The Value of Active-Active Sites with Q Replication for IBM DB2 for z/OS

An Innovative IBM Client's Experience

- Learn best practices for continuous availability with active-active sites
- Deploy sites using IBM InfoSphere Q Replication and Lifeline
- Study an IBM client's solution

Serge Bourbonnais
Jiong Fan
Michael Fitzpatrick
Wei He
Xiang Wei Zhou

ibm.com/redbooks

Redpaper

International Technical Support Organization

The Value of Active-Active Sites with Q Replication for DB2 for z/OS An Innovative Customer Experience

January 2015

REDP-5140-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (January 2015)

This edition applies to Version ???, Release ???, Modification ??? of ???insert-product-name??? (product number ????-???).

This document was created or updated on January 21, 2015.

Contents

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.

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:

CICS®
DB2®
DRDA®
FICON®
FlashCopy®
GDPS®
Geographically Dispersed Parallel Sysplex™
Global Technology Services®
HyperSwap®
IBM®
IMS™
InfoSphere®
MVS™
NetView®
Parallel Sysplex®
PowerHA®
RACF®
Redbooks®
Redpaper™
Redbooks (logo) ®
System z®
Tivoli®
WebSphere®
z/OS®
z/VM®
zEnterprise®

The following terms are trademarks of other companies:

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Preface

Any business interruption is a potential loss of revenue. Achieving business continuity involves a tradeoff between the cost of an outage or data loss with the investment required for achieving the recovery point objective (RPO) and recovery time objective (RTO).

Continuous system availability requires scalability, as well as failover capability for maintenance, outages, and disasters. It also requires a shift from standby to active-active systems. *Active-active* sites are geographically distant transaction processing centers, each with the infrastructure to run business operations and with data synchronized by using database replication, such as the Q Replication technology that is part of IBM® InfoSphere® Data Replication software.

This IBM Redbooks® publication describes preferred practices and introduces an architecture for continuous availability and disaster recovery that is used by a very large business institution that runs its core business on IBM DB2® for z/OS® databases. This paper explains the technologies and procedures that are required for the implementation of an active-active sites architecture. It also explains an innovative procedure for major IT upgrades that uses Q Replication for DB2 on z/OS, Multi-site Workload Lifeline, and Peer-to-Peer Remote Copy/Extended Distance (PPRC-XD).

This paper is of value to decision makers, such as executive and IT architects, and to database administrators who are responsible for design and implementation of the solution.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Serge Bourbonnais is a Senior Technical Staff Member and the Lead Architect for InfoSphere Replication Server development at the IBM Silicon Valley laboratory in San Jose, California, in the US. He was the development lead for the origin of Q Replication technology.

Jiong Fan is an Executive IT Specialist from IBM Global Technology Services® organization at IBM in China. He has more than 15 years of experience in China large account support and has worked at IBM for 18 years. His areas of expertise include sysplex, DB2, performance management, data center consolidations, and management.

Michael Fitzpatrick is a Senior Technical Staff Member of the IBM Enterprise Networking Software Group, based in Research Triangle Park, North Carolina, in the US. He is the architect for the Multi-site Workload Lifeline software. Mike has worked in the networking area for 18 years, with a focus on network design and performance.

Wei He is a Consultant IT Specialist for IBM in China. He has 15 years of experience with IBM Global Technology Services and in supporting clients with large IBM System z® installations. His areas of expertise include storage management and storage products, remote copy, and IBM Geographically Dispersed Parallel Sysplex™ (IBM GDPS®).

Xiang Wei Zhou is an Advisory IT Specialist for IBM Global Technology Services at IBM in China. He has seven years of experience in supporting major IBM mainframe clients. For the last three years, he has been working with Q Replication technologies, planning and deploying active-active sites, and doing DB2 performance tuning. He holds a master's degree in Communications Engineering, and his areas of expertise includes Q Replication, DB2 for z/OS, IBM IMS™ databases, and IBM Parallel Sysplex.

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

Pete Siddall, Lead Architect, IBM WebSphere® MQ for z/OS
IBM Hursley laboratory

Cheng Jing, Chief Technology Officer
IBM China

Dell Burner, Information Development Lead, Replication Server
Jonathan Wierenga, Software Development Lead, Replication Server
IBM Silicon Valley Development laboratory

Paolo Bruni, IBM Redbooks Project Leader
International Technical Support Organization, Poughkeepsie Center

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 either 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:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form:

  ibm.com/redbooks

- Send your comments in by email:

  redbooks@us.ibm.com

- Mail your comments:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

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

Executive summary and contents

An active-active sites configuration provides disaster recovery across unlimited distance with a recovery objective of a few minutes or even seconds. Using an active system for failover rather than a standby system also provides for better use of hardware resources at the recovery site. With active-active sites, backup arrangements can be as far away from the primary site as necessary to avoid being subject to the same set of risks as the primary location. But a disaster recovery infrastructure is, foremost, an insurance policy against events that we hope will never happen. But active-active sites also offer more immediate, tangible value:

- Offloading workloads to optimize resource use on a production site
- Eliminating business interruptions during major upgrades and maintenance
- Eliminating the impact of unexpected failures

Active-active sites provide scalability by allowing workloads to be distributed across sites to eliminate contention for data. This improves response time by accessing data in the location where business is conducted. Active-active sites provide nearly immediate failover during outages, because there are no shared resources to release, no file system to mount, and data is transaction-consistent at the alternate site. Zero-downtime (online) upgrades and migrations are achieved by switching applications to another site while performing the activity. Risk is minimized because business activities can continue running for an extended period on the alternate site. The databases of active-active sites are synchronized by using asynchronous software replication, which allows for practically unlimited distance between sites and has no performance impact on the applications. The replication infrastructure can also be extended for enterprise data integration in near real-time.

The Q Replication technology that is included with IBM InfoSphere Data Replication for DB2 for z/OS is the cornerstone of an active-active sites solution for workloads that use DB2 for z/OS. It provides synchronization between two (or more) active IBM DB2 databases. Over the years, several IBM clients have deployed active-active sites based on the Q Replication technology, complementing the solution by using proprietary gateways and automation processes. In this paper, we describe a new solution that includes IBM Multi-Site Workload Lifeline (Lifeline) for distributing workload connections to each active site.

The configuration that is described in this paper was used in production for over a year, Then, it was enhanced to include IBM Geographically Dispersed Parallel Sysplex (IBM GDPS) active-active for its ability to fully automate site switching from a single control panel and for controlling and responding to events such as failures or degraded performance.

Note: Deploying the fully automated IBM GDPS active-active continuous availability solution is beyond the scope of this paper. For more information, see the IBM Redbooks publication titled *GDPS Family: An Introduction to Concepts and Capabilities*, SG24-6374 and the GDPS web page:

http://www.ibm.com/systems/z/advantages/gdps/

An active-active architecture requires three layers of technologies:

- Replication
- Workload routing
- Automation

It can be implemented by combining user-developed technologies with Q Replication, Lifeline for routing, and GDPS for automation.

Customers' existing infrastructures, as well as their requirements, will influence the roadmap for deploying an active-active solution. A pragmatic roadmap often starts with setting up the replication infrastructure, validating operational processes and replication performance, and then progressively deploying the routing and automation components of the solution. An initial replication infrastructure allows offloading selected workloads to the alternate site early in a project for rapid return on investment.

Replication latency is critical for meeting the *recovery time objective* (RTO) and the *recovery point objective* (RPO) of an active-active solution. The recovery time is how long it takes for the application workload to be made available again, and the recovery point is how much data might be lost following an unplanned outage. Q Replication with IBM MQ can deliver subsecond replication latency.

But achieving predictable and reliable performance for different types of workloads requires following preferred practices for system configuration. Q Replication is an IBM DB2 transaction replay technology. As such, all best practices for DB2 performance are also applicable for the performance tuning of the Q Apply process. One difference is that row-level locking is often necessary at the target system to reduce contention, at least for the tables against which large batch updates are run. Other considerations include database objects, such as sequences. Those might need to be altered to avoid conflicts between the values generated at each site.

An active-active sites configuration is particularly flexible for upgrades and migrations. Each site can be different from the other in terms of hardware, software, operating system levels, DBMS versions, capacity, and configuration of sites. In an active-query configuration, where one of the active sites is used for standby of the business transactions, the alternate site can have less capacity. On IBM zEnterprise® servers, Capacity BackUp (CBU)[1] can then be used to augment processing capacity before failover.

[1] For details about CBU for IBM zEC12 systems, see the *IBM zEnterprise EC12 Technical Guide*, SG24 8049.

In this paper, we explain how to perform major upgrades that can include invasive database schema changes and hardware and software upgrades combined with configuration changes. In designing the upgrade procedure, you must answer the following questions:

- How will you resynchronize data between down-level and upgraded systems?
- Do you upgrade each site in sequence or clone the first site updated over on the down-level systems?
- How do you switch application workloads between sites to prevent conflicts and minimize down time?

The answer to each question is “It depends.” In this paper, we explain the pros and cons of usual options before presenting a complete procedure that has been used routinely for a very large installation on z/OS.

The procedure detailed in this paper updates once and then copies the upgraded system over the down-level system. For the case cited, overwriting the down-level system is simpler, with fewer steps and fewer risks of errors, than repeating the upgrade. Automation can be used for the copy procedure, which can be done while transactions continue to be submitted at the source system. The method is particularly suitable for the first deployment of a new active site that is to be kept synchronized in near real time with the source site by using Q Replication technology.

We describe an innovative method that uses PPRC Extended Distance (PPRC-XD) disk copy for fast initialization without downtime. We explain how a major system upgrade that involves disruptive application, database, hardware, and middleware changes, as well as some reconfiguration, is performed with minimal business interruption for a very large DB2 for z/OS database (hundreds of TBs) that services billions of changes daily.

Using IBM Multi-site Workload Lifeline, business operations are switched to another active system while the upgrade is performed on the original system. Q Replication is used to resynchronize the sites after the upgrade. The upgraded DB2 is cloned over the down-level system, preserving transaction integrity by combining synchronous PPRC and synchronous disk copy PPRC-XD with Q Replication technologies. This avoids the need to repeat the upgrade procedure. I/O is briefly suspended to establish a consistent synchronization point in the disk copy, from which the target DB2 system is restarted. Q Replication applies changes that take place during the copy process by capturing the DB2 logs to include any transactions that were in process when the disk copy started. The entire copy process is performed without any application impact.

In summary, this IBM Redpaper™ publication gives a broad understanding of a new architecture for continuous availability for DB2 for z/OS and helps you design a solution, and then install, customize, and configure Q Replication and Lifeline.

This paper includes seven chapters:

- Chapter 1, “Continuous availability with an active-active sites architecture” on page 1, establishes the motivation for active-active sites and defines some of the terms (also see the “Glossary” on page 79).
- Chapter 2, “The technologies necessary for active-active sites” on page 11, explains the technologies required, it recommends using Multi-site Workload Lifeline, Q Replication for DB2 for z/OS, and GDPS active-active sites.
- Chapter 3, “Prerequisites and considerations for deploying active-active sites” on page 21, details the requirements for deploying active-active sites in an enterprise and provides best practices recommendations for Q Replication for DB2 for z/OS and Multi-site Workload Lifeline. (GDPS is beyond the scope of this paper.)

- Chapter 4, “How active-active sites can eliminate outages during IT upgrades” on page 31, describes the customer’s architecture and explains the rationale for configuration and operational choices. It explains the constraints and tradeoffs for which this configuration is adequate.
- Chapter 5, “Case study: An IBM client’s architecture for disaster recovery and continuous availablity” on page 41, discusses disruptive upgrades and how to minimize risk and outage by using active-active sites.
- Chapter 6, “A client’s procedure for major upgrades with active-active sites” on page 57, details the upgrade procedure that is routinely used in a large z/OS installation for hardware, software, and reconfiguration changes.
- Chapter 7, “The zero-downtime copy procedure with PPRC and PPRC-XD” on page 63, describes details of the zero-impact DB2 system copy step of the significant upgrade procedure, which uses an IBM Metro Mirror copy with PPRC-XD and Q Replication for DB2 for z/OS.

1

Continuous availability with an active-active sites architecture

In this chapter, we discuss the motivation, objectives, technologies, tradeoffs, and prerequisites for delivering continuous availability and then make specific recommendations. Providing continuous availability requires a shift from active-standby to active-active systems, but it must also use the infrastructures and processes already in place. Software-based replication is combined with existing technologies that include disk copy and IBM Parallel Sysplex technology.

This chapter includes the following topics:

- Keeping business open 24x7, rain or shine
- Establishing the recovery objectives
- Replication technologies

1.1 Keeping business open 24x7, rain or shine

Every modern enterprise strives for protection against any business interruption. “Service Momentarily Unavailable” can lead to loss of revenue and loss of reputation.

Meeting business continuity objectives takes many forms, based on the cost-benefit tradeoff: Cost of outage or data loss vs. cost of continuous availability.

Modern IT infrastructures provide very high availability with downtime measured in hours per year. But as enterprises increasingly go global and operate across all time zones, idle periods are rare. Business hours become 24 hours a day, 7 days a week. The cost of downtime is increasing, while maintenance window periods are decreasing.

The cost of an outage is at its highest during peak business hours. Consider, for instance, the impact of an outage on a stock brokerage institution even for a few minutes if the outage happens at stock market opening time. Or consider a website that cannot handle the load when a promotional event is launched. *High* availability must move toward *full* availability.

1.1.1 The major causes of business interruptions

An internal IBM client survey found that the following causes are most common causes of business interruptions:

- Planned maintenance:
  - System and application upgrades
  - Reconfigurations
  - Migrations
  - New deployments

  Maintenance is generally scheduled during off hours to minimize its impact. The need for major upgrades is more frequent among fast-growing enterprises, such as following acquisitions for business consolidation. Maintenance requires careful planning and a swift execution. If there is a mishap during the maintenance, it might need to be called off and rescheduled or the outage might need to be extended.
- Component failures or performance degradation, which might happen because of these factors:
  - Human error
  - Software defects
  - Disk failure, subsystem failure, power grid outage
  - System overload
  - Faulty configuration

  Such problems are unavoidable and more frequent with increasingly complex applications and systems. Avoiding the impact of such failures require redundancy of components and services with the ability to increase capacity and to bypass inoperative systems.
- Data corruption, which often results from these causes:
  - Human error (such as data deleted by mistake or running the wrong command)
  - Software defects

  These events are more rare but have damaging consequences. Addressing data corruption requires detection and undoing the mistake or going back to a reliable copy.

- Disasters, such as floods, earthquakes, fires, hurricanes and other natural and human-caused disasters

  Disaster recovery requires geographically remote backup sites.

Data replication technologies in combination with high-availability systems, such as IBM Parallel Sysplex, can be used to address each one of these causes.

For verifying data integrity, IBM also provides tools such as the IBM InfoSphere Data Replication **asntdiff**[1] table comparison utility to detect data corruption in a copy. The replication process itself detects many data corruptions by reporting replication exceptions when the data is not as expected on a replication target, for example, a missing row or a row with unexpected data values.

1.1.2 How to provide continuous availability

Achieving continuous availability at the IT infrastructure level requires several safeguards:

- Failover during maintenance

  Isolating changes to eliminate risk of impact to production systems

- Scalability

  Handling sudden increases of workload to avoid resource contention

- Failover during localized outages and failures

  Bypassing degraded systems

- Failover for disaster recovery

  Running business on a site geographically distant from the production site

We observe that *disaster recovery*, which is restoring business operations after the permanent loss of a site, storage, or systems, is only one requirement for continuous availability.

A failover system can be in standby or active mode. As a matter of fact, achieving continuous availability requires a shift from *standby* to *active* failover systems to avoid restart time. Failing over to another active system can often be done in a fraction of a second. For example, an application that gets a connection error can immediately reconnect to another active site. In the IBM Redpaper titled *Always On: Assess, Design, Implement, and Manage Continuous Availability*, REDP-5109, the author describes the characteristics of the active-standby pattern as well as three other always on patterns.

Figure 1-1 on page 4 summarizes how the requirements and technologies have evolved from the need to provide disaster recovery by making disk copies stored in a safe location, to increased availability by system redundancy, to eliminating any outage by replicating transactions for keeping multiple active systems synchronized.

[1] The **asntdiff** command of IBM InfoSphere Data Replication compares the columns in one table to their corresponding columns in another table and generates a list of differences between the two in the form of an IBM DB2 table.

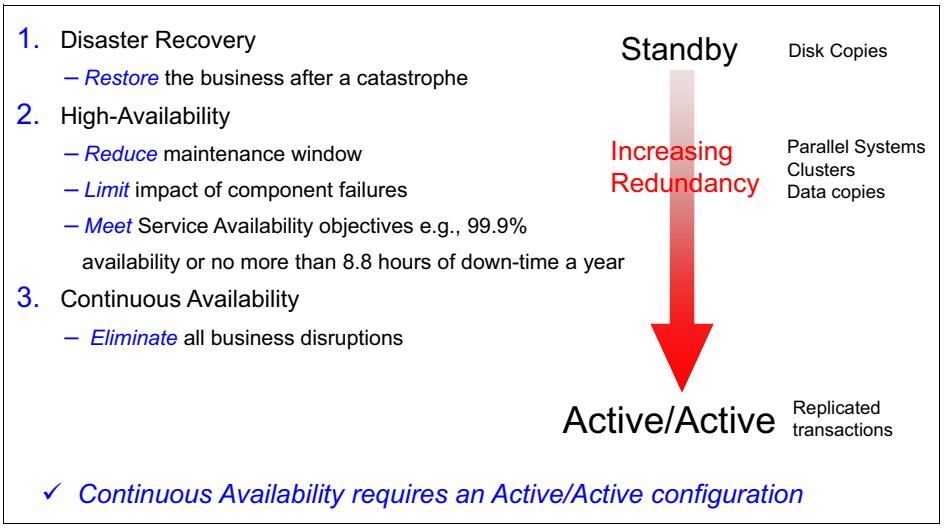

Figure 1-1 Achieving continuous availability

1.2 Establishing the recovery objectives

Continuous availability objectives are measured in terms of the following criteria:

Recovery time objective (RTO) How long does it take for the workload to be made available again?

Recovery point objective (RPO) How much data can be lost[2] after an unplanned outage?

For disaster recovery, this criterion also applies:

Distance objective How far does the recovery site need to be to avoid being subject to the same set of risks as the primary location?

For high-value applications, the RPO must be close to zero. No enterprise can afford not to honor business transactions contracted with their customers. However, we differentiate between business-level and system-level RPOs. A database copy maintained by asynchronous replication will lose some data after a disaster, such as one second's worth of committed transactions. So there must be systems and procedures in place for asynchronous replication to recover the transactions that are not delivered to the standby database management system (DBMS). The RTO must be seconds, or minutes at most. Distance depends on where the business is located and on the risks inherent to the location. It can be from tens to thousands of kilometers.

RPO and RTO objectives can be met at the application, infrastructure, or hardware level, with different compromises and tradeoffs with each technology. Large enterprises use a combination of methods for meeting all of their objectives.

[2] We differentiate RPO of a given technology from RPO at the business level. A technology might lose transactions if there is a disaster, but the business typically deploys complementary technology or processes to recover business transactions. For example, a log of purchase orders might be used to detect and resubmit lost transactions after a major natural disaster causes the destruction of a site. A business might have a phone operator process high-value orders manually while a system is unavailable.

1.3 Replication technologies

Replication technologies are essential for any continuous availability solution. They complement system high-availability features, such as IBM Parallel Sysplex, by providing data redundancy. The redundancy of multi-system images provides transaction processing high-availability, and the copies maintained by replication technologies provide data protection.

Figure 1-2 summarizes the requirements and technologies.

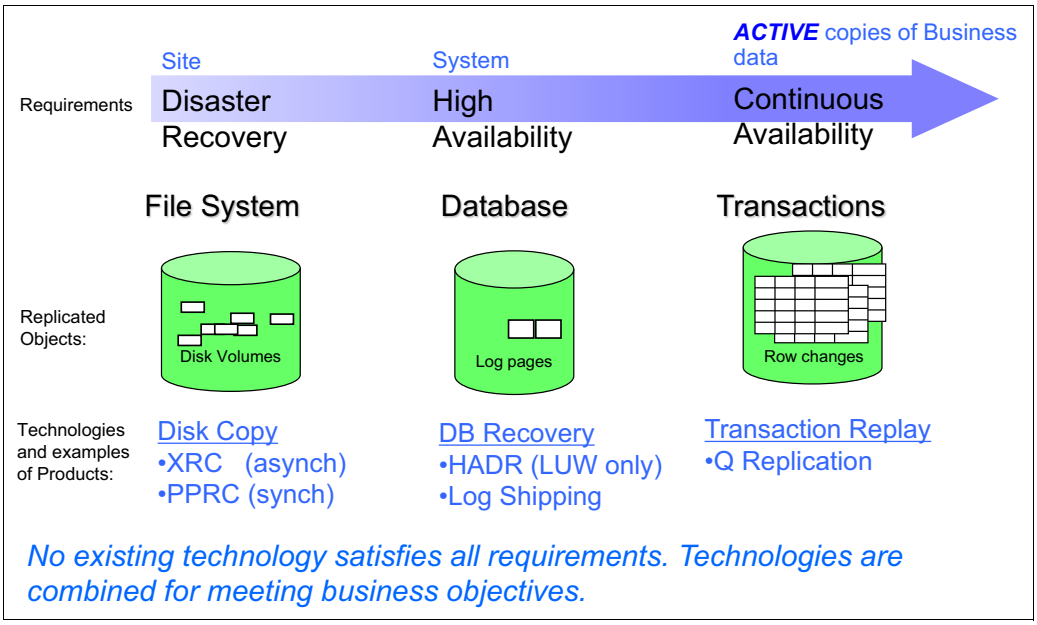

Figure 1-2 Examples of replication technologies

In this chapter, we focus on the data replication requirements for data protection and discuss the tradeoffs offered by commonly used technologies, contrasting disk-level replication with DBMS-level transaction replay replication, such as IBM Q Replication technology.

