accelerator option**	**RPO DB2**	**RTO DB2**	**RPO Accelerator**	**RTO Accelerator**
PPRC active/active	2 active	0	0 (HyperSwap®)	No data loss	0
PPRC active/standby	2 active	No data loss	>0	No data loss	0
XRC active/standby	2 active	A few seconds (potential data loss in case of disaster)	1-2 hours	No data loss (reload if out of sync)	0
XRC active/standby	1 active, 1 standby	A few seconds (potential data loss in case of disaster)	1-2 hours	> 0	> 0 (needs reload)

Additional information

For more information, see the following resources:

- *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370:

  http://www.redbooks.ibm.com/abstracts/sg246370.html
- *GDPS Family: An Introduction to Concepts and Capabilities*, SG24-6374:

  http://www.redbooks.ibm.com/abstracts/sg246374.html

Authors

This paper was produced by specialists working at the IBM Research & Development Lab in Boeblingen, Germany.

Patric Becker is a Software Architect at the Data Warehouse on System z Center of Excellence, IBM Germany. The team specializes in IBM DB2 Analytics Accelerator, performs IBM-internal and external education for this product, and supports customer installations and proof-of-concepts. Patric has more than 16 years of experience with DB2 for z/OS. He co-authored five IBM Redbooks®, focusing on DB2 for z/OS, high availability, performance, and Data Warehousing.

Frank Neumann is a Senior Software Engineer and technical team lead at the Data Warehouse on System z Center of Excellence, IBM Germany. As part of the development organization, his responsibilities include customer consultancy, proof-of-concepts, deployment support, and education for DB2 Analytics Accelerator. During his 16 years with IBM, Frank worked as a developer, team leader, and architect for software components and solutions for WebSphere® and Information Management products.

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

Bill White
International Technical Support Organization, Poughkeepsie Center

Ute Baumbach
Michael Dewert
Knut Stolze
IBM Boeblingen Lab, Germany

Gary Crupi
Nin Lei
Dino Tonelli
IBM US

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

http://www.ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:

  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm
- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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-5104-00 was created or updated on May 8, 2014.

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 email 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:

DB2®
GDPS®
HyperSwap®
IBM®
IBM PureData™
InfoSphere®
Parallel Sysplex®
PureData™
Redbooks®
Redbooks (logo) ®
System z®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

Netezza, Netezza Performance Server, NPS, and N logo are trademarks or registered trademarks of IBM International Group B.V., an IBM Company.

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