Note: Application-layer replication technologies are not addressed in this paper. The focus is on infrastructure-level replication solutions.

In selecting technologies, the following requirements need to be evaluated:

- What must be recovered to restore business operations:

  The site? The entire DBMS? Or selected application data?
- Recovery point objective:

  Zero data loss? Seconds or minutes worth of transactions?
- Recovery time objective:

  A few seconds to two hours or more
- Distance required between sites for disaster recovery:

  Tens of kilometers? Hundreds? Thousands?
- Hardware use:

  Active/standby to active/active

- Impact on application response time:

  Direct performance overhead (synchronous technologies) to no impact (asynchronous technologies)

- CPU cost:

  Negligible (hardware-based replication, such as PPRC) to proportional to the workload replicated (transaction replay technology)

A continuous availability solution requires a tradeoff between cost and the level of availability achieved. As a simple example, meeting an RPO of several hours is relatively simple and minimal cost. You just need to make a backup of the data every few hours and ship it to a remote site. But meeting an RPO to 0 requires real-time replay of changes to a remote site, which raises other considerations, such as how far can the remote site be without noticeable degradation of performance for users?

The active-active sites infrastructure model proposes a reliance on asynchronous technology for business-critical applications. It is often combined with existing site recovery procedures to achieve near-zero data loss during major disasters. It can also be complemented with business processes for reconciliation or replay of lost transactions after a disaster where the original site becomes permanently unavailable.

1.3.1 Comparison of methods for DB2 for IBM z/OS offsite disaster recovery

Disaster recovery is only one aspect of a continuous availability solution, but it is perhaps the most essential function for business continuity. Table 1-1 compares the tradeoffs between Geographically Dispersed Parallel Sysplex (IBM GDPS) disk copy technologies, both synchronous and asynchronous, and Q Replication technology for disaster recovery.

Table 1-1 Comparing GDPS disk copy and Q Replication for disaster recovery

Consideration and technology:	**GDPS and XRC z/OS Global Mirror GDPS Global Mirror**	**GDPS and PPRC Metro Mirror**	**Q Replication**
Requirements covered	Disaster recovery (site)	Disaster recovery (site) Continuous availability (site)	Disaster recovery (IBM DB2) Continuous availability (DB2) Information integration (DB2)
Replicated objects	Disk volumes	Disk volumes	Relational tables
Scope of consistency	Related disk I/O updates	Related disk I/O updates	DB2 database transactions
Technology	Asynchronous disk replication with volume group consistency Sysplex (need to time stamp I/O)	Synchronous (Metro Mirror) disk replication with volume group consistency for Parallel Sysplex, IBM z/VM®, and Linux for IBM System z images	Asynchronous Replay database transactions using SQL
Automation	IBM Tivoli® System Automation IBM NetView® Fully automated solution with minimal manual intervention	Tivoli System Automation NetView Fully automated solution with minimal manual intervention	Manual takeover Proprietary automation Automation with GDPS active-active solution

Consideration and technology:	**GDPS and XRC z/OS Global Mirror GDPS Global Mirror**	**GDPS and PPRC Metro Mirror**	**Q Replication**
Platforms	z/OS, Linux for System z, and z/VM GDPS control system on z/OS but any data for IBM Global Mirror	z/OS GDPS control system on z/OS but any data for Metro Mirror	z/OS and DB2 for Linux, UNIX, and Microsoft Windows
Configurations	Active/standby	Active/active (z/OS only) Active/standby	Active/active
Data loss (RPO)	< 3 seconds	0 (sync)	< 3 seconds
Time to recover (RTO)	< 1 hour. (restart workload to release shared resources)	A/S < 1 hour. IPL z/OS. Release shared resources, restart applications. Active-active < few mins	< 1 min (detect and redirect time)
Distance	Unlimited	10s of km (possibly up to 100 km)	Unlimited
Impact on application response time	No impact (asynchronous)	Yes (synchronous), increases with distance: 1ms/100 km	No impact (asynchronous)
CPU overhead	XRC: for System Data Movers on controller LPAR. zIIP enabled	PPRC: Negligible (hardware)	2 - 10% extra at source database Cost of workload replay at target

For more information about GDPS, see *GDPS Family: An Introduction to Concepts and Capabilities*, SG24-6374.

From Table 1-2, you can see that providing enough distance with a minimum RTO requires the Q Replication active-active configuration.

Table 1-2 The major tradeoffs are with RTO, RPO, and distance

	RPO	**RTO**	**Distance**
GDPS/PPRC (active-active)	0(sync)	2 minutes	< 20 kilometers
GDPS/XRC or GDPS/GM (active-standby)	< 3 seconds	1 hour	Unlimited
Q Replication (active-active)	< 3 seconds	< 1minute	Unlimited

1.3.2 Active-active sites with Q Replication

A DB2 active-active sites configuration refers to two or more geographically distant sites, each with active DB2 instances that are completely independent of each other, not sharing any data. A workload that accesses this DB2 data can run at either site. Each site is ready for failover from any direction at the workload level. A workload can be immediately switched between sites or run at each site simultaneously whenever routing can prevent data change conflicts. Sites can be asymmetric with different capacities, different hardware and software levels.

Replication is configured to replicate between all sites, but it is not necessarily running continuously in each direction. Replication can be started in the reverse if and only when needed. The sites can be synchronized by using pairs of Q Replication *unidirectional* configuration setups.

In this paper, we focus on a two sites active-active configuration. More sites are possible with different topology choices, either hub and spoke (that is, replication must pass through Site A to go from B to C) or fully connected (each site replicates to each other site). More than two sites in an active-active configuration is beyond the scope of this paper. Nonetheless, a multiple sites configuration using Q Replication is in frequent use on the distributed platform, particularly for ensuring continuous availability of content data for web hosting according to the ibm.com reference model. For more information, see the IBM Redpaper publication titled *Always On: Assess, Design, Implement, and Manage Continuous Availability*, REDP-5109.

In an active-active configuration, routing determines the role of each site. For example, as Figure 1-3 shows, an OLTP workload can be routed to Site A while a query workload against the data updated by OLTP is routed to Site B until failover. The OLTP workload can be switched to Site B and run on the same site as the QUERY workload.

Routing must prevent conflicts. But replication will detect and resolve any conflicts that are encountered.

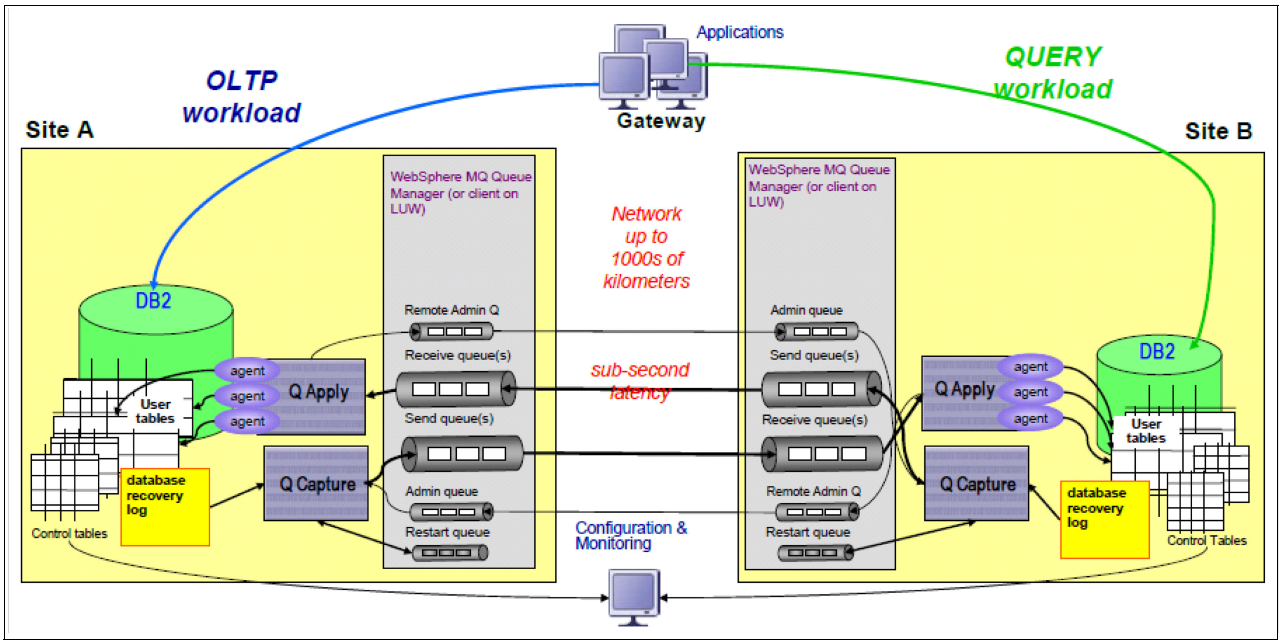

Figure 1-3 Active-active sites with Q Replication

Q Replication provides conflict detection and conflict resolution action. A preferred action, on the premise that conflicts are a rare and a bad thing, is often to stop replication. But Q Replication also provides other methods that include *ignore* or *force* resolution. The latter method, force, is commonly used. A detected conflict is reported to the DB2 IBMQREP_EXCEPTIONS table for analysis and problem determination. Any row in the exception table must be analyzed, because it indicates that a database change might have been lost. Well-designed active-active systems rarely experience replication conflicts. When they are observed, they generally expose either process or human errors.

Conflict resolution is, nonetheless, essential for certain scenarios, such as database initialization from a fuzzy image of a live system that is resynchronized by using Q Replication. Catching up after the copy requires dropping the secondary unique indexes until the two sites are synchronized. Resynchronization is explained in 6.5, “Client upgrade procedure with active-active sites” on page 60.

Bringing back a site after a failover

Replication is configured on both directions between the two sites and uses IBM MQ for staging replicated data whenever needed. After a failover, replication is simply restarted warm to resynchronize the databases. A warm start resumes replication from the point that it was interrupted. With IBM MQ V7.1, up to 64 GB of changed data can be staged at each queue manager. If staging is insufficient, the rest of the changes are retrieved from the source DB2 logs.

If the outage is abnormally long, for example several days, it might be necessary and probably faster to reload selected tables or to rebuild the database.

Note: For details about recent versions, see the IBM MQ product page:

`http://www.ibm.com/software/products/en/ibm-mq`

2

The technologies necessary for active-active sites

In this chapter, we explain what kind of technologies and functions you need for configuration of active-active sites, provide a technical description of them, and discuss considerations and tradeoffs for their use. This chapter covers the following topics:

- Requirements for an active-active sites solution
- Q Replication technology
- IBM Multi-site Workload Lifeline

2.1 Requirements for an active-active sites solution

Asynchronous software-based log capture and transaction replay replication is the basic building block of an active-active sites solution. Routing and automation are necessary to complete the solution. Existing infrastructure can be integrated.

Table 2-1 summarizes the requirements and proposed technologies.

Table 2-1 The requirements for an active-active sites solution

Requirement	Function needed	Technology
Near-real-time log capture and transaction replay Asynchronous database replication	Replication latency of seconds. Performance is the most critical factor, because it affects RPO and RTO objectives.	► IBM InfoSphere Replication Server ► Q Replication technology
Multi-site application gateway and connection routing	Prevent conflicts and provide a function to switch workloads between sites.	► IBM Multi-site Workload Lifeline ► Proprietary gateways
Automation	Control operations, including site switch for maintenance and outages.	► IBM System Automation ► GDPS active-active continuous availability ► Proprietary enterprise automation

2.2 Q Replication technology

Q Replication is a technology that is included with IBM InfoSphere Data Replication. It is a software-based asynchronous replication solution that is designed for high throughput, low latency, and continuous operations.

Q Replication technology uses RDBMS (Relational Database Management System) log capture and transaction replay replication techniques: Committed database changes are captured from the database recovery log, transmitted as compact binary data over IBM MQ messaging software (formerly called IBM WebSphere MQ), and then applied as transactions to the remote databases by using the SQL interface. Transactions are replicated for selected tables, columns, transactions, and operations. For example, it is possible to replicate all transactions for a set of tables except the transactions that come from a specific job or authid (with the IBMQREP_ IGNTRANS table). It is also possible to replicate all operations except any delete operation for a specific table (with the `suppress_deletes` subscription option). Further data transformations are possible and particularly useful for feeding an operation data store (ODS). However, for the purpose of continuous availability, same-to-same replication is generally used.

2.2.1 The Q Replication process

The Q Replication process (Figure 2-1 on page 13) shows how transactions are replicated. The Q Capture program publishes each committed IBM DB2 database transaction that is captured from the log as a *compact* MQ message, meaning that only the data is sent. The message does not contain column names or SQL text. In a data sharing environment, each

Capture program reads the logs for all members of the data-sharing group. The function of merging the logs is provided by the DB2 IFI 306 interface.

Typically, each MQ message contains one DB2 transaction. Q Apply rebuilds SQL statements by using values captured from the log records, with a `WHERE` clause on a unique index to guarantee that only the row that changed at the source is changed at the target. Statements are prepared dynamically once and then kept in a cache for optimal performance. Configuration, control, and monitoring are managed through database tables.

Performance scalability is achieved by executing transactions in parallel, using a pool of apply agents. With Q Apply, transaction consistency is preserved:

- Non-dependent database transactions are applied and committed in parallel, in any order.
- Dependent database transactions are applied serially, in source commit order.

Dependent transactions are any two database transactions that update the same DB2 row or update rows that are related by secondary unique indexes or foreign constraints. A Q Replication consistency group consists of all tables that are replicated on the same receive queue. It corresponds to a receive queue on the apply side.

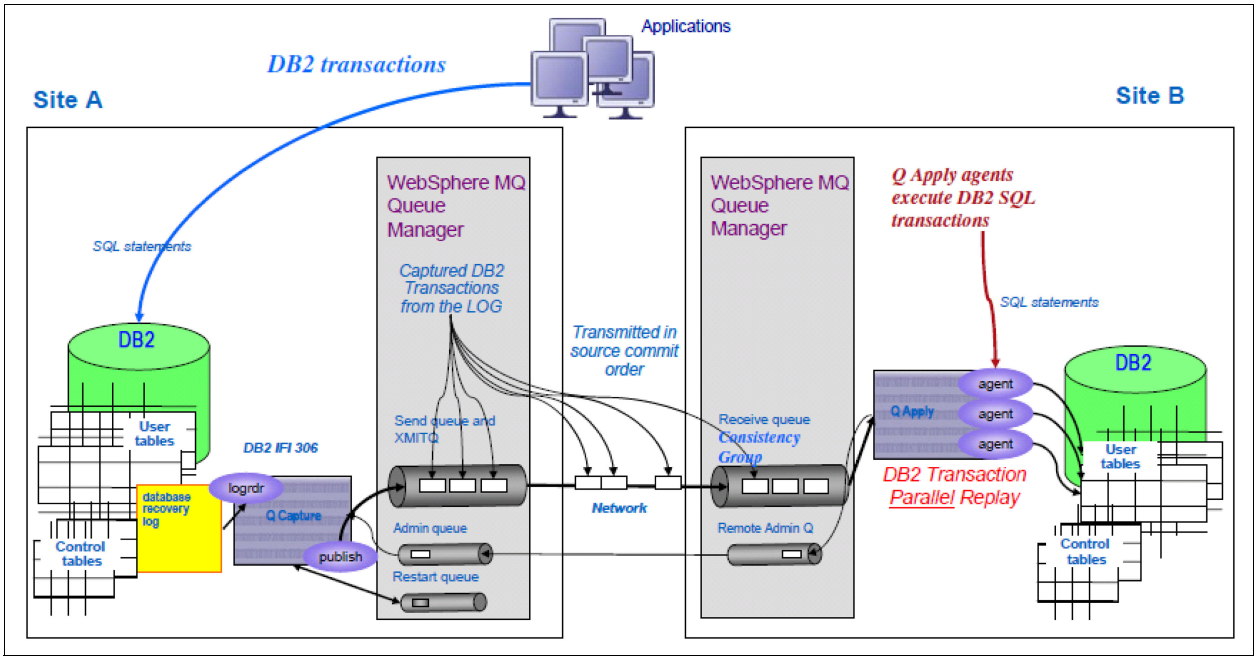

Figure 2-1 The Q Replication process

2.2.2 Log capture and transaction replay technology

Q Apply rebuilds SQL by using values captured from the log records.

The SQL statement that is generated by the Q Apply program is written with a `WHERE` clause to detect conflicts. These checks verify that the row still exists at the target and has the expected values. For an update statement, the apply program `WHERE` clause includes checks on the replication key columns for ensuring that the correct row is updated and, optionally, additional checks on other columns for detecting conflicting changes. The checks always include the replication key columns (`conflict_action='K'`). They can also include changed columns (`conflict_action= 'C'`) or even all columns in the table (`conflict_action= 'A'`).

Example 2-1 shows how an update statement is augmented to include clauses that compare the old values at the target for the replication key (key1, key2) and the changed column (col3).

Example 2-1 WHERE clause for detecting a replication conflict

```
where key1='n' and key2 = 'p' and col3 = 9999
```

If the conflict action is `force`, an update for a row that is missing is changed to an insert.

Figure 2-2 illustrates the end-to-end replication process. Q Apply always includes the replication key in the `WHERE` clause, which is `CUSTID` in this table. Any unique columns can be used as a replication key; it does not have to be the primary key. If an update fails with a duplicate row (sqlcode -803), Q Apply follows the `CONFLICT_ACTION` subscription attribute.

Each replicated DB2 transaction is transported using one or more MQ messages. The transaction is augmented by Q Apply with an additional SQL statment for inserting of the message identifiers into the IBMQREP_DONEMSG table. This enables Q Replication to provide crash recovery without a two-phase commit protocol with MQ. The DONEMSG table contains the message identifiers of the transactions that have already been applied. It is pruned continuously during the replication process.

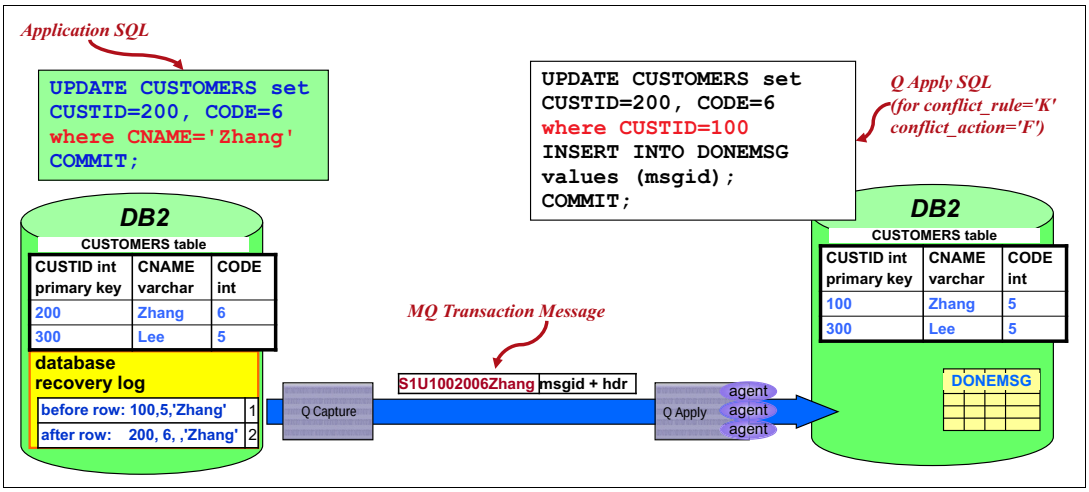

Figure 2-2 Q Replication Log-capture/transaction replay technology and conflict detection

2.2.3 The role of IBM MQ

IBM MQ is normally used by Q Replication with persistent queues under transaction control. It is possible to run Q Replication with a non-persistent message, but it is not recommended for a disaster recovery solution, because messages might be lost if MQ goes down during a failure. A robust disaster recovery solution requires persistence.

MQ provides recovery of messages following an outage, so message delivery is assured. Messages are retrieved by the Q Apply program in source commit order.

Impact of MQ on continuous replication operations

MQ provides transport and persistent changed data staging to Q Replication for continuous operation of the replication process, even when some components are unavailable. For example, an administrator can independently start, stop, or resume capture or apply of a single table to perform maintenance on this table, such as a reorg. Or replication can be started, stopped, or resumed for an entire queue during an application upgrade or migration.

Q Apply can dynamically create a spill queue to stage changes while they cannot be applied. The spill queue mechanism is also used in IBM MQ when the automatic load option is selected (subscription definition `has_loadphase='Y'`). Changes that take place at the source system while the load is being performed are spilled into a queue, which is deleted automatically by the Q Apply program after the table is loaded and the spilled changes have been applied.

MQ role in replication performance

The buffering and batching, with recoverability, is an important function provided by MQ. It maximizes network bandwidth use and prevents falling behind in reading the logs. If and when there is a temporary slowdown of apply actions at the target. Capture will continue sending data, the data is simply buffered in the persistent receive queue.

IBM has been optimizing IBM MQ to provide predictable and stable performance with full recoverability under varied workloads, where a mix of small (OLTP) and large (BATCH jobs) messages are expected at very high volumes. See “Tune IBM MQ” on page 72 for tuning recommendations.

MQ impact on recovery point objective

MQ plays a crucial role for meeting RPO. With Q Replication, the RPO corresponds to the latency for persisting data at the target MQ. Even when data cannot be applied at the target DBMS or when the Apply process is slowed down (for instance, there is sudden CPU shortage caused by running a batch job) or when there is contention for locks at the target, the data replication process continues accumulating data in MQ at the target site. This protects the data from potential disasters at the source. With MQ, change capturing and staging can continue until the target database is restarted.

MQ impact on recovery time objective

A large accumulation of messages in the receive queue affects RTO but not RPO.

In general, clearing a backlog of accumulated messages in a receive queue can be done very quickly. For example, it is not uncommon for Q Apply to clear a full hour of changes in a few minutes. That’s because clearing the backlog can usually be done at peak Q Apply processing speed, which generally far exceeds the average rate while the system was down. The receive queue backlog catchup time must be included in the RTO.

By contrast, without MQ, data must be recaptured from the source DBMS after an outage. If the logs have since being archived, it might be impossible to catch up and a full refresh of the target tables might be required.

2.2.4 Performing the initial load at the alternate site

When creating the initial copy, tables must be loaded at the target without interrupting applications. That is, while the copy of a replicated table is being loaded at the target, changes are still made against that table at the source.

Loading can be done without stopping replication by using Q Replication load with spilling or by stopping replication and using a data copy for the initial load.

Q Replication load with spilling

Target tables can be loaded for the first time, or reloaded, without interrupting the replication process. During the load, changes are spilled by the Q Apply task into an MQ spill queue created by Q Apply at the target, with one temporary spill queue per table being loaded. The loading of the data can be done automatically by Q Apply (internal load) or by the user (external load).

Internal load

Q Apply connects to the source DB2 and does a `LOAD FROM CURSOR` into the target table.

External load

The user unloads the data, transports the load file to the target (by using FTP, for instance), and loads the table. After the load is finished at the target, the user must issue a **`LOADDONE`** modify command to the Apply program.

After the load is done (either internal or external), Q Apply proceeds to clear the backlog of accumulated changes in the MQ spill queue. After it has caught up, it deletes the spill queue[1], and sets the subscription state to `A`. The table is synchronized with the source table within the replication for active latency (by then, it might be one second behind.)

Automatic load involves spilling. It is requested by specifying `HAS_LOADPHASE='I'` on the IBMQREP_SUBS subscription table.

Initialization from a copy

When initializing from a copy, the Q Capture program must be stopped. The data is continuously updated during the copy process. To create a complete copy at the target, all changes must either be in the copy or recaptured, if necessary, from the DB2 log.

Q Capture must be restarted either with a log sequence number (LSN) or with a time stamp that includes all inflight transactions that might or might not be in the fuzzy copy.

A commonly used procedure follows these steps:

1. Stop Q Capture. It will remember where to restart reading the log program.
2. Wait for inflight transactions to be committed to ensure that changes will be in the copy.
3. Copy the data. You can use database backup, unload and load, or disk copy, depending on how much data needs to be copied. In this paper, we describe a procedure to copy an entire DB2 instance by using disk copy. Unload/load for a single table space is also routinely used. Whenever Q Apply spilling is not used, replication must be stopped while loading the data.
4. Set subscription to `HAS_LOADPHASE='N'` (no load with spilling) and `STATE='N'` (for new).
5. Drop secondary unique indexes at the target.
6. Start Q Capture. It will initialize all subscriptions that are in initial N state.
7. All subscriptions will immediately change to A state at source and targets.
8. After replication has caught up (end-to-end latency is within one second), stop replication.
9. Restore indexes.
10. Restart replication.

The target tables are now ready to use.

[1] The IBMQREP_SPILLQS table is an internal table used by the Q Apply program to record the temporary spill queues that it creates to hold messages while target tables are being loaded. The Q Apply program removes spill queues when they are no longer needed.

2.3 IBM Multi-site Workload Lifeline

IBM Multi-site Workload Lifeline enables intelligent load balancing of TCP/IP workloads across two sites at unlimited distances for near continuous availability. It also helps with planned outages by rerouting workloads from one site to another without disruption to users. Lifeline requires IBM z/OS V1.12 or later, with the z/OS Communications Server configured as the TCP/IP stack.

2.3.1 Lifeline functions provided

To aid in providing continuous availability for business-critical workloads, Lifeline provides the following functions:

- Lifeline can direct load balancers to distribute connections for workloads between sites. This can be achieved either by using Lifeline commands to manually switch the workload distribution or through Lifeline configuration to perform these workload switches automatically. These load balancers must support the Server Application State Protocol[2] (SASP). Examples of load balancers that support SASP include the F5 Big IP Switch, Cisco Catalyst 6500 Series Switch Content Switching Module, and Citrix NetScaler appliances.
- Lifeline can detect workload or site failures. These failures can be detected by monitoring the capacity, health, and availability of systems and server application instances within a site for each workload.
- Lifeline enables the switching of workloads connections from one site to another site for planned or unplanned outages. A graceful switch of the workload can be performed in preparation for site maintenance (application or database updates). A forced switch of the workload to the backup site can also be done after a workload or site disaster.
- To avoid a single point of failure, Lifeline maintains workload state information across multiple instances of the workload manager (called the Lifeline Advisor), in case a workload manager fails. This allows a peer workload manager to automatically takeover after the failure of the primary workload manager.

2.3.2 Lifeline commands to influence workload routing

To facilitate the switching of workloads from one site to another, for both planned and unplanned outages, Lifeline provides a list of IBM MVS™ (Multiple Virtual Storage) operator commands:

- To signal Lifeline to direct load balancers to distribute all workload connections for a workload (called WORKLOAD1) to a specific site (called SITE1), an **`ACTIVATE WORKLOAD`** command is provided. The syntax of the command is:

  ```
  MODIFY procname, ACTIVATE,WORKLOAD=WORKLOAD1,SITE=SITE1
  ```

- To signal Lifeline to direct load balancers to stop distributing new connection for this workload, a **`QUIESCE WORKLOAD`** command is provided. The syntax of the command is:

  ```
  MODIFY procname,QUIESCE,WORKLOAD=WORKLOAD1
  ```

  Any active connections for the workload will continue to be routed by the load balancers.

[2] The Server Application State Protocol is an informational RFC that describes a protocol to enable load balancers to receive connection distribution recommendations from Workload Managers. See “Server/Application State Protocol v1” at https://tools.ietf.org/html/rfc4678 for information about this RFC. Request for Comments (RFC) is a publication of the Internet Engineering Task Force (IETF) and the Internet Society.

- To signal Lifeline to ensure that any remaining active connections are reset for this workload after a QUIESCE WORKLOAD command, a **DEACTIVATE WORKLOAD** command is provided. The syntax of the command is:

  ```
  MODIFY procname,DEACTIVATE,WORKLOAD=WORKLOAD1
  ```

2.3.3 Lifeline workload routing

The Multi-site Workload Lifeline software consists of Lifeline Advisors and Lifeline Agents. They work together to monitor a workload's server applications and provide routing recommendations to load balancers.

In a typical configuration, there is a first tier of load balancers that determine which site to route new workload requests to and forwards the request to a second-tier routing infrastructure. The second-tier routing infrastructure determines which server application instance within the site to route the workload connection request to and forwards it to the z/OS system where the server application is running.

Lifeline Agents, which are on each z/OS system across both sites, retrieve information about the health and availability of the server applications and the system where the Lifeline Agent is running. This Information is returned to the Lifeline Advisor, which uses it to create routing recommendations for the workload.

The primary Lifeline Advisor (or the peer Lifeline Advisor in the event of a failure of the primary advisor) provides routing recommendations to the first-tier external load balancers, using the SASP messaging. These load balancers use the recommendations to determine which site to select that will distribute workload connection requests to a second-tier routing infrastructure. The second-tier routing infrastructure routes the workload connections to a server application instance that is running within the site. The second-tier routing infrastructure can be external load balancers supporting SASP, a z/OS sysplex distributor, or customer-provided gateways.

2.3.4 Lifeline configurations

Lifeline supports several different types of workload configurations, based on how a workload is accessed:

- For workloads using z/OS sysplex distributor to load balance connections across application instances, Lifeline can direct first-tier load balancers to route connections to a z/OS sysplex distributor in the primary or alternate site.
- For workloads using customer-written gateways to select application instances within a site, Lifeline can direct first-tier load balancers to route connections to gateways residing in either the primary or alternate site.
- For workloads using customer-written gateways to target a specific application instance, Lifeline can direct first-tier load balancers to route connections to the application instance within either the primary or alternate site.

2.3.5 Lifeline routing infrastructure

The example in Figure 2-3 on page 19 shows a case where a first-tier load balancer distributes workload connections to a second-tier routing infrastructure that is z/OS sysplex distributor nodes.

When the first-tier load balancer handles workload requests based on routing recommendations from the Lifeline Advisor, the load balancer selects a site (Site 1 in this case) and forwards the requests directly to the z/OS sysplex distributor in the site. When the z/OS sysplex distributor node handles the workload requests, it selects a server application within the site, Site 1, and forwards it to the system that is hosting the server application. In this example, sys_1a, sys_1b, sys_1c, or sys_1d.

The site selected by the first-tier load balancer is the site recommended by the Lifeline Advisor, which is the site where the workload is currently active. The server application instance selected by the z/OS sysplex distributor is based on the health and availability of each server application and the system where the server application resides.

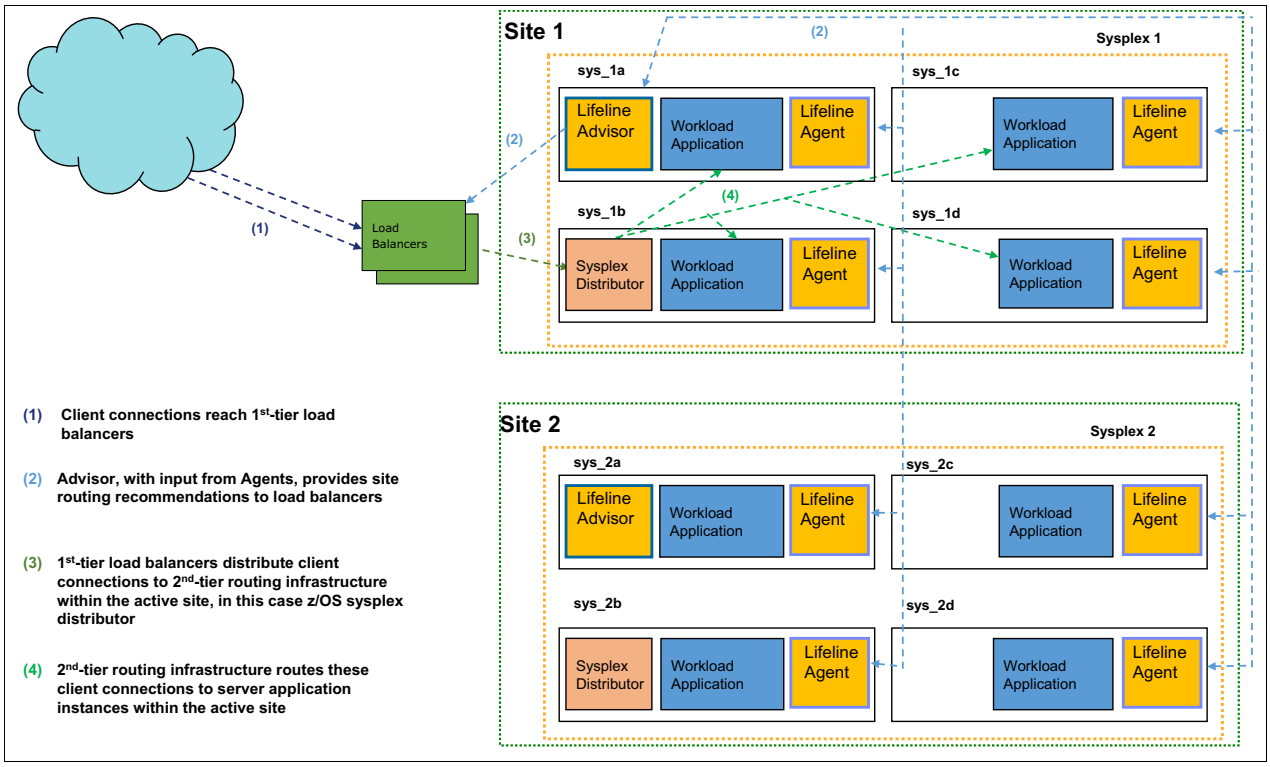

Figure 2-3 Lifeline routing infrastructure

3

Prerequisites and considerations for deploying active-active sites

In this chapter, we discuss the requirements for deploying multi-site applications, considerations for routing, site maintenance, workload management, and database configuration, and then recommend preferred practices. At a high level, the premise is simple: Business operations must be able to run at either site and mechanisms are required to switch these operations between sites.

This covers contains the following topics:

- Solution-level considerations
- Requirements for using IBM Multi-site Workload Lifeline
- Implications of using log capture and transaction replay
- Q Replication configuration considerations for active-active sites

3.1 Solution-level considerations

Each site in an active-active solution should have a compatible configuration to transparently take over workloads during a site switch after a site failure. The following points from system and maintenance perspectives need to be considered when building the active-active sites:

- Business operations must be able to run at either site.

  Applications are installed and IBM DB2 packages are bound at each site.
- Workloads that require continuous availability are identified.
- OLTP query-only workloads can be isolated to run at the failover site.
- Consider positioning the failover site as an analytic hub to handle these functions:
  - Read-only data extraction batch for the data warehouse
  - Data extraction or clone for the data exchange platform
  - Generating operational reports
  - Historical data queries
  - Near-realtime statistical reports
- Maintenance and operation must follow these procedures:
  - They must be defined and carried on at each site. Examples are DB2 runstats and backup of nonreplicated, site-specific files or data loads using non-IBM utilities for which no logging is taking place.
  - Before DB2 V11 for z/OS, `REORG` without the `KEEPDICTIONARY` option must be run on the same member as the capture for the table space. This is to avoid a compression dictionary error if a prior version of the compression dictionary is required to decompress a log record.
- Query-only workloads must tolerate some data staleness due to replication propagation.

  Typically, a latency objective of 5 seconds is reasonable. In practice, most customers average around 1 second with Q Replication. But the application must tolerate some variance in latency. A mechanism must be put in place to switch the application to the production site if and when latency exceeds tolerable limits.

  Possible strategy: Application retries in case of error because data not yet propagated.
- The distribution of a workload's connections must be done so that data conflicts are avoided. This is accomplished by ensuring that transactions that flow on these connections do not update the same data at different sites. Load balancers that are responsible for routing these connections must be aware of which site is active for the workload. This can be accomplished by using Lifeline with load balancers that support Server Application State Protocol (SASP).
- CPU capacity might need to be increased.

  Consider a 10% CPU increase over the workload requirements for the replication process.
- Have a plan for to compensate for data loss.

  Consider using application logs to recover transactions that are not yet replicated by asynchronous replication if and when a disaster happens.
- Implement a backup plan for batch jobs in case they get interrupted.

  You can resubmit the jobs at the failover site after a disaster if the batch jobs can tolerate being restarted from the beginning. In this paper, we provide an alternative approach that is suitable for distances up to 300 kilometers. It uses synchronous disk replication to complement active-active, which allows batches to be restarted from the disk copy of batch files.

There are 3 key elements for batch jobs: DB2 data, IBM Tivoli Workload Scheduler control information, and batch files. It becomes a problem to restart batch jobs at the backup site if they are not synchronized with each other.

Consider optimizing jobs so that steps are reentrant, without dependencies (file, database) between jobs. One solution is to save the batch job checkpoint in a DB2 table, which is replicated with the rest of the DB2 data. You provide a procedure to overwrite Tivoli Workload Scheduler control information at the failover site after a disaster to sync it with the batch job checkpoint stored in the DB2 table.

3.2 Requirements for using IBM Multi-site Workload Lifeline

There is a minimal set of requirements to be able to use Lifeline in an active-active configuration:

- At least one SASP-enabled load balancer, preferably two or more for redundancy, to serve as a first-tier load balancer that is responsible for distributing workload connections to the active site.
- A Lifeline Advisor[1] needs to be configured and active in each site. One Lifeline Advisor serves as the primary Advisor, the other as the peer or secondary Advisor.
- A Lifeline Agent must be configured and active in each system where workload applications are active across both sites.
- Network connectivity must be provided between both sites. This allows Q Replication to send captured transactions to the alternate site. It enables the primary and secondary Lifeline Advisors in each site to communicate with each other, and it allows Lifeline Agents in the backup site to communicate with the primary Lifeline Advisor in the local site.
- Network connectivity must also be configured between all systems within a site where the applications are active. This allows workload connections to be distributed to any application instance within a site.
- Hardware Management Console (HMC) network connectivity must also be enabled between systems across both sites. This enables the primary Lifeline Advisor to query the status of each system's LPAR. That capability is required for Lifeline to detect when a site failure has occurred.

3.2.1 Workload definitions

When providing configuration information for Lifeline, the largest effort involves identifying and separating the business-critical workloads from remaining work running in the site. Lifeline monitors applications and systems and routing for workloads, based on workload definitions in its configuration. Lifeline determines the applications that make up a workload by the IP addresses and ports used by the applications.

3.2.2 Selecting first-tier load balancers

The first-tier load balancers used by Lifeline must support the Server-Application State Protocol (SASP). Many IBM clients already have F5 Big IP switches in their environments but have not enabled SASP in them. It should be a simple matter of configuring the F5 switches to handle SASP messaging.

[1] Up to 128 load balancers are currently supported.

3.2.3 Determining second-tier routing infrastructure

There are two common configurations used by for the second-tier routing infrastructure:

- The default method is to use z/OS sysplex distributor as the second tier to distribute workload connections to applications within the site.
- For customers that have invested in their own gateways to route connections within a site, Lifeline must be configured to recognize this different type of routing infrastructure.

3.3 Implications of using log capture and transaction replay

Replicating by capturing the recovery log implies that only operations that are logged by DB2 for recovery can be detected by Q Replication.

Because the DB2 instances at each site are completely independent of each other, the DBA must ensure that DB2 administrative tasks are consistently performed at each site. For example, tasks such as creating new tables, indexes, or views; binding new packages; load processes using non-IBM utilities; and database maintenance, such as runstats, reorgs, and granting authorities.

These requires two precautions:

- People responsible for applications might need to be informed if and when there is any lost change that results from a replication conflict.
- Failover and fallback procedures might require DBA or operator intervention.

In planning the solution, the technology also imposes some stipulations on database design and workload characteristics.

3.3.1 Some workloads might require special consideration

During the planning phase, understanding how workload characteristics affect replication performance is useful for capacity planning. Review the following list to determine whether any item requires special consideration in your environment:

- Very large volume batch jobs
  - There is a tradeoff between CPU resources at the target and replication latency objectives. A spike in workload requires more CPU capacity.
  - One option to consider: Do not replicate some of the large batch jobs (for instance, a purge job). Run these jobs at both sides instead, using `ignore transaction` by the plan name or job name of Q Replication (IBMQREP_IGNTRAN[2] table). Other transactions are still replicated, but transactions issued by the batch are not. The implementation described in this paper replicates all batch jobs.
  - Some batch jobs with very high degrees of parallelism across several LPARs might cause replication latency to increase, because one Apply program can connect to only a single DB2 data-sharing member at the target. Use Q Replication multiple consistency groups if necessary for more parallelism, and distribute Apply processing across several LPARs.

[2] The IBMQREP_IGNTRAN table can be used to inform the Q Capture program about transactions that you do not want to be captured from the DB2 recovery log.

- Extremely large single transactions
  - Run Q Capture with the **warntxsz**[3] option to detect very large transactions. Establish a policy for applications to commit more frequently.
- *Hot row* updated by many concurrent transactions
  - Example: An application is getting its next application transaction ID by updating a single row in a table that contains the next available ID. Best: Use a sequence instead to generate unique values.
  - Some applications are legitimate hot row applications. For example, a batch application might transfer money from the corporate account to all employees. If such applications exist and DEPENDENCY_DELAY in the Q Apply monitor table exceeds the latency objective, Q Capture TRANS_BATCH_SZ cannot be used. Q Replication will keep up with such workloads if it is given enough resources and without TRANS_BATCH_SZ. See Appendix A, “Appendix” on page 71 for information about performance tuning.
- Multi-row update statement on columns with unique constraints
  - For example, `update t1 set tkey = tkey+1` (this statement updates all the rows in the table). This is not supported by replication without a workaround.
  - A workaround is to add an identity column and make it a replication key.

3.3.2 Some database constructs require special treatment for replication

The source and target DB2 DBMS are independent of each other, and each is administered separately. Any changes other than transactions replicated by Q Replication must be performed at each site.

The replication Apply process is a DB2 application that uses the SQL interface. Therefore, it requires performance tuning and configuration like any other DB2 applications.

The following database constructs require review before deploying a software replication solution:

- Database *sequences* and tables with *identity* columns

  Sequences and identity columns are database objects for which DB2 automatically generates values. Because source and target DB2 instances operate independently, the next value available for an object will almost always be different at the target and the source.
  - Identity columns must be defined as `GENERATED WITH DEFAULT`.
  - Ideally, segregate generated ranges between the two sites. One example is the use of odd/even scheme
- Tables with no unique index

  You must have a unique index at the target and use it as the replication key. If no index exists at the source, one approach is to alter the table to add a hidden identity column to an existing (non-unique) index and use it as the replication key. Adding a hidden column avoids any impact on applications that might select all columns from the table.
- Page-level locking

  This can cause lock contention during large batch job that sequentially updates all of the rows in a table, there are many rows on each DB2 data page, and the batch commits frequently. The Apply process uses several parallel agents that might end up contending for the same data pages when each page contains many rows and the batch sequentially

[3] The Q Capture WARNTXSZ parameter issues a message in the job log to warn you of very large transactions.

updates each row. Therefore, you must use row-level locking at the target, at least for the tables where deadlocks are observed. Change back to page-level if it is needed before a failover. Consider row-level locking for both source and target for simplicity.

- Tables with secondary unique indexes
  - The initial load requires either suspending an application during unload or temporarily dropping the secondary unique indexes until the load is done. See “Dropping secondary unique indexes during restart of Q Capture with old LSN” on page 75.
- Triggers
  - Exclude triggered statements from replication (Q Capture `IGNTRIG` and `IGNCASDEL`).
  - For active-standby workloads, a good practice is to temporarily drop them during fallback for faster resynchronization after a long outage.
- Large objects
  - The presence of large objects (LOBs) reduces replication throughput because LOBs are fetched by Capture from the source database when reading the log record, using the replication key.
  - Define LOBs as inline (if the actual lob data fits on the page), or use VARCHAR whenever possible.

3.3.3 Why row-level locking is generally required at the target

At the source DB2, the order in which the transactions are executed is controlled by the application. A DB2 user can control the order of the SQL operations to avoid lock contention on pages by ensuring that the threads in a batch job are each processing different ranges of rows. For example, thread1 may update rows with keys 1 - 1000000 and thread2 the rows with keys 10000001 - 20000000. Each thread will commit frequently for best performance, for example, every 500 row updates.

At the target DB2, Q Replication replays the transactions with a degree of parallelism up to the number of agents. Parallelism is essential for scalability. The Q Apply program serializes only if there are dependencies. If two DB2 transactions are updating the same DB2 row, they are executed in source commit order by Q Apply. Anything else is executed in parallel. Also, Q Apply knows only about rows and replication keys, not pages. In our example, when Q Apply replays those 500 row transactions with a large number of parallel Q Apply agents, it increases the probability that some of the rows will be on the same DB2 page. Two Q Apply agents will then be competing for the same DB2 page lock, and they may deadlock on the page latch.

It is not uncommon for batch jobs that sequentially update each and every row of a table to cause deadlocks in Q Apply with page-level locking at the target. Reducing the number of apply agents with `MAX_AGENTS_CORRELID`, or `NUM_APPLY_AGENTS` mitigates the problem by reducing the probability of hitting the same page for any table that experiences page lock contention, but reduced parallelism can cause excessive replication latency. Row-level locking is generally required for tables that large sequential update batch jobs are run against regularly.

3.3.4 DB2 impact of converting from page to row locking

For existing DB2 instances, particularly when there are very large volumes of database activity, changing the locking model from page-level to row-level locking raises a concern about potential performance impact. For this reason, some customers keep page-level locking at the source system and change the locking mode only at the target DB2. To ensure predictable performance when there is a failover to the target DB2, they change the locking model at the target before a prolonged site switch.

Changing the lock setting from page to row will not increase the number of locks acquired for random access. But for sequential access, the acquired locks by the thread will increase multiple times. In the worst case, the number of locks will be the number of rows per page.

Consequently, changing the locking mode requires a DB2 configuration adjustments:

- CF lock structure size

  Make sure that the lock structure is large enough. The general recommendation is to double it and monitor it. If your lock size is already oversized, keep its current setting but monitor the CF structure use carefully after the change.

- `NUMLKTS` and `NUMLKUS`

  The `NUMLKTS` subsystem parameter specifies the default maximum number of page, row, or LOB locks that an application can hold simultaneously in a table or table space. If a single application exceeds the maximum number of locks in a single table or table space, lock escalation occurs, and that results in poor system performance.

  The `NUMLKUS` subsystem parameter specifies the maximum number of page, row, or LOB locks that a single application can hold concurrently for all table spaces. The thread will abort if the lock number acquired exceeds this value.

  We recommend increasing `NUMLKUS` by *N* times where:

```
N= average row number per page (a)
   * percentage of sequential access of total access per thread (b)
   * percentage of access for lock setting changed table of total table access
   per   thread (c)
```

  However, the values for b and c are hard to get and may vary between different batch jobs. To make it simple, we generally use 50%.

Q Replication configuration considerations for active-active sites

Active-active sites require replication to be configured to run in each direction.

The most critical requirement for replication between active-active sites is a low replication latency, which is essential for meeting the RPO and RTO objectives. With Q Replication, the RPO corresponds to the latency for persisting messages to the receive queues at the target. Data is safe in the MQ receive queue if there is a disaster.

The end-to-end replication latency, which is the latency before committing the replicated transactions at the target, is critical if you are going to be running queries against near-live data.

Meeting latency objectives raises considerations for replication configuration. For guidance, see “Following Q Replication performance preferred practices” on page 72, which describes these actions:

1. Tune IBM MQ. Use buffer pool read-ahead and dedicated Q Managers.
2. Tune DB2. For Q Apply best performance, follow DB2 best practices for SQL applications.
3. Tune Q Replication. Use multiple send queues.
4. Optimize I/O configuration.
5. Allocate sufficient CPU.
6. Define the appropriate workload manager service class.

Other considerations are related to operations in an active-active sites architecture for continuous availability:

- Use synchronized Q Apply for disaster recovery

  This is to assure consistent point-in-time recovery if and when multiple Q Apply programs or Q Apply receive queues are used to replicate transactions for a site, which is referred to as a *multiple consistency group* (MCG) configuration.[4]

- Replication configuration:
  - Use a unidirectional replication configuration for each direction. This offers more flexibility, particularly during upgrades. The major difference of the unidirectional mode with the bidirectional configuration mode is that subscriptions need to be activated for each direction rather than once from either site.

- Conflict detection and resolution:
  - A fundamental premise of using replication technology in an active-active configuration for business-critical data is that conflicts are exceptions or expected only during controlled operations, such as during initialization using a fuzzy disk copy. Routing prevents conflicts.
  - The conflict rules determine how much of the data is checked to detect a conflict and the types of conflicts that are detected. Suggestions:
    - `conflict_rule = 'K'` for key is recommended. `conflict_rule='C'` can also be adequate, if it is preferred, and it can detect more conflicts.
    - `conflict_action = 'F'` is established in each direction. Its role is during initialization, where Capture resends changes that might also be included in the disk copy.
    - If you initialize using load with spilling, consider using `conflict_action = 'I'`. This enables you to use column suppression for updates during replication for better performance by sending only the changed data. Conflict action of `Force` requires sending all of the data for updates so that the update can be changed to an insert if and when the row is not found at the target.

- Initial target load, without stopping source workload:
  - For very large databases with very high transaction volumes, consider using disk copy or a system backup for initialization. This requires stopping replication during the initial copy. The subscriptions are created with `state = N`, which is defined with `has_loadphase='N'`, and initialized by restarting Q Capture with the LSN option specifying a restart LSN before the copy. This approach requires `conflict_action = 'F'`, but it can be later changed to `'I'` after the subscription is fully activated.
  - For adding tables to an existing replication configuration, reloading individual tables, or initializing smaller databases, use Q Replication load with spilling to MQ. This does not

[4] You use the ASNCLP command-line program to define a multiple consistency group name, assign this name to individual consistency groups, and activate synchronized Apply processing.

require interrupting the replication process for tables that are already replicating. The subscriptions are created with initial state of `I`, which is defined with `has_loadphase='Y'` and initialized by sending a **CAPSTART** signal to Capture.

3.4 Adding hidden identity column to create a unique key

Q Apply requires a unique index on the target table so that the modified row can be uniquely identified, but also for performance. Without a unique index, deleting a single row on a table might require a full table scan, which is very slow. Some tables might not have any. For example, some tables have only a partitioning index. A solution is to add an extra unique column to an existing non-unique index, using a DB2 sequence as hidden column so that it is not returned to queries. This is shown in Example 3-1.

Example 3-1 Adding an extra unique column to the existing index

```
ALTER TABLE SAMPLE.TABLE1
ADD UHIDE DECIMAL(13 , 0) NOT NULL
GENERATED BY DEFAULT AS IDENTITY(START WITH 1,INCREMENT BY 2,
MINVALUE 1,MAXVALUE 9999999999999,CYCLE,CACHE 200) IMPLICITLY HIDDEN ;

DROP  INDEX SAMPLE.INDEX1;
COMMIT;
CREATE UNIQUE INDEX SAMPLE.INDEX1 ON SAMPLE.TABLE1
( OLDKEY ASC ,UHIDE ASC )
USING  STOGROUP SAMPLESG1
         PRIQTY 5000
         SECQTY 20000
         ERASE NO
         FREEPAGE 10
         PCTFREE 5
     BUFFERPOOL BP7
CLOSE YES ;
```

3.4.1 Using DB2 soft fence at the failover site

Site B is used for read-only queries. To protect Site B from unauthorized updates, the installation uses the table space mode RREPL[5], also referred to as *soft fence*. DB2 V11 and DB2 10 (respectively, with APAR PM94354 and PM94353) provide support for RREPL table space access mode. Tables are then read-only for applications while the Q Replication Apply program is allowed to update those tables. Only programs that are APF-authorized and attach via RRS can identify themselves as replication programs and be allowed to update DB2 tablespace access to RREPL mode. For example, all table spaces for a database named trgDB can be set to RREPL mode by using the following command:

```
START DATABASE(trgDB) ACCESS(RREPL)
```

When soft fence is enabled, Q Apply identifies itself to DB2 as a replication application on the RRS attach and then issues the following message:

```
"ASN8055D "Q Apply" : "QALLTYPE" : "BR00000AG002" : Successful call to
DSNRLI('SET_REPLICATION')
```

[5] The DB2 for z/OS RREPL table space option is used to enforce that only a subset of partitions (partitioned table spaces) or tables (non-partitioned or partitioned tables) are allowed to be updated at the spoke nodes.

The soft fence function is used by the GDPS active-active sites product solution, which set the table spaces in RREPL after a workload site switch.

4

How active-active sites can eliminate outages during IT upgrades

In this chapter, we discuss some disruptive system upgrades that often require taking an outage. Some business enterprises limit the number of changes allowed during upgrades to minimize risk, but this can have an impact on their competitiveness. Fast-growing enterprises cannot have a static IT infrastructure. IT infrastructure must keep up with business requirements. An active-active sites configuration is ideal for controlling risk and reducing or eliminating the need for any business interruption.

This chapter covers the following topics:

- Business growth requires IT infrastructure to constantly evolve
- Disruptive upgrades
- Business risks
- Risk mitigation
- Leveraging active-active sites
- Q&A: Designing the upgrade procedure
- Replicating between dissimilar databases during upgrades
- Upgrading the second site by disk copy

4.1 Business growth requires IT infrastructure to constantly evolve

Businesses must constantly evolve their service offerings to stay competitive. The IT infrastructure must follow. Business growth, acquisitions, and enterprise consolidation puts further pressure on IT. No modern business can grow without a dynamic IT infrastructure. Some application upgrades require extensive database schema changes. These changes might require the application to be stopped, which interrupts service to the users.

New or upgraded applications often require hardware upgrade, DBMS upgrade, database maintenance and reconfiguration for extra capacity or tuning. Several upgrades must often be bundled together.

4.2 Disruptive upgrades

These are some of the changes that generally disrupt service availability without an active-active architecture:

Reconfigurations	Tuning parameters, installing new hardware, deploying new sites
Migrations	Moving data to a different DBMS, replacing an application, upgrading middleware
Application cutovers	Deploying new or upgraded applications, which often requires DBMS changes that include modifying existing data structures

4.3 Business risks

Upgrades and migrations introduce business risks:

- Will the application behave the same after upgrading the DBMS to a new version?
- Will the upgraded application consume resources unexpectedly?
- Is the new hardware correctly configured?
- Can you prevent any human mistake during deployment to production and cutover?

The answer to all of these questions, of course, is that you cannot guarantee nothing will go wrong. Tests cannot be exhaustive. Hidden problems might surface only after the upgrade and force an emergency fix or possibly another outage to correct the issue.

4.4 Risk mitigation

With an active-active architecture, risk can be mitigated. Here are answers to questions that are often aised:

- Can I run both versions until the new version is stable?

  Yes. A sound approach for very complex migrations, such as cutting-over redesigned applications, or HW, OS, DBMS and middleware migration.

- For how long? 24 hours? A week? Months?

  Some customers have been running in such dual modes, using Q Replication, for months.

- Can I keep the old version until the new version is satisfactory?

  Recommended. Upgrade the second site only when the new site passes all stability criteria with live production workloads.

4.5 Leveraging active-active sites

Business transactions are redirected to another site while a first site is being upgraded. After the upgrade is completed, transactions that ran on the old version during the upgrade are applied to the upgraded version.

After the upgrade, replication might need to be reconfigured. The data layout at the source and target can be very different, even incompatible. For example, new tables might have been added, other tables dropped, some dropped and re-created with new different columns or data types.

The sequence of actions in the upgrade procedure is outlined in Figure 4-1.

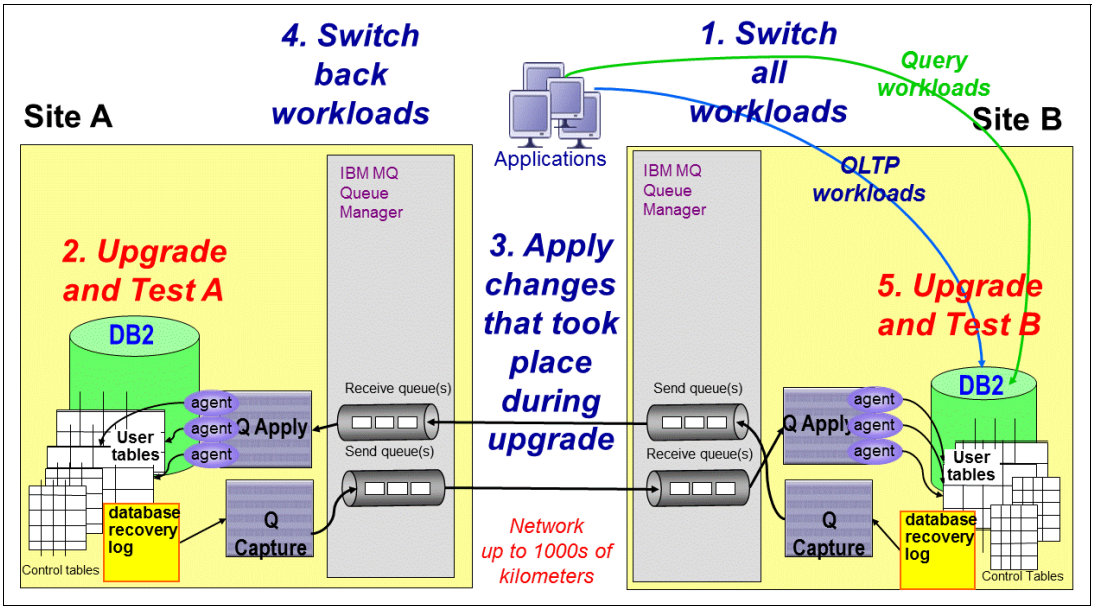

Figure 4-1 Leveraging active-active sites for IT upgrades

At a high level, the upgrade procedure consists of five steps:

1. Switch all workloads to Site B. Stop all replication.
2. Upgrade Site A. Test and verify the upgrade.
3. Resynchronize Site A with changes that took place on the not-yet-upgraded system at Site B during the upgrade.
4. Switch all workloads back to Site A. Stop replication.
5. Upgrade Site B. Verify the upgrade. Restart replication.

4.6 Q&A: Designing the upgrade procedure

In defining the upgrade procedure, several choices must be made. Environment constraints and application requirements influence which procedures will be most adequate. They will determine whether or not any outage is required and for how long.

Site switch of workloads

Q: Does switching workloads from one site to another require an interruption of service?

A: It depends. If transactions can be routed so that conflicts are avoided while connections are being switched between sites, no interruption of service is required. Otherwise, the application must be interrupted while routing is modified. IBM Multi-site Workload Lifeline switches workloads by rerouting connections, so a small interruption is required. Some client's private gateways implement transaction-based routing to ensure that two transactions updating the same data are routed to the same site, even during a switch.

Applying the changes that took place during the upgrade

Q: Can we use replication to resynchronize the databases after the upgrade?

A: It depends on how complex the DB2 schema and application logic changes are. Replication can handle most data changes.

Upgrading the second site

Q: How do we upgrade the second site? Do we use replication to make the DDL changes? Do we run the upgrade procedure again? Or do we clone the upgraded site?

A: It depends on many factors. How large is the database? How high is the transaction volume? How complex are the changes? How much time is allocated?

These steps are explained in more detail in the next sections.

4.6.1 Switching workloads with IBM Multi-site Workload Lifeline

Lifeline provides an operator-driven interface for rerouting connections for a workload from the active site to the failover site. It results in a small outage of the workload during the workload switch.

The following steps can be taken to ensure that no outstanding connections remain on the active site before switching the workload to the alternate site.

1. First, the workload needs to be quiesced by Lifeline so that first-tier load balancers are notified that new connection requests for the workload are to be distributed to either site. Active connections will remain for the workload.
2. Expect a small delay to allow any transactions over these active connections to complete and the connections to be closed.
3. The workload then needs to be deactivated by Lifeline. This ensures that any remaining active connections for the workload are reset and no new transactions can be sent to this site.
4. Another small delay occurs to allow any pending transaction updates to be replicated and applied on the backup site

5. Then, the workload needs to be activated by Lifeline to the backup site. First-tier load balancers are notified that new connection requests for the workload can now be distributed to the new active site.

For a large configuration, rerouting connections using Lifeline can take approximately one minute.

4.6.2 Applying changes that took place during the upgrade

Q: How do we resynchronize after the upgrade?

A: Two techniques can be used to apply transactions from the old version to the new one. Which one is the most adequate depends on the complexity of the changes to the database.

- Whenever the changes are compatible, such as a data type changing from integer to decimal, replication from old to new is the best approach.
- Whenever the changes require complex business logic, such as merging data from multiple tables into a single table, an application-level resynchronization procedure might be the most practical option to apply the business logic that corresponds to the new database layout. For example, that might mean replaying the transactions by using the new interface. This approach requires either capturing the transactions for later replay or developing a migration application that is installed above the old database, so you can then directly transform those transactions to the new format.

  This introduces an extra step before the upgrade: First, install the application on Site B that will capture the transactions against the old database, store them in a format compatible with the new database format, and then use a proprietary method to apply them on the upgraded database. For this procedure, reconciling data demands a short outage.

4.6.3 Upgrading the second site

Q: Can we upgrade the old database without business outage and without repeating the upgrade procedure, which can often be quite involved?

A: In this paper, we describe a method that overwrites the old database by copying over the upgraded site. This can be a good practice for very large installations with a very large number of changes. The caveat is that during the copy process, the old database is no longer available for failover if a disaster occurs at the primary site while the upgrade of this second database is taking place. In the IBM client's architecture described in Chapter 5, failover would then be to the remote standby site that is maintained by disk copy.

The most appropriate method depends on the size of the database, the velocity of transactions, and the complexity of the changes. There are three choices:

- Repeat the upgrade.

  This is suitable when changes are relatively minor and easy to repeat. One example is a small number of tables to create or alter. This might require making modifications to the replication setup, such as adding subscriptions for the new tables.
- Use Q Replication to replay the DDL on the down-level system.

  This is a good approach when the upgrade involves only alter table DDL that adds columns or modify data types, both of which can be automatically replicated using Q Replication. For more information, see 4.7.1, "Impact of database changes on replication configuration" on page 36.

- Copy the upgraded site over the down-level site.

  This is a good approach when the upgrade involves massive changes and you already have a disk copy infrastructure in place.

In 4.7, “Replicating between dissimilar databases during upgrades” on page 36, we review the functions provided by Q Replication technology. Some IBM clients have performed zero-downtime upgrades by using these functions. Chapter 7, “The zero-downtime copy procedure with PPRC and PPRC-XD” on page 63, describes a no-downtime upgrade.

4.7 Replicating between dissimilar databases during upgrades

In an active-active sites configuration, replication goes in each direction. After a first site is upgraded, replication needs to handle differences between databases, first to catch up to apply the changes that took place during the upgrade, and then, optionally, to upgrade the second site. So there are two ways to use replication:

- Replication is used to replicate changes from the down-level site to the new site. There will generally be type mismatches, for example, a data type is changed from integer to character string.
- Optionally, replication can be used to replicate changes from the new site to the down-level site, to perform the upgrade automatically. But this is currently limited to simple changes that include adding columns and altering data types.

Q: How do we handle each type of change? If we are adding a column, do we add the column at Site B or do we let Q Replication add the column? When are we ready to add the column at Site B?

A: Let’s first review the impact of differences between databases on a replication configuration.

4.7.1 Impact of database changes on replication configuration

Q Replication can replicate between very dissimilar databases. Here, we list the impact of each category of changes on the replication configuration:

- Changes that be automatically replicated, if needed

  The following DDL changes can be automatically replicated:

  - ALTER TABLE ADD COLUMN
  - ALTER TABLE ALTER COLUMN SET DATA TYPE
  - Creating or dropping tables (not available on z/OS, only on distributed

  Q Replication does not need to be stopped when the changes are made. If a table is altered to add a column, Q Replication attempts to add the column at the target, if it is not already added, and updates the replication subscription accordingly. Therefore, you do not have to rely on replication to replay the DDL operations. You can stop replication while you make all DDL changes and restart replication after the changes, because replication ignores changes already made at the target.
- Changes that can be handled with replication configuration changes
  - New tables: Create new subscriptions.
  - Removed tables: Drop the subscription. Both adding and removing subscriptions can be done while replication keeps running for other tables.
  - Dropped columns: Change the subscription, and then stop and restart replication.

- Changes that can be handled with Q Apply stored procedures or Q Apply column expressions

  These changes require stopping replication to modify existing subscriptions:

  - Merging or splitting columns
  - Splitting a table into two or more tables
- Database changes that cannot be handled by replication
  - Merging two or more tables into one table

These conditions are other considerations for taking an outage:

- Need to run business validation scripts before switching sites
- Complexity of writing stored procedures versus impact of a short outage
- Reuse existing and proven proprietary process for post-upgrade synchronization

4.7.2 Replicating from old to new

Replicating from old to new is used for applying changes that took place at Site B while Site A was being upgraded. Replication must be stopped while the changes are made. Q Replication tolerates changes already made. For instance, if replication is stopped and a new column is already added, there is no problem. Q Apply will simply verify that it is the expected data type when replicating the DDL operation.

Table 4-1 and Table 4-2 on page 38 summarize the actions needed by type of change and for each direction.

Table 4-1 Actions by change type

Change	Comments	Action	Outage?
Dropped column	None	Q Apply specifies only the column from the old tables in the SQL.	No
Char or Varchar to Numeric	Expression or stored procedure	Using SQL functions INT, FLOAT, or BIGINT for conversion.	No
Extend length of Char/Varchar	Expression or stored procedure	Using SQL functions LPAD or RPAD to add more characters before or after the original data.	No
Shrink length of Char or Varchar	Expression or stored procedure	Using SUBSTR, SUBSTRING, or TRIM to get part of the original data.	No
Other data type change with user logic	Expression or stored procedure	If the conversion or user logic can achieve by DB2 SQL functions, Q Replication can support by adding an Apply expression to the subscription.	No
One table splitting into 2+ tables	Configuration change	Add Q subscription and use expression or stored procedure if needed.	No
New table load data from 1 original table	Expression or stored procedure	Add a Q subscription with Apply expression or stored procedure. New table can rely on Q Replication load phase (with spilling to MQ) for loading.	No
Drop table	Configuration change	Remove the Q subscription.	No
Two or more tables merged to fewer table	Cannot be done with replication	Must use application to convert the data.	Yes, likely

4.7.3 Replicating from new to old

Table 4-2 illustrates what is possible with Q Replication technology. Some customers have performed zero-downtime upgrades by using these techniques. Chapter 7, “The zero-downtime copy procedure with PPRC and PPRC-XD” on page 63 describes a no-downtime upgrade. Replicating can be used for simple schema change, an added column, and altered data types. New tables can rely on Q Replication load phase (with spilling to MQ) for loading.

Q Replication tolerates changes already made. For instance, if replication is stopped and a new column is already added, there is no problem. Q Apply will simply verify it is the expected data type when replicating the DDL operation.

Table 4-2 Q Replication action by table change type

Table change	Q Replication action	Comments
Added column	Supported, no action needed	The column added in new version should have default value
Add column related with previous data	Expression or stored procedure	Adding column mapping with expression used to port/convert data from existing column
Change unique index	Changing the column mapping settings or using all columns as replication key	NA
Alter data type	Supported, no action needed	Changing data type is always automatically replicated. Q Apply tolerates a change already made at the target.

4.7.4 How zero-downtime upgrades are achievable

This paper describes a solution for extremely complex upgrades for which a short outage is a reasonable compromise. The short outage is used to verify the integrity of the data after the upgrade and when switching connections between sites.

It is possible to completely avoid any application outage during upgrades, and it has been successfully achieved by Q Replication customers over the years. Achieving this goal requires multi-site transaction-based routing that prevents conflicts and resynchronizing the databases by using replication.

The tradeoff is between the impact of the maintenance window outage and the cost of developing routing and replication procedures.

Routing

You must have the ability to run online transaction processing (OLTP) simultaneously at both sites and to keep replication running in both directions while workloads are switched between sites. Before the upgrade, transactions might be distributed across both sites. They are then progressively switched to the same site. The gateway must not allow a new business transaction to be routed to the switchover site until any previous transaction for the same user has completed on the other site. That is, the gateway must serialize transactions for the same user across sites. After all transactions are running on a single site, replication can be stopped and the first site upgraded.

Replication

Replication must be reconfigured to replicate between dissimilar databases. Almost all differences between databases can be handled by replication, but some changes require defining expressions or stored procedures. Writing stored procedures might be a good value proposition for migration scenarios where old and new systems are to co-exist for extended periods of time or for application vendors who want to deliver a zero-downtime upgrade procedure for their products.

The procedure goes as follows:

1. All transactions are progressively switched to Site B.
2. Replication is stopped in both directions.
3. Site A is upgraded.
4. The replication configuration is modified, adding subscriptions and Q Apply expressions as needed.
5. Replication is restarted in both directions.
6. All transactions are progressively switched from Site B to Site A.
7. Replication is stopped in both directions.
8. Site B is upgraded.
9. Replication configuration is modified.
10. Replication is restarted.

There is no interruption of service with this upgrade procedure.

4.8 Upgrading the second site by disk copy

When copying the entire DB2 with disk copy, or a system backup, the copy includes the Q Replication control tables, as well as the IBM DB2 objects and catalogs. You need to configure replication at the source database with the correct configuration for the target. Some changes to the DB2 data are also required.

Without an existing target, it is not possible to test replication before the copy is made. After the copy, any error when replication is started, such as a subscription failing to activate properly, might force you to reload the table. Therefore, there is no margin for error. A wrong MQ name, wrong queue map or subscription definition can cause a subscription activation to fail and force a reload of the table.

However, by using the process that we describe, you can avoid this problem by testing subscriptions after the copy, without replicating any data. An error can then be corrected and the subscriptions restarted. Replication is started only when all subscriptions successfully activate.

4.8.1 Validating the subscription without replicating any data

We validate that all subscriptions activate successfully, without replicating any data, by using Q Apply `applyupto`[1] with a time stamp from the past. This activates the subscriptions, but then it stops before any data is applied.

[1] You can use for the `applyupto` parameter to stop the Q Apply program when it finishes processing all source transactions with a time stamp earlier than or equal to the specified time stamp.

4.8.2 Site-specific tables

These tables must be backed up before the copy and then restored. For example, a journal table records transaction volume at Site B. These tables must be backed up and restored after the copy.

4.8.3 Site B DB2 changes after the copy

After the copy, the target DB2 is identical to the source. Some site and DB2 instance-specific information needs to be corrected at Site B:

- Reset sequence objects, identity, and hidden columns.
  - Change the next value to `max+1` so that it becomes ODD if it was EVEN at the source.
  - Using odd at the source and even at the target to identify adds the benefit of providing a way to identify at which site a change was made by the application. Before you copy, back up the target table minimum and maximum values for the identify columns. After you copy, reset the restart value with the backup one.
- Change distributed data facility settings.

  For Q Replication, the process can be simplified by having the control tables for both source and target at each server and using the procedure described above. This is possible by having the target schema names different from the source schema names. Only the addressing information, such as the DB2 location namem then needs to be corrected. For details about these changes, see 7.3, “Copy procedure steps” on page 65.

5

Case study: An IBM client's architecture for disaster recovery and continuous availablity

In this chapter, we describe an IBM client's architecture for a very large, high volume DB2 system that delivers several advantages:

- Disaster recovery across near and long distance
- Continuous availability for outages, maintenance, or upgrades
- Scalability
- Query isolation

We cover the following topics:

- Client background
- Client objectives
- Customer solution architecture
- System configuration
- Active-query routing considerations
- Value of active-active sites

5.1 Client background

This IBM client is large financial institution with IBM Parallel Sysplex, IBM z/OS system, and IBM DB2 for z/OS products.

Workloads are 24x7, with particularly heavy nightly batch jobs, and peaks during specific periods such as start of business day. There is a mix of online and batch transactions throughout the day. Transactions volumes reach billions of DB2 row changes on a typical business day, modifying terabytes of data.

5.2 Client objectives

Before deploying active-active sites, the client had an active-standby architecture for disaster recovery, based on disk copy replication technologies. The wanted to improve their existing IT infrastructure, specifically in these ways:

- Eliminate or reduce outages for maintenance and upgrades
- Provide failover for unplanned outages, with flexibility: all-site or selected workloads
- Have disaster recovery with geographic isolation and without affecting response time

5.3 Customer solution architecture

The architecture was extended to three data centers across two geographically distant cites, both using Q Replication and disk copy for HA and remote disaster recovery. We refer to the data centers as A, B, and C. The original architecture had Site A and C, and it was extended with Site B for the active-active continuous availability solution. This architecture is illustrated in Figure 5-1 on page 43.

- Site A is the production site where high availability is assured with PPRC disk copy and IBM HyperSwap®[1].
- Site B is the site that was added for queries and failover.

  A and B are the active-active sites.

  Site B provides failover for outages, upgrades, and disaster recovery with an RTO of minutes. This IBM client uses Q Replication for DB2 data to active database at Site B and PPRC for batch data in standby mode at Site B (mounted after PPRC-recover) with sufficient geographic distance to protect from localized disasters, such as floods, earthquakes, and power grid outages. There is a graceful switch of business transactions to Site B during outage or upgrades and migrations. Scalability via query distribution to Site B eliminates contention with business transactions running on Site A. Data is accessible from Site B in near-real-time, with less than one second latency, on average.

  In this configuration, batch-related files are replicated by using PPRC synchronous disk replication, which provides a solution for restarting a batch after a disaster. This solution for batch processing is suitable for distances up to approximately 300 km.
- Site C is used for testing and for catastrophic wide-scale disasters.

  Site C provides disaster recovery with RTO of two hours and sufficient geographic distance to protect from any major catastrophe.

[1] See “HyperSwap for IBM PowerHA® SystemMirror” in the IBM Knowledge Center: http://www.ibm.com/support/knowledgecenter/SSPHQG_7.1.0/com.ibm.powerha.pprc/ha_hyperswap_main.htm

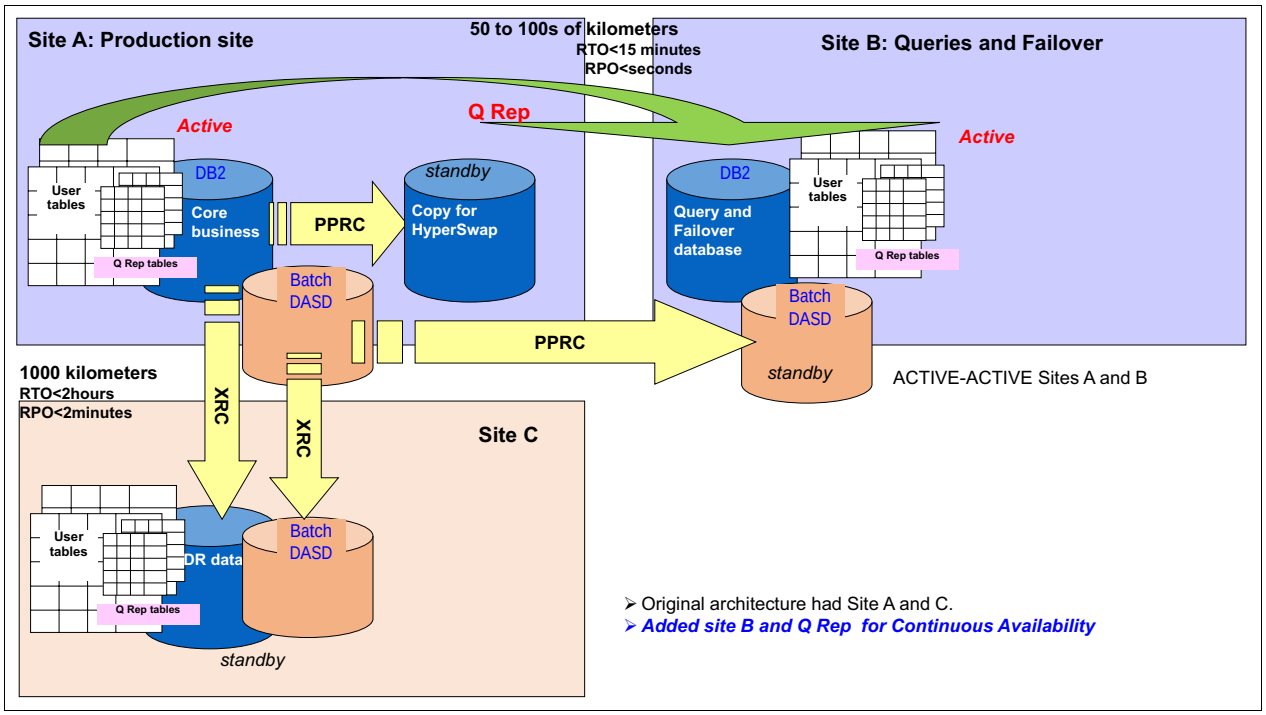

Figure 5-1 Large z/OS installation with continuous availability architecture

5.3.1 Why Lifeline is used for controlling workload connections

IBM Multi-site Workload Lifeline global load-balancing software facilitates the movement of workload connections from one site to another with minimal disruption. This provides the ability to minimize outages for maintenance updates or other planned events.

By providing this workload-switching capability, using Lifeline helps clients with their verification of disaster recovery procedures. Simpler, non-disruptive testing of disaster recovery procedures is accomplished by validating that workloads remain accessible on the recovery site without requiring a site outage on the production site.

5.3.2 Monitoring latency for query workload at Site B

End-to-end replication maximum latency tolerated by the query applications selected for active-active is 5 seconds during the online window and 30 seconds during the nightly batch window. If this threshold is exceeded, queries must be routed to Site A, the production site.

Enforcing query workload policy

The client is using the Replication Alert Monitor utility, **`asnmon`**[2], which is configured to query the Q Apply monitor tables every 10 seconds to monitor Q Replication latency. When the latency exceeds the threshold (5 secs), the ASN5194W warning message is issued. If the latency exceeds 5 seconds in a few consecutive apply monitor intervals, the query workload at Site B switches back to Site A by using the client's proprietary solution based on system automation scripts and gateway routing. See Figure 5-2 on page 44 for the routing configuration.

[2] The asnmon utility provides an alert monitor.

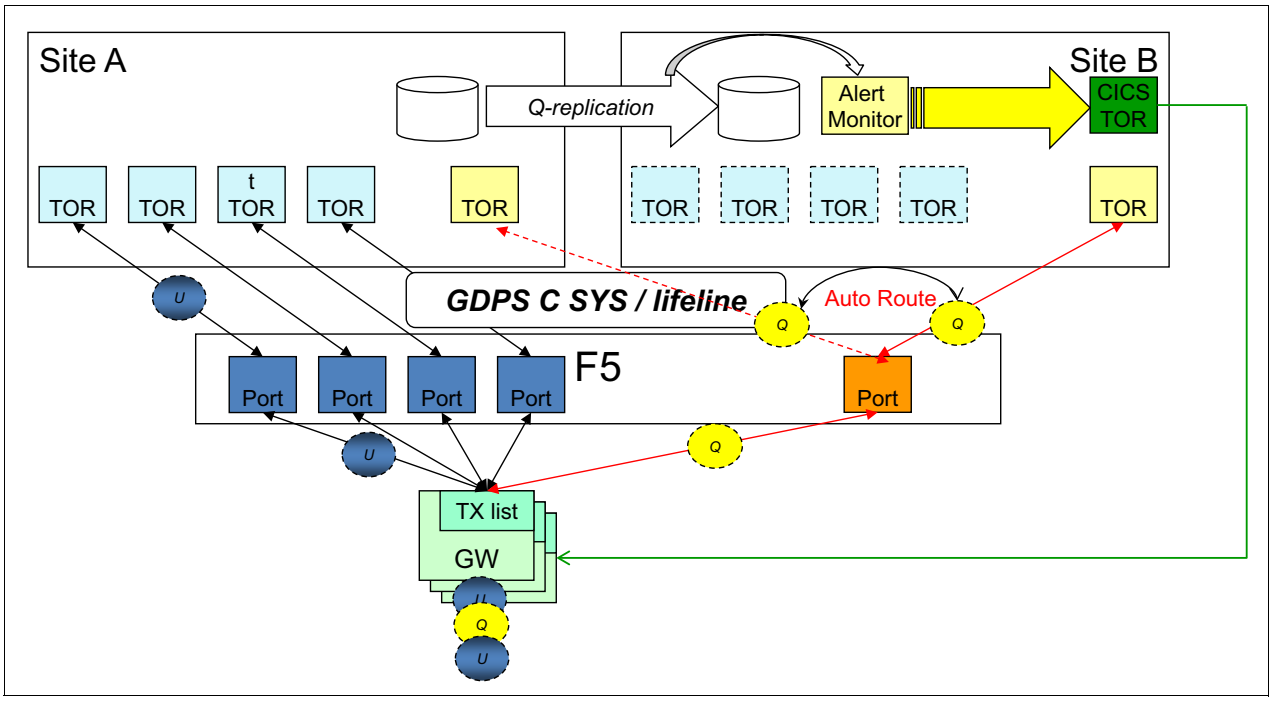

Figure 5-2 Customer's gateway-based transaction routing, query rerouting caused by latency

Although Lifeline can be configured to route workloads between both sites, based on the replication latency of the workload, this client chose to use their own procedures for determining how to route query workloads. So, in the client's environment, Lifeline is configured to always route query workloads to Site B.

System automation (SA) policies monitor asmon messages ASN5194W for Q Replication latency exceeded notices. When the threshold condition is detected, the SA script spawns an IBM CICS® transaction that broadcasts the alert to the client gateway program, which stores this information and changes the routing of query workloads back to Site A, based on the latest control table information.

It takes about one minute to change all gateway control tables. During this minute, some query transactions can still be routed to Site B. Stale data is tolerated during that period.

Query transaction switch back is manually operated by another user command. During site failover, the query workload is first switched back to Site A.

This installation also uses IBM Globally Distributed Parallel Sysplex (IBM GDPS) for automation. From a GDPS active-query workload perspective, the `PROMPT` option is specified. If an active-query workload becomes `ACUTE`, the write-to-operator-with-reply requests (WTORs) will be ignored and no actions are required.

5.3.3 Restarting batch jobs at Site B after an unplanned failover

Restarting batch jobs in a GDPS and active-active environment is a challenge, especially after an unplanned failover. The reason is that you replicate DB2 data and batch-related files by different methods. See Figure 5-3.

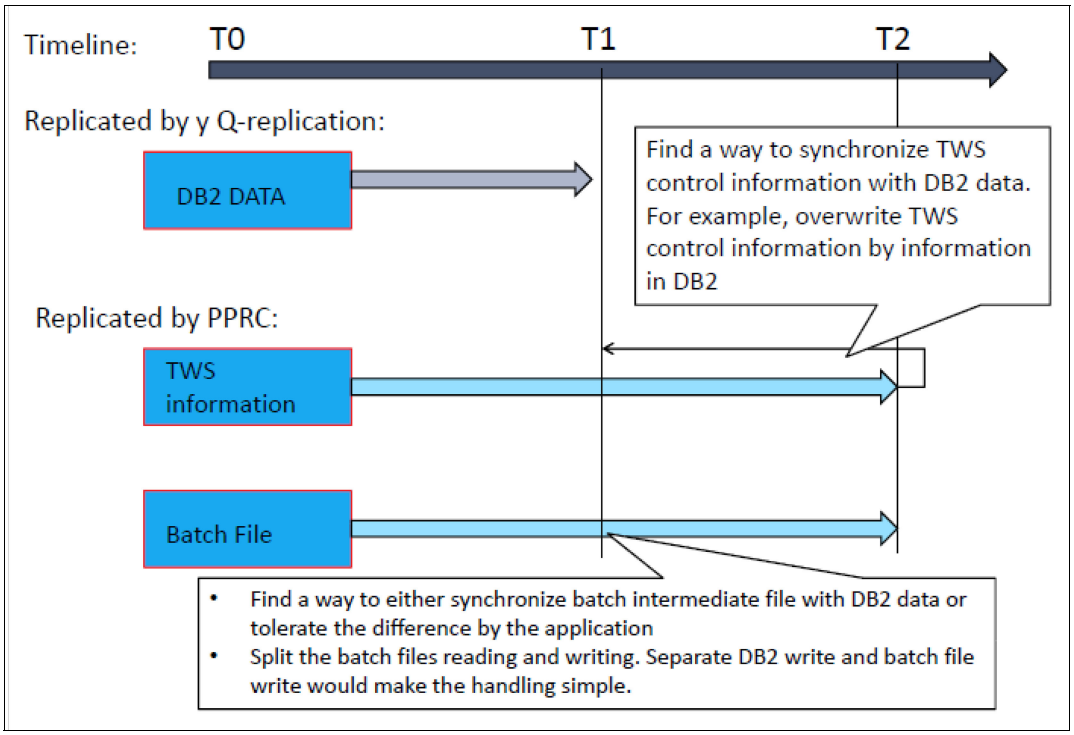

Figure 5-3 Solution for restarting batch after a failover

You might have to face the loss of synchronicity between these two kinds of data. Then, restarting a job could fail or damage the data. The batch-related files include intermediate batch files and control files that are used by job scheduling tools or application, for example IBM Tivoli Workload Scheduler[3] control files.

You might have to develop your own way to handle this situation if you have requirements to resume the batch after an unplanned failover. One possible technology is to use DB2 data as the base. You need to either have a way to synchronize batch-related files or tolerate the difference by the application.

If the distance between active-active sites is less than 300 kilometers, consider using IBM Metro Mirror technology to replicate batch-related files synchronously. This reduces the possibility of mismatch. The batch files will be ahead of the DB2 data that is replicated asynchronously. This solution requires the ability to segregate disk volumes for batch files from those for DB2 data. The Metro Mirror session for disaster recovery is copying DB2 data within Site A and the batch-related files to Site B.

If you use disk copy service to replicate batch-related files, you need be careful, because that file system might flush the update to a direct access storage device (DASD) at the end of the batch job step. Splitting reading and writing to the batch files into two separate steps simplifies the solution. The goal is to further separate the DB2 update and batch file writing.

[3] IBM Tivoli Workload Scheduler automates, monitors, and controls workflow throughout the enterprise IT infrastructure.

Tip: The update of the batch file is not copied to the target at commit, because it is stored in a data set buffer. It is flushed at the end of the job step.

It is better to avoid updating both DB2 data and batch files in the same step. This prevents DB2 data from being ahead of the batch file because of a delay of data set buffer flush.

Most clients' batch processes are already enabled for these functions:

- The batch processing restarts from the stop point if the processing fails.
- The batch can run concurrently with an online workload.

In a GDPS active-active environment, the that needs to be further enhanced to support batch restart after a GDPS active-active workload switch. If you can guarantee that batch files are always ahead of DB2 data at the step level in your batch process, the enhancement can be easy.

5.3.4 Extending the active-active sites configuration

The client architecture is further enhanced to provide near-real-time feed for analytics and data integration on distributed platforms, using Q Replication parallel native Apply technology with Oracle databases. By replicating transactions in near real time, the application can query up-to-date business data from the distributed platform.

Applications running on the distributed platforms have access to mainframe data in near real time. See Figure 5-4.

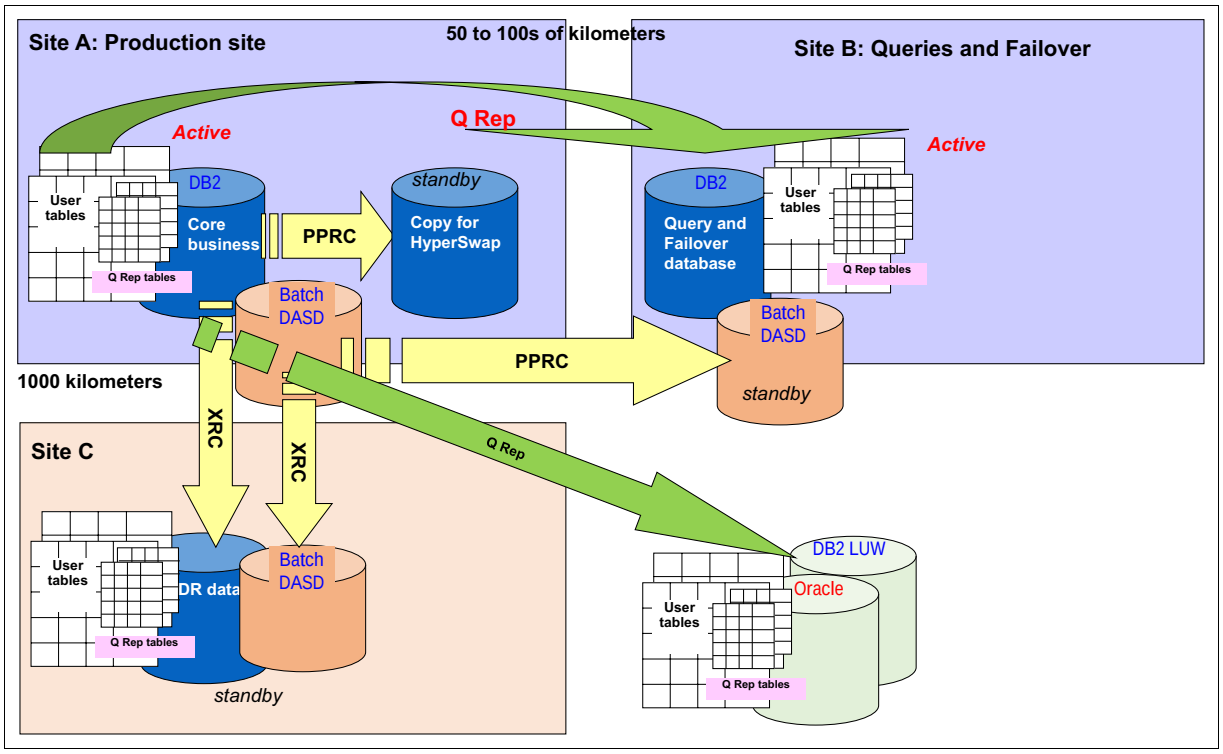

Figure 5-4 Extending the architecture to real-time analytics and enterprise data integration

5.4 System configuration

The client has symmetrical four-way Parallel Sysplex at Site A (source) and Site B (target). This is required for switching the entire business to the other site. More MIPs are configured at the source than at the target for daily active query use. The Capacity Backup (CBU) feature is used to increase Site B before failover. The hardware and software configurations are as follows:

- Hardware
  - IBM zEnterprise EC12 2827-H89 731
  - DS8800 as PPRC source at Site A
  - DS8700 as PPRC target at Site A
  - DS8870 as Q-REP target at Site B
  - DS8700 as XRC secondary at Site C
- Client gateway and GDPS/active-active with Lifeline controls connection routing between Site A and Site B, which are about 50 km away from each other.
- GDPS Global Mirror is used to replicate DB2 and others volumes used for the Q Replication initial copy.
- GDPS and PPRC HyperSwap for disk continuous availability, but batch files, application logs, and IBM Tivoli Workload Scheduler control files are copied using PPRC to 50 km Site B for batch recovery and data loss compensation.
- Six data-sharing groups at each sysplex. The largest data-sharing group has 12 members across 4 LPARs, as illustrated in Figure 5-5 on page 48.
- Page-level locking is preserved at the production Site A, row-level locking at Site B. Page-level locking on Site A can be used because software replication is not used to synchronize Site A after a switch back to Site A. IBM FlashCopy® is used to update DB2 on Site A, instead.

Multiple Q Replication consistency groups (CGs) are used to distribute CPU use across LPARs, using one capture per LPAR. The Apply CGs reproduce parallelism of the source.

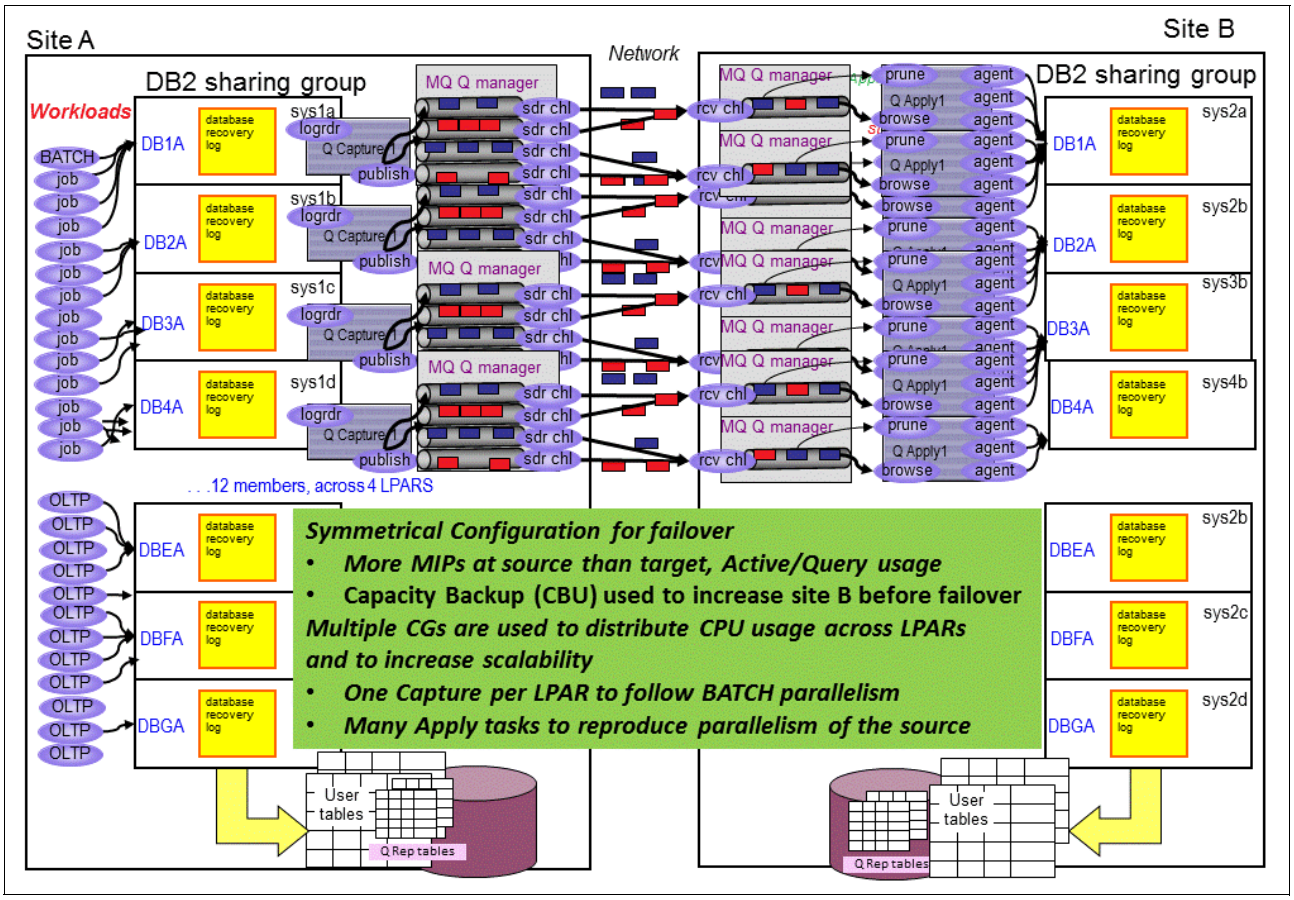

Figure 5-5 Large client active-active sites replication configuration

5.4.1 Customer replication volumes and performance

Table 5-1 illustrates a typical sample period. Performance data is easily available from the Q Replication monitor tables. Clients can maintain continuous and accurate performance statistics that can be recorded every few seconds without any measurable overhead.

Table 5-1 Sample replication volumes

Business period	Total replicated rows (billion)	Total replicated terabytes	Maximum aggregate Q Apply throughput (rows/second)	Maximum single CG Q Apply throughput (rows/second)
Online	1,4	1.2	177,000	108,000
Batch	1,2	0.8	188,000	131,000
Total	2,6	2.0		

Average latency is less than 1 second with occasional spikes. Volumes exceeded 2 TB of changed data in the sample period.

The aggregated results are for 20 CGs in 6 DB2 data sharing groups. The day that the results were captured for this example, the 20 CGs combined did not exceed 188,000 replicated rows per second. A single CG on one of the data sharing groups had peaks of 108,000 replicated rows per second.

Replication performance is directly affected by the size of the data modified and transmitted. For example, updating a DB2 row that is 4 KB in size is more costly than updating a row that is 200 bytes in size. DB2 update operations are also more costly than DB2 insert operations. For this client, average DB2 row size is around 1 KB and batch jobs are predominantly updates.

For more information about replication performance, see *InfoSphere Data Replication for DB2 for z/OS and WebSphere Message Queue for z/OS Performance Lessons,* REDP-4947.

The Total Replicated Terabytes value is obtained from the Q Replication monitor table, IBMQREP_APPLYMON, adding MQBYTES values. It corresponds to the transaction data that Q Capture sends to Q Apply as the payload of IBM MQ messages. The replicated data is a small subset of the DB2 log data. Capture mostly sends only actual data for insert, update, and delete operations, so the DB2 logs contain many log records that are not sent, including transaction management and index management log records.

5.4.2 Customer choices for Q Replication configuration

We highlight several configuration choices that are particularly relevant for active-active configuration operations and performance objectives in Table 5-2 on page 50. These choices and implications are quite common for active-active configurations and are applicable to other users. Review these choices, taking into consideration your objectives and environment constraints.

Table 5-2 Configuration alternative choices

Choice	Rationale	Implications
Dedicated queue manager for each Capture and each Apply program	Performance Q manager version 7.1 has a limit of 1.3 GB buffer pool. A larger buffer pool minimizes data accesses to the pageset.	Additional MQ objects to manage.
Q Capture Multiplexing with parallel send queues	Performance Gives better MQ channel throughput. Often 40% or more. See the IBM Redpaper titled *Always On: Assess, Design, Implement, and Manage Continuous Availability*, REDP-5109.	Additional channels.
Multiple consistency groups MCG to run Capture programs across LPARs	Performance and CPU distribution Batch job volume exceeds capacity of a single consistency group. Customer batch jobs are scheduled to use equivalent amount of CPU across all LPARs. Capture programs are distributed to run across all LPARs.	Transactions are replicated with eventual consistency.
MCG sync Apply function	Disaster recovery Optional. Can be turned off by operator if needed.	A slow CG holds back other CGs.
ASNMON with system automation to monitor alerts	Monitoring Leverage existing SA use Report exceptions to ensure data consistency	Alert about latency will trigger site switch.
Unidirectional configuration for each direction IBMQREP_IGNTRANS contains Apply plan name to prevent recapture	Replication runs only from A to B for queries. It is started in the reverse direction only if there is a failover to Site B. Updates are not allowed at Site B. Not running replication from B to A has advantages: 1. Protect Site A. Any update by mistake that corrupts data at Site B will not be replicated and corrupt Site A. Site B critical table spaces are also protected using RREPL mode, but this mode must be remote to run special batch job. Operator mistakes cannot be ruled out. 2. Saves CPU. 3. Fast restart is achieved by starting the reverse direction following a switch to Site B using the LSN before data switch.	Replication must be started for each direction.
Q Capture and Q Apply attach to specify data sharing member	Control DB2 member to which Capture and Apply attach for better CPU balance across the LPARs If member fails, restart on an-other LPAR after changing the attach name manually. After DB2 10, the subgroup function can be used to define a subset of DB2 members you want QREP attach to. QREP can attach these DB2 members with the subgroup attach name.	HA of replication requires specifying a new member to attach.

Choice	Rationale	Implications
The installation does not use automatic load with spill queues	All subscriptions are defined with `NO LOAD PHASE`. Subscriptions are initialized with disk copy. *Tradeoff:* Stop replication for existing subscriptions while adding new ones. *Reason:* If one table is reloaded, the entire CG is considered unsuitable for queries. The application cannot tolerate a single table out of synchronization within one CG.	Must stop replication when adding subscriptions. If a single table needs to be reloaded, replication is stopped, and restarted WARM. All subscriptions are defined with HAS_LOADPHASE=N
Row level locking at target, page level locking at source	Concern over potential impact of changing locking mode at main production site	Might need to change locking mode, if needed, for site switch.
Soft fence at target (RREPL mode)	Prevent inadvertent updates at Site B	Must reset before failover.
SEQUENCES and IDENTITY odd/even	Need to identify the site origin of each change. It would also prevent conflicts if both sites are updated simultaneously.	Need alter schemas, databases are not identical at source and target.
Use hidden identity column for unique index	Some tables do not have unique index. Some applications may have queries of the type SELECT *	Need to alter tables to add columns.
Q Capture and Q Apply tasks are started with TERM=N	HA of Q Capture and Q Apply started tasks is provided by System Automation policies	LPAR failure requires restarting MQ and DB2 replication tasks.
Striped MQ log data sets in different storage controllers	Performance: Spread the I/O stress	
Identify and move busy MQ log data sets to the storage controllers with lower activity	Performance: Enhance the MQ log I/O performance	
Enable IBM System z High Performance FICON® (zHPF)	Performance: More efficient channel use	

IBM Parallel Sysplex provides high-availability. Failure of a DB2 system does not affect other systems, and transactions can be redirected transparently to the surviving members of a DB2 data sharing group. The replication process must also be resilient to system failures. Q Replication uses the high availability features of DB2, such as group attach, so that the failure of a DB2 member can be transparent.

Because IBM MQ for z/OS does not support a client interface, the Q Capture and Q Apply tasks must run on the same LPAR as the queue manager. An MQ destination is identified by an IP address and port, dynamic virtual IP addresses must be used so that MQ addresses are not tied to a specific LPAR.

There are two failure scenarios to address for Q Replication:

- DB2, MQ subsystems or the Q Replication task fail, but the LPAR is still available
- The LPAR where the Q Replication task runs fails

After a failure, the Q Replication tasks need to be restarted using automation as provided by z/OS System Automation (SA) or MVS Automatic Restart Manager (ARM).

DB2, MQ, or Q Replication failure, LPAR still available

If the DB2 or MQ subsystem fails, this will bring down replication if the replication tasks were started with the option TERM=Y. With TERM=N, Q Replication tasks will keep trying to reconnect to these subsystems until they are available. Use TERM=Y and rely on system automation, because the Q Replication tasks cannot restart themselves. A complete HA solution requires automation.

If the replication tasks fail, but the LPAR where Q Replication is running is still available, restart on the same LPAR

- DB2 member going down. Restart the failed DB2 member. DB2 automatically releases any retained locks as part of the restart.
- MQ subsystem going down. Restart the queue manager and the CHINIT, this will resolve all inconsistent states.
- Restart CAPTURE STARTMODE=WARMSI and APPLY to resume replication.

LPAR where replication runs goes down

If the LPAR where the Q Replication tasks are running fails, the Q Replication tasks, MQ manager and CHINIT and the DB2 system must be restarted on another LPAR:

1. Restart the DB2 member on an available LPAR of the Sysplex with light mode to release any locks that it was holding
2. Restart the queue manager and CHINIT on another LPAR of Sysplex
3. Restart Q Capture STARTMODE=WARNSI and Q Apply on the same LPAR as the queue manager.

To allow System Automation to move the subsystems between LPARS in a sysplex, dynamic IP addresses must be used for the MQ channel. When MQ is restarted to another system, there is no need to manually change the MQ CONNAME IP address and channel initiator ports.

- Use DDVIPA for MQ CHANNEL CONNAME
- If you have other MQ instances on other LPARS for other CG, use different ports definitions for different MQ channel initiators to avoid the same port conflict after a failed MQ has moved to new LPAR.

A sample of DDVIPA definition is shown in Example 5-1.

Example 5-1 Sample DDVIPA definition

```
VIPADYNAMIC
    VIPADEFINE 255.255.255.128 84.16.65.150 ;
    MQ DVIPA FOR SD
VIPADISTRIBUTE
    DISTM ROUNDROBIN 84.16.65.150 PORT 20001
    DestIP 84.51.20.1  84.51.20.2 84.51.20.3 84.51.20.4
ENDVIPADYNAMIC;
```

A sample of MQ definitions (source and target) is shown in Example 5-2.

Example 5-2 Sample MQ definitions

```
SOURCE:
//MQ1ADEFQ JOB CLASS=A,MSGCLASS=X,
// MSGLEVEL=(1,1),NOTIFY=&SYSUID,
// TIME=1440,REGION=0M
//CRTQ EXEC PGM=CSQUTIL,PARM='MQ1A'
//STEPLIB DD DSN=MQ.V7ROM1.SCSQANLE,DISP=SHR
// DD DSN=MQ.V7ROM1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  COMMAND DDNAME(CMDINP)
/*
//CMDINP DD *


DEFINE CHANNEL(QREP.CX.MQ1A.MQ2A)                                +
       CHLTYPE(SDR)                                              +
       TRPTYPE(TCP)                                              +
       DESCR('SENDER CHANNEL TO MQ2A')                           +
       XMITQ(QREP.QX.MQ1A.MQ2A)                                  +
       CONVERT(NO)                                               +
       DISCINT(0)                                                +
       CONNAME('84.16.65.150(20001)')

* RECV CHANNEL
DEFINE CHANNEL(QREP.CX.MQ2A.MQ1A)                                +
       CHLTYPE(RCVR)                                             +
       TRPTYPE(TCP)                                              +
       DESCR('RECEIVER CHANNEL FROM MQ2A')

TARGET:
//MQ2ADEFQ JOB CLASS=B,MSGCLASS=X,
// MSGLEVEL=(1,1),NOTIFY=&SYSUID,
// TIME=1440,REGION=0M
//CRTQ EXEC PGM=CSQUTIL,PARM='MQ2A'
//STEPLIB DD DSN=MQ.V7ROM1.SCSQANLE,DISP=SHR
// DD DSN=MQ.V7ROM1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  COMMAND DDNAME(CMDINP)
/*
//CMDINP DD *

* SEND CHANNEL
                                                                  +

DEFINE CHANNEL(QREP.CX.MQ2A.MQ1A)                                +
       CHLTYPE(SDR)                                              +
       TRPTYPE(TCP)                                              +
       DESCR('SENDER CHANNEL TO MQ1A')                           +
       XMITQ(QREP.QX.MQ2A.MQ1A)                                  +
       CONVERT(NO)                                               +
```

```
       DISCINT(0)                                                +
       CONNAME('84.16.65.151(20002)')

* RECV CHANNEL
DEFINE CHANNEL(QREP.CX.MQ1A.MQ2A)                                +
       CHLTYPE(RCVR)                                             +
       TRPTYPE(TCP)                                              +
       DESCR('RECEIVER CHANNEL FROM MQ1A')
```

Use the DB2 group attach name rather than specific member name for CAPTURE_SERVER and APPLY_SERVER in the capture and apply started tasks as shown in Example 5-3. This avoids having to modify the DB2 member name when restarting capture/apply on another LPAR in a Sysplex.

Example 5-3 Group attach names for CAPTURE_SERVER and APPLY_SERVER

```
QCAP EXEC PGM=ASNQCAP,REGION=0M,
PARM='ENVAR("_CEE_ENVFILE=DD:MYENV")/CAPTURE_SERVER=DBOA    //group name
CAPTURE_SCHEMA=QASN STARTMODE=WARMSI'

QAPP EXEC PGM=ASNQAPP,REGION=0M,
PARM='ENVAR("_CEE_ENVFILE=DD:MYENV")/APPLY_SERVER=DBOA      //group name
APPLY_SCHEMA=QASN'
```

5.5 Active-query routing considerations

Lifeline supports routing of connections for workloads configured as active-query workloads to both sites. The preferred configuration is to use dynamic routing, to allow Lifeline to recommend connections be distributed to the site that is best able to handle the additional work. The decision is based on health and availability of the applications and systems, as well as the average database replication latency between the sites.

This client has chosen to configure Lifeline so that all active-query workload connections get directed to the failover site. This removes any lock contention in DB2 with the production workload running in the active site.

The client then monitors the replication latency for the workload on its own, and will direct Lifeline to route all active-query workload connections to the active site, if the replication latency exceeds their thresholds

5.5.1 Controlling routing of connections between sites

Lifeline requires both a first-tier of load balancers as well as a second-tier routing infrastructure. The client wanted to use their existing gateways for routing transactions to their server applications within a site. Their gateways were modified to target load balancer application groups that were configured in the first-tier load balancers, rather than the server applications within a site.

Each load balancer application group that is configured in their first-tier load balancers consists of two members. These members map to an instance of their application that is running in each site. Client connection requests that are received by these gateways are now routed to the first-tier load balancers, targeting a specific load balancer application group. The

first-tier load balancers then distribute the connections to the server application in the load balancer application group that matches the active site.

An IBM z/OS sysplex distributor is configured in each site so that Lifeline can monitor the health and availability of the server applications, but these distributors are not used to distribute connections to these server applications. The server application selection within a site is done by their gateways.

5.5.2 Customer configuration with Lifeline

Figure 5-6 shows the Multi-site Workload Lifeline configuration that the client uses. The client connections arrive at the client gateways. Based on the type of client request, a specific load balancer application group configured in the first-tier load balancers is targeted. When a connection request is received by the first-tier load balancer, the possible server applications to distribute to are determined by the load balancer application group that is selected by the gateways. After the set of server applications is identified by the load balancer, the connection is distributed to the server application in the active site.

This configuration allows the client to preserve their investment in their gateways, while still allowing Lifeline to control which site workload connections are to be directed. This eliminates possible data conflicts by ensuring that all updates to specific DB2 data are performed on only one site.

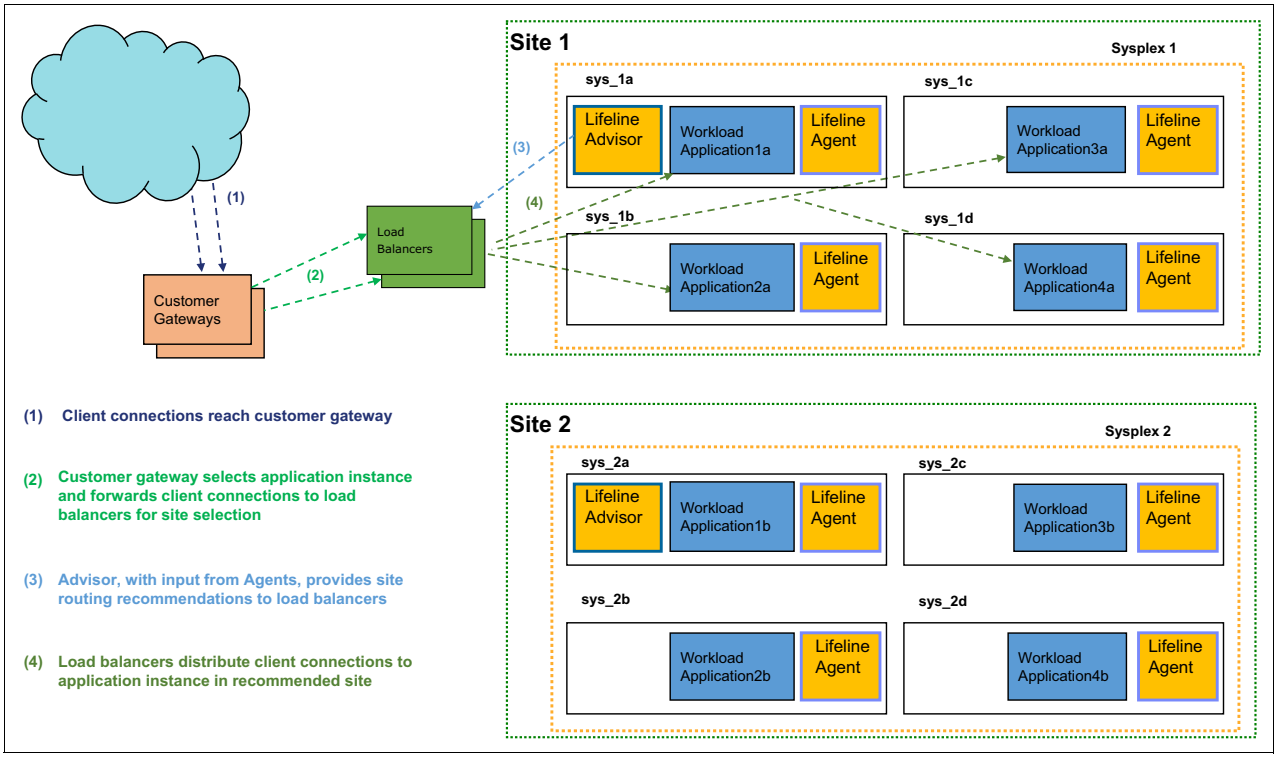

Figure 5-6 Customer active-active sites Lifeline configuration

5.6 Value of active-active sites

Active-active sites synchronized with Q Replication can provide continuous availability:

- They achieve zero-downtime upgrades with graceful switch for planned maintenance.
- Distribution of the client's query-only workloads to the failover site. This helps reduce resource contention on the DB2 data within their production site. This distribution is done only if the end-to-end replication latency remains less than a few seconds.
- Outage protection provides emergency switchover during failures or performance degradation.
- Disaster recovery is possible at unlimited distance, with RPO < 1 second and RTO of a few minutes. The tradeoff is RPO. If using Site B for disaster recovery, data loss must be compensated through business processes.

6

A client's procedure for major upgrades with active-active sites

In this chapter, we explain the upgrade procedure used by an IBM client that has deployed the architecture described in Chapter 5, "Case study: An IBM client's architecture for disaster recovery and continuous availablity" on page 41.

This chapter covers the following topics:

- Client active-active environment
- Client's previous experience with major upgrades
- Improving the upgrade procedure with active-active sites
- Client choices for the upgrade with active-active sites
- Client upgrade procedure with active-active sites
- Client upgrade procedure with active-active sites

6.1 Client active-active environment

Core business applications use IBM DB2 for z/OS. As described in Chapter 5, “Case study: An IBM client’s architecture for disaster recovery and continuous availablity” on page 41, two sites, A and B, are configured as active-active sites.

- Site A is production site that runs the core business, both OLTP and batch.
- Site B is used for running read-only queries. It is a standby for OLTP and batch workloads.

The OLTP workload is active-standby, normally running at Site A, but it can be switched to run at Site B. A query application runs at Site B, but it can be switch to A if and when replication latency exceeds a preset threshold.

The query workload is switched to Site A by using IBM Multi-site Workload Lifeline global workload balancing software. Lifeline is used for controlling on which site each of the workloads’ connections should be distributed.

6.2 Client’s previous experience with major upgrades

In this very fast-growing enterprise that is committed to being at the leading edge of technology and customer service offerings, change is a standard mode of operation. Major upgrades involve thousands of application changes and hundreds of DB2 schema changes. Although they pick a window for upgrades with the lowest workload, online transactions are 24x7 and need to be temporarily stopped during maintenance windows for upgrades.

Their past experience is summarized in Table 6-1. The production Site A must be taken offline for a few hours, or longer if there is a problem. There is no fallback. If the upgrade fails or the upgraded application misbehaves, the upgrade must be reverted and rescheduled to the next maintenance window.

Table 6-1 Outages during past experience

Step	Site A workloads	Outage duration
Normal state	OLTP, queries	
Stop application		5 to 10 minutes
Upgrade DB2 and applications at Site A Verify data integrity, test system		3 to 5 hours
Restart applications		5 to 10 minutes
Back to normal state	OLTP, queries	

Applications are unavailable over several hours, longer if there is a problem. Maintenance windows are still prevalent in the industry. Today, many large mainframe clients have application outages of several hours on a quarterly basis.

6.3 Improving the upgrade procedure with active-active sites

Leading-edge clients want to minimize risk and eliminate or greatly minimize any impact to their users during such upgrades.

The client had the following objectives:

- No application outage
- Finish load and restore R/O queries at Site B as fast as possible
- Prevent mismatches between sites
- Low-cost, reuse existing IT infrastructure as much as possible
- Minimize complexity

They also had challenges:

- Very large database: 100s of terabytes
- Very high transaction rates: Billions of changes per batch window
- Multiple data sharing groups, tens of thousands of DB2 tables
- Very disruptive periodic upgrades: Thousands of application changes, hundreds of DB2 schema changes combined with hardware and infrastructure upgrades
- Critical business data: No margin for error

6.4 Client choices for the upgrade with active-active sites

As explained in 6.3, “Improving the upgrade procedure with active-active sites” on page 59, factors such as availability objectives, complexity of the upgrade, size of data, and volume of transactions determine the most adequate procedure for routing and catching up after the upgrade and for repeating the upgrade. This section explains the choices made by the client.

6.4.1 Routing

A short outage (up to 2 minutes) is tolerated when switching connections from one site to another. The time is for terminating connections and changing routing table. This is a tradeoff.

An alternative to avoid outage during routing is to allow OLTP transactions to be distributed across sites in a manner that eliminates conflicts. This requires transaction-level routing that is aware of which data is being updated.

The client chose a simpler approach: Routing that is connection-based and based on Lifeline for controlling connections. Simplicity and time-to-market for approximately one minute interruption during site switch were the determining factors for accepting a trade-off of tolerating an approximately one minute interruption of the application during site switch.

6.4.2 Applying changes that took place during an upgrade

Resynchronization uses a combination of Q Replication (for simple schema changes) and proprietary resync processes (for application logic changes). This method is required because of complexity of changes.

6.4.3 Upgrading the second site

The best method is to upgrade once, on one site, and then copy the upgraded database over the old database on the other site. This method results in no downtime during the copy by avoiding a repeat of the upgrade procedure. The client chose to use PPRC-XD to copy the data. Then, the DB2 system is started from the new copy and remaining changes are caught up with Q Replication.

This client was already using disk copy, so this method allowed reusing their existing IT infrastructure. GDPS and PPRC are used together to create the consistency copy (freeze) without stopping the application.

Asynchronous remote copy supports a long distance without an impact on production. As illustrated in this paper, full PPRC-XD data copy takes around 4 hours for 120 TB, which is much faster than other options. Q Replication load using spilling to MQ was not a good choice given the client's constraints.

One caveat is there is no active failover site during the copy process. Disaster recovery is provided by the remote cold standby site. This risk can be minimized by having an extra set of disks so that DB2 does not have to be stopped for the duration of the copy. This is a cost-risk tradeoff.

6.5 Client upgrade procedure with active-active sites

By running business operations on Site B during upgrades of the production Site A, the client can now combine application upgrades with hardware, middleware, and configuration changes. Downtime for major quarterly upgrades is reduced from hours to minutes.

6.5.1 Upgrade procedure with active-active sites

The client's procedure stops the query workload during the resynchronization, but these queries do not have to be stopped. They can continue even if OLTP is stopped for interrupted availability of the query workload, although that means querying data that can be up to 10 to 30 minutes stale during the resynchronization step.

A further advantage to upgrades using active-active sites is that OLTP can stay longer on Site B if the migration has an issue at Site A.

Table 6-2 summarizes steps and outages.

Table 6-2 Steps and outages with active/active sites configuration

#	Step	Site A workloads	Site B workloads	Duration	Outage
0	Normal state: Replication from A to B	OLTP	Query	On-going	
1	Switch OLTP workload to Site B		Query	1 to 3 minutes	1 to 3 minutes
2	Upgrade Site A: Applications and, possibly, HW, DB2 version, configuration changes. Then, test new system.		Query OLTP	3 to 5 hours	None

#	Step	Site A workloads	Site B workloads	Duration	Outage
3	Stop OLTP workloads. Synchronize Site A with changes that took place at Site B during the upgrade. Tables without schema changes are synchronized with Q Replication Complex changes: The application handles the other changes.		Query	10 to 30 minutes	10 to 30 minutes
4	Switch workload to Site A			1 to 3 minutes	1 to 3 minutes
5	Upgrade Site B by copying over Site A using PPRC-XD disk copy and Q Replication. Copy over 100 TB of data. Catch up applying changes that accumulate during the XD copy.	OLTP Query		3 to 5 hours	None
6	Restart query workload on Site B	OLTP		1 minute	1 minute only for the query workload
	Back to normal state	OLTP	Query	Ongoing	

6.5.2 Switching workloads

Switching connections between sites for all data sharing groups takes one to three minutes. The workload switch can be divided into three phases: Preparation before the switch, the switch, and the post-switch actions. See Table 6-3.

Table 6-3 The preparation and actions to switch with active-active sites

Phase	Action	Site A workloads	Site B workloads	Duration	Outage
Prepare to switch	1. Stop or cancel batch jobs at Site A and B 2. CBU to expand capacity, online CPU at Site B	OLTP	Query	5 minutes	
Switch	1. Quiesce workload at Site A. 2. Stop replication of A to B after data is applied at B. 3.) Check status (TOR,QREP) and trigger switch. 4. Change Site A to soft fence (RREPL table space mode). 5. Change Site B to read-write. 6. Establish DB2 and IBM CICS connection to Site B.			1 to 3 minutes	1 to 3 minutes
Post-switch	1. Perform routine checks and business verification. 2. Start replication of B to A.		Query OLTP	5 minutes	

7

The zero-downtime copy procedure with PPRC and PPRC-XD

This chapter describes an innovative procedure for initializing a new active site without any impact on the source site, across unlimited distance. The procedure can be used for major system upgrades. After the first site is upgraded, the second site is upgraded by copying over the site already upgraded.

This chapter covers the following topics:

- Overview of the copy procedure
- Prerequisites for using PPRC-XD for DB2 copy
- Copy procedure steps

7.1 Overview of the copy procedure

After upgrading the first site and synchronizing it with the transactions that were executed on the down-level site during that upgrade, you upgrade the second site by copying over the site already upgraded. The second site is upgraded by copying Site A over Site B using PPRC-XD disk copy and Q Replication. This is completely transparent to applications running at Site A; no outage is necessary.

During the copy process, Site B is temporarily unavailable for failover. If a disaster occurs, recovery would rely on remote Site C, which is maintained as a cold standby with PPRC Global Mirror disk copy.

The DB2 instances at Site B are stopped, and the data of Site A that was copied from the local Metro Mirror copy at Site A is copied with PPRC-XD to the disks at Site B. After the copy is complete, the DB2 instances are restarted, and data is synchronized by starting Q Replication from A to B.

The process uses the PPRC local mirror copy at Site A. The remote copy using PPRC-XD is done from that copy. Figure 7-1 shows the process:

- PPRC-XD is used to copy all DB2 volumes asynchronously from local PPRC copy to remote Site B. When XD copy is 99% complete:
  a. Freeze local PPRC.
  b. Finish the XD copy.
  c. Resume local PPRC.
- After you get a consistent copy, restart DB2 at Site B from the PPRC-XD consistent copy.
- After DB2 is active and site-specific changes are made, use Q Rep to synchronizes Site B DB2:
  a. Drop secondary unique indexes at Site B.
  b. Start capture with restart LSN before PPRC freeze time and with `MACMTSEQ=00000000` (resend all committed transactions after restart of the LSN).

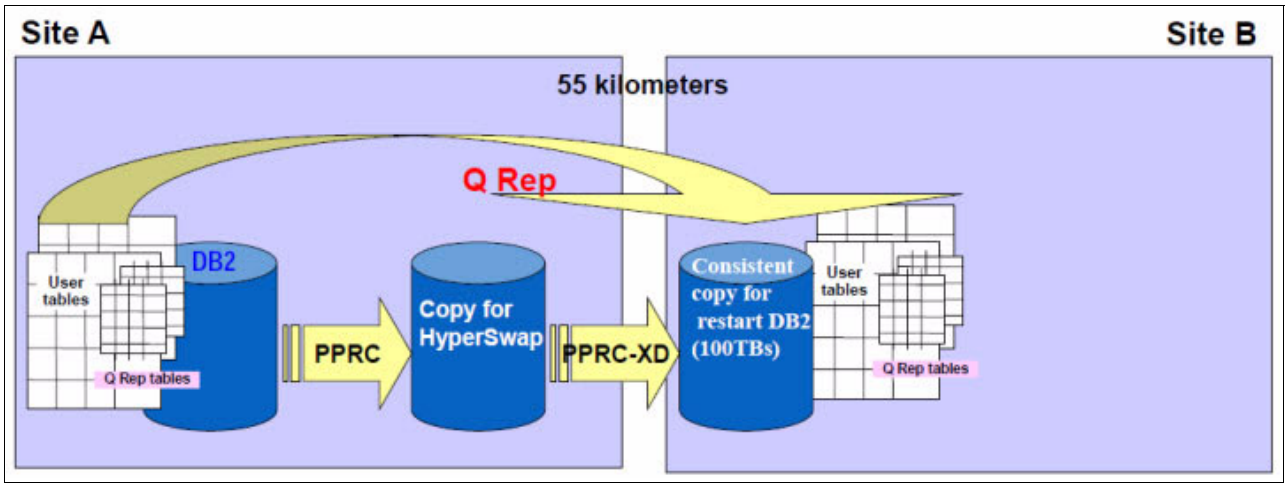

Figure 7-1 Replication technologies used

7.2 Prerequisites for using PPRC-XD for DB2 copy

To be able to use PPRC-XD to copy the data from the DB2 system at Site A to the DB2 system at Site B, you must have the following conditions:

- PPRC-XD primary and secondary DASD have the PPRC feature installed.
- A PPRC link connection is established between the primary and secondary DASD (with DWDM and IBM FICON director, if it's over a distance greater than 10 kilometers)
- DB2 data sets are isolated at the volume level, including user catalogs.
- DASD in Site B is not PPRC or XRC primary.
- All secondary volumes must be offline before copying.

Note: With additional DASD capacity at Site B, it is possible to do the PPRC-XD initial copy without stopping DB2 and with Q Replication running. This reduce the downtime of the DB2 instances at Site B.

7.3 Copy procedure steps

The subsections that follow describe the necessary steps.

7.3.1 Stop all replication

1. Stop Capture address space at Site A:

   ```
   / f myqcap,stop
   ```

2. Stop Apply at Site B:

   ```
   / f myqapp,stop
   ```

7.3.2 Stop DB2 instances at Site B

Before stopping DB2, any site-specific data needs to be saved (for example, the next value for DB2 sequence objects).

1. Save DB2 values for sequence objects
2. At Site B, save identity and sequence next values. Find the min/max/next value for a sequence:

   ```
   SELECT NAME,BIGINT(START), BIGINT(MAXASSIGNEDVAL),
   BIGINT(MINVALUE),BIGINT(MAXVALUE),BIGINT(RESTARTWITH) FROM SYSIBM.SYSSEQUENCES
   WHERE SEQTYPE='S' AND SCHEMA='XXXX' AND NAME  IN (XXX,XXX,……)
   ```

3. At Site B, find min/max/next for the tables where an identity column was created for replication key sequence:

   ```
   SELECT P.DNAME, DCOLNAME, BIGINT(S.START),  BIGINT(MAXASSIGNEDVAL),
   BIGINT(MINVALUE), BIGINT(MAXVALUE), BIGINT(RESTARTWITH)
    FROM SYSIBM.SYSSEQUENCESDEP P,
   SYSIBM.SYSSEQUENCES S,
   WHERE P.BSEQUENCEID = S.SEQUENCEID AND P.DTYPE = 'I' AND
   P.DCREATOR = 'NGDBA' AND P.DNAME  IN(XXXX,XXXX,XXXX…….)
   ```

4. Back up Site B unique DB2 tables.

   Back up the nonreplicated and Site B unique tables that need to be restored after copy (for example, the Site B query transaction journal table, APPLY MCGSYNC table).

5. Shut down DB2 at Site B:

   ```
   /cpf stop DB2
   ```

6. Force DB2-related structure at Site B:

   ```
   SETXCF FORCE,STR,STRNM=DSNDBOA_SCA    // Force the SCA structure
   SETXCF FORCE,STR,STRNM=DSNDBOA_LOCK1  // Force the LOCK1 structure
   ```

7. Deallocate the DB2-related user catalog at the site:

   ```
   F CATALOG,UNALLOCATE(ucatname)
   ```

7.3.3 Get a consistent disk copy for DB2 restart

1. First, set up PPRC-XD copy to Site B DB2 DASD.

   a. Set DB2 volumes at Site B to offline:

      ```
      V xxxx,OFFLINE
      ```

   b. Start PPRC-XD from Site A PPRC secondary DASD(DB2) to Site B Q Replication DASD:

      ```
      ESTPATH
          CESTPATH DEVN(X'devn') -
             PRIM(X'ssid' pri-wwnn X'lss') -
             SEC (X'ssid' sec-wwnn X'lss') -
             LINK(X'aaaabbbb',X'aaaabbbbb',X'aaaabbbb',X'aaaabbbb', -
             X'aaaabbbb',X'aaaabbbb') CGROUP(NO)
      ESTPAIR in XD mode
          CESTPAIR DEVN(X'devn') -
           PRIM(X'ssid' pri-serial X'cca' X'lss') -
           SEC (X'ssid' sec-serial X'cca' X'lss') -
           MODE(COPY) OPTION(XD)
      ```

2. Wait for 99% complete copy and no inflight UR at the source:

   ```
   CQUERY DEVN(X'devn')
   ....PERCENT of COPY COMPLETE = 99%
   ```

   This might take three to four hours for a very large database.

   Save the time stamp for when you restart Q Replication from A to B later. The restart time stamp must include any inflight transactions before the freeze. You will provide this time stamp as a value for the Q Capture LSN parameter when replication is restarted after the copy. A method to safely determine a time stamp that includes all inflight transactions is using `ZPARM URCHKTH` and `CHKFREQ`, as described in 7.3.6, “Start A to B replication with an LSN that includes inflight transactions” on page 70.

3. Freeze Site A PPRC copy:

   ```
   GDPS script: DASD = 'STOP SECONDARY'
   ```

   Suspend Site A PPRC, and establish a consistency point at secondary DASD:

   ```
   IMPACT on application: 1 second
   ```

4. Stop Site A to Site B PPRC-XD.

   Change PPRC-XD to synchronous mode, make sure that all volumes are in duplex status:

```
CESTPAIR DEVN(X'devn')-
     PRIM(X'ssid' pri-serial X'cca' X'lss') -
     SEC (X'ssid' sec-serial X'cca' X'lss') -
     MODE(COPY) OPTION(SYNC)
```

5. Delete the volume pairs, and make sure that all DB2 volumes are in simplex mode:

```
CDELPAIR DEVN(X'devn') -
   PRIM(X'ssid' pri-serial X'cca' X'lss') -
   SEC (X'ssid' sec-serial X'cca' X'lss')
```

You now have a point-in-time consistent copy of all DB2 data for all data sharing groups at Site B.

7.3.4 Restart DB2 instances at Site B and make site-specific changes

1. VARYON volumes for DB2:

```
V xxxx,ONLINE
```

2. Change DB2 location name and DDF setting. Use the DB2 DSNJU003 utility to change the Site B DB2 location name that different from Site A for cross sites DB2 access:

```
/DSNTLOG EXEC PGM=DSNJU003,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DSNA10.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DBOALOG1.DB1A.BSDS01 /* BSDS1 */
//SYSUT2 DD DISP=OLD,DSN=DBOALOG2.DB1A.BSDS02 /* BSDS2*/
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
DDF LOCATION=DSBDBOA,LUNAME=DB2BLU,          /*location and LU */
     GENERIC=DBOBGRP,                        /* generic LU name */
     RESPORT=4091,PORT=4009,SECPORT=0
```

3. Change the communications database for the IBM DRDA® access with Site A DB2 location names and add user DRDAOPER for DRDA access for ASNMON:

```
/* Update the CDB table to create the entries of site A DB2*/
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBOA)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA10) -
LIB('DSNDBOA.V10.RUNLIB.LOAD')
END
//SYSIN DD *
DELETE FROM SYSIBM.LOCATIONS;
DELETE FROM SYSIBM.LUNAMES;
DELETE FROM SYSIBM.MODESELECT;
DELETE FROM SYSIBM.LULIST;
DELETE FROM SYSIBM.USERNAMES;
INSERT INTO SYSIBM.LOCATIONS(LOCATION,LINKNAME)
VALUES('DSNDBOA','DBOAGRP');
INSERT INTO SYSIBM.LUNAMES(LUNAME,SECURITY_IN,
```

```
SECURITY_OUT,SYSMODENAME,MODESELECT)
VALUES('DBOAGRP','A','A','IBMRDB', 'Y');
INSERT INTO SYSIBM.MODESELECT(LUNAME,MODENAME)
VALUES('DBOAGRP','IBMRDB');
INSERT INTO SYSIBM.LULIST(LINKNAME,LUNAME)
VALUES ('DBOAGRP', 'DB1ALU');
INSERT INTO SYSIBM.LULIST(LINKNAME,LUNAME)
VALUES ('DBOAGRP', 'DB2ALU');
INSERT INTO SYSIBM.USERNAMES(TYPE,LINKNAME,NEWAUTHID)
VALUES('I','DBOAGRP','DRDAOPER');
INSERT INTO SYSIBM.USERNAMES(TYPE,LINKNAME,NEWAUTHID)
VALUES('O','DBOAGRP','DRDAOPER');
```

4. Enlarge DB2 NUMLKTS and NUMLKUS.

   Run the DSNTIJUZ to increase the number of locks of Site B because using of row level locking at Site B, but page-level locking at the source. For this customer case, the smallest DB2 row length is about 200 bytes. They use 4K page for most tables. So on average, each page might have up to 4K/200=20 rows per page. To estimate the increase in acquired lock we use the formulas shown in 3.3.3, “Why row-level locking is generally required at the target” on page 26.

   ```
   N=20*50%=10
   ```

   Enlarge NUMLKTS and NUMLKUS 10 times to prevent any lock escalation or Q Apply thread failure.

   ```
   //DSNTIZA EXEC PGM=ASMA90,PARM='OBJECT,NODECK'
   //STEPLIB  DD  DISP=SHR,DSN=DSNA10.SDSNLOAD
   //SYSLIB   DD  DISP=SHR,
   //         DSN=DSNA10.SDSNMACS
   //         DD  DISP=SHR,
   //         DSN=SYS1.MACLIB
   //SYSLIN   DD  DSN=&&LOADSET(DSNTILM5),
   //             DISP=(NEW,PASS),
   //             UNIT=3390,SPACE=(800,(50,50,2)),
   //             DCB=(BLKSIZE=800)
   //SYSPRINT DD  SYSOUT=*
   //SYSUDUMP DD  SYSOUT=*
   //SYSUT1   DD  UNIT=3390,SPACE=(800,(50,50),,,ROUND)
   //SYSUT2   DD  UNIT=3390,SPACE=(800,(50,50),,,ROUND)
   //SYSUT3   DD  UNIT=3390,SPACE=(800,(50,50),,,ROUND)
   //SYSIN    DD  *
       DSN6SPRM NUMLKTS=900000,
                NUMLKUS=1000000,
   ```

5. Restart DB2 group.

   Inflight transactions are rolled back. They will be captured from the DB2 log by Q Replication.

   Resolve DB2 object in LPL or GRECP status.

6. Restore data for nonreplicated tables and tables that unique for Site B, such as Site B query transaction journal table, MCGSYNC table.

7. Alter DB2 table space locking size to row-level-locking (RLL):

   ```
   -ALTER TABLE SPACE SAMPLE.TABLETS LOCKSIZE ROW
   ```

8. Alter DB2 sequence number or hidden column and identity column initial value and odd/even attribution.

9. Drop the secondary unique index:

   ```
   -DROP INDEX SAMPLE.INDEX2
   ```

10. Change Q Replication configuration:

    In the MCGSYNC table, restore the entries for Site B from the backup.

11. Change `SOFT FENCE` at both A and B and start the soft fence for SITE B:

    ```
    START DATABASE(trgDB) ACCESS(RREPL);
    ```

7.3.5 Validate subscriptions for Q Replication from A to B after the copy

Replication is started from A to B only to activate and validate the subscriptions. No data is replicated. The procedure prevents unexpected errors during this restart (for example, incorrect table name in the subscription, incorrect authorization, and so on) by validating before replication needs to be restarted. The basic idea is to restart Apply using the applytupto time stamp from the past; Apply validates the subscriptions and then stops.

1. Create the Capture-side subscriptions in the NEW state so they are activated automatically when the Capture task is started. Capture will allocate the sub_id. Do not suppress columns, because you will later force changes for conflicts while applying them to the initial copy. All columns are required if an update needs to be changed to an insert.

   Create the Apply-side subscriptions in the INACTIVE state, and use the conflict action of `FORCE`.

   ```
   UPDATE IBMQREP_SUBS set state = 'N', sub_id=NULL, CHANGED_COLS_ONLY='N',
   HAS_LOADPHASE='N'
   UPDATE IBMQREP_TARGETS set state='I', sub_id=NULL, conflict_action='F',
   conflict_rule='K'
   ```

2. *Repeat* the following steps until all subscriptions are successfully activated:

   a. Cold start Capture and start Apply with `applyupto` with a time stamp from the past.

      - Start Q Capture in COLD mode:

        ```
        //QCAP EXEC PGM=ASNQCAP,REGION=0M,TIME=NOLIMIT,
        // PARM='ENVAR("_CEE_ENVFILE=DD:MYENV")/SH1A CAPTURE_SCHEMA=QASN
        // STARTMODE=cold logstdout=y'
        ```

      - Start Q Apply `applyupto` using a time stamp from the past:

        ```
        //QAPP EXEC PGM=ASNQAPP,REGION=0M,TIME=NOLIMIT,
        // PARM='ENVAR("_CEE_ENVFILE=DD:MYENV")/APPLY_SERVER=AAOA AUTOSTOP=N
        // APPLY_SCHEMA=QASN logstdout=y'
        //SYSIN DD *
            applyupto=2000-01-01-00.00.00.000000
        ```

   b. Clear both receive and send queues, and verify that the queue depth is 0:

      ```
      CLEAR QLOCAL(queue_name)
      DISPLAY QSTATUS(queue_name) CURDEPTH
      ```

   c. Identify any subscription that failed to activate, and fix the error that caused the subscription activation failure (examples of possible errors are missing authorization or typos in a name):

      ```
      SELECT SUBNAME FROM QASN.IBMQREP_SUBS WHERE STATE <>'A' WITH UR;
      SELECT SUBNAME FROM QASN.IBMQREP_TARGETS WHERE STATE <>'A' WITH UR;
      // FIX THE ERROR
      //Reset the failed subscriptions to NEW at the source and INACTIVE at the
      target
      ```

```
UPDATE QASN.IBMQREP_SUBS SET SUB_ID=NULL , STATE='N' WHERE STATE <>'A'
UPDATE QASN.IBMQREP_TARGETS SET SUB_ID=NULL ,STATE='I' WHERE STATE <>'A'
```

7.3.6 Start A to B replication with an LSN that includes inflight transactions

Q Replication captures inflight transations from the DB2 logs at the time of the disk copy freeze and all changes that took place during the disk copy.

1. Start Apply at Site B.
2. Start Capture WARM with LSN override, using the LSN that was saved during disk copy.
3. When the replication catches up, latency is within 1-2 seconds:
   a. Stop replication.
   b. Rebuild the secondary unique index.
4. Warm restart capture and apply.
5. Rebuild secondary unique indexes.

   Because replication is restarted with an LSN from the past, the changes for some replicated transactions might already be in the copy, so applying them will conflict. The secondary unique indexes must be dropped until replication has caught up past the end of the disk copy point. See “Dropping secondary unique indexes during restart of Q Capture with old LSN” on page 75 for more about why indexes need to be dropped.

```
/* Rebuild secondary unique index*/
DROP INDEX SAMPLE.INDEX2 ;COMMIT;
/* Recreate secondary unique index*/
CREATE UNIQUE INDEX SAMPLE.INDEX2
ON SAMPLE.TABLE1
(COL1 ASC,
COL2 ASC)
USING STOGROUP SAMPLESG1
PRIQTY 5000 SECQTY 5000
ERASE NO
FREEPAGE 10 PCTFREE 5
GBPCACHE CHANGED
NOT CLUSTER
COMPRESS NO
BUFFERPOOL BP5
CLOSE YES
COPY NO
DEFER YES
DEFINE YES
PIECESIZE 4 G;
/* Rebuild index*/
//STEP1 EXEC PGM=DSNUTILB,PARM='DB0A,RBLDI001',MEMLIMIT=5000M
//SYSPRINT DD SYSOUT=*
//UTPRINT DD SYSOUT=*
//SYSIN DD *
REBUILD INDEX (SAMPLE.INDEX2)
SORTKEYS SORTDEVT SYSDA REUSE
```

A

Appendix

This appendix provides details and explanations of the following topics:

- Following Q Replication performance preferred practices
- Dropping secondary unique indexes during restart of Q Capture with old LSN
- Capturing restart (LSN) time stamp that includes all inflight transactions

Following Q Replication performance preferred practices

Q Replication is an IBM DB2 and IBM MQ application. Achieving the best possible performance requires following preferred practices for these two subsystems. There are also considerations for workload characteristics and database design that might affect replication performance. By following preferred practices, most IBM clients can achieve less than two seconds replication latency on average, even across thousands of kilometers.

Run InfoSphere Data Replication V10.2.1 or later

The first recommendation is to upgrade to version 10.2.1 or later. This version provides several significant performance improvements:

- DB2 IFI 306 log reader filtering, which requires DB2 V10 APAR PM90568 or DB2 V11
- Using DB2 multi-row-insert (MRI), which requires DB2 V10 APAR PM90568/UK97013
- Q Apply improved handling of key dependency on a single table
- Q Capture throughput performance improvements

Adopt configuration recommendations

- Use a dedicated Q Manager for each Capture and Apply.

  This allows larger buffer pools and prevents any potential logging contention.
- Run Capture and Apply tasks with the Workload Manager service class as high as DB2 MSTR.
- Use multiple parallel send queues, at least two per consistency group.
- Consider column suppression (`changed_cols_only=y`).

  It reduces the amount of data transmitted over the network for updates and deletes, but it has implications during catchup after an upgrade. Column suppression must be turned off during the catchup phase to use `conflict_action of 'F'` (`Force`).
- Use DB2 row-level locking (at least at the target).

  Increase DB2 `LOCKNUM` accordingly if you are changing from page to row-level locking.
- Define subscription replication keys to match a unique index.

Tune IBM MQ

- Run V7.1 or later.
- Check these parameters:

  batchlim(1MB) This parameter requires PTF PM79000/UK90868

  batchsz(800) This determines the maximum number of messages in a unit of work (UOW, or transaction) that are sent across an IBM MQ channel. It is configured at the sender (capture) end of the channel between Capture and Apply queue managers. For an OLTP workload, 800 is best. You must also set Q Capture MAX_TRANS and COMMIT_INTERNAL accordingly.
- Define buffer pool as 1 GB divided among the receive queues for the queue manager.
- Enable SSD (this affects pageset I/O speed).

- Enable MQ buffer pool read-ahead (z/OS). This requires PM90110 and PM81785. You can use the MQSC commands:

  ```
  RECOVER QMGR(TUNE READAHEAD ON) - buffer pool read-ahead
  RECOVER QMGR(TUNE RAHGET ON)
  ```

  The functions can be activated dynamically by entering at a console or added to the queue manager configuration by adding to a file processed in the `CSQINP2` concatenation during queue manager start.

  The read-ahead adds significant performance benefits for **`MQGET`** when the queue is spilling to the pageset.

- Consider upgrading to V8 or later to get significant improvements for Q Rep in MQ):
  - A 64-bit buffer pool provides more and faster storage for queue access. It allows more data on the queue before it is spilled to disk, potentially 64 GB of (each) `XMITQ` entirely in storage.
  - Enhanced deferred write (cast-out) means faster **`MQPUT`** by the capture when the queue exceeds the buffer pool.
  - Enhanced logging provides benefits, particularly for larger messages.

Consider workload transactions

The following considerations apply to the DB2 transactions:

- Avoid defining very large object (LOB) data. Use inline LOBs whenever possible.
- Avoid very large transactions that execute millions of row changes in a single unit of work.
- Commit frequently.

Configure Transmission Control Protocol

Production Q Replication systems are likely to be replicating data over significant distances, so the network roundtrip time can be significant. There are a couple of effects of high network latency:

- At the MQ level, it becomes apparent when a channel batch completes. The MQ network protocol is to send a confirmation flow to the partner receiver channel, causing it to commit the batch of messages and acknowledge the receipt. Larger batches mean fewer confirm flows and less time spent by the channel waiting for the partner to acknowledge receipt of a batch.
- At the Transmission Control Protocol (TCP) protocol level, the amount of data that can be sent before a packet acknowledgment is received is controlled by the TCP receive window size. The sender continues to transmit data until the receive window is met and then blocks until acknowledgements are received. For high-capacity or high-latency networks, unless the receive window is sufficiently large, the sending process could be blocked frequently.

  IBM Communication Server incorporates a facility known as Dynamic Right Sizing (DRS). It is enabled when the TCP/IP default receive buffer size that is specified with the TCPRCVBUFRSIZE profile statement is configured as 64 KB or higher. With DRS enabled, if necessary, a connection's advertised receive window size will be allowed to grow very large to accommodate the amount of data in transit on a high-bandwidth or high-latency network

- To enable an MQ to use DRS, it is necessary to alter TCP configuration and restart the Chinit[1]. Alternatively, MQ APAR PM71966 allows manipulation of the TCP send and receive buffer sizes for all MQ channels from a single MQ, rather than for the entire TCP stack. MQ configures the send and receive window sizes of TCP connections so that DRS is enabled.

Tune Q Capture

Enable DB2 log read IFI 306 filtering by installing DB2 APAR PM90568 or DB2 V11 and IIDR V10.2.1 or later. IFI 306 filtering is enabled by default at these product levels. Consider adjusting the following Q Capture parameters:

sleep_interval 50 ms

max_trans=800 Default is 200. Commit after 800 MQ messages have been put or the `commit_interval` is reached.

commit_interval 200 ms

memory_limit No significant impact. 200 MB is OK, but increases if spilling because of monster transactions. Check `IBMQREP_CAPMON(trans_spilled)`.

trans_batch_sz=4 Or more. But only if there is no issue with “hot row” workloads. If `applymon(dependency_delay)` is high, use `trans_batch_sz=1` (default).

The goal is to get the MQ message size more than 10 KB, which is optimal. The goal in using `trans_batch_sz` is to get MQ message size to more than 10 KB, which is optimal for MQ performance. With Q Replication, one MQ message (or more, if needed) is used to transmit each capture DB2 transaction. If the average DB2 transaction size modifies 2 KB of data, then `trans_batch_sz=5` gives an average message size of 10 KB by batching 5 DB2 transactions per MQ message.

max_message_size = 1 MB Combine this with `max_message_size = 1 MB` (large transactions will not be batched beyond 1 MB).

If you are using Q Apply synchronized Apply for disaster recovery, set `HEARTBEAT_INTERVAL= 200 ms`.

Tune Q Apply

- Populate the PARALLEL_SENDQS column in the IBMQREP_RECVQUEUES table with a value of `Y` (yes) so that Q Apply detects the parallelism and a `get` method by `msgid` is used. Set this parameter even when you are *not* using parallel send queues.
- Use the following settings:

  maxagents 32 or more. Check `agent_sleep_time`. Maximum supported is 128.

  This requires adequate CPU and DB2 resources. More agents might not perform better if resources are inadequate. Start with the default of 16.

[1] Chinit address space is used in the running of your IBM MQ (or IBM WebSphere MQ) subsystem on z/OS.

maxagents_correlid	This is needed if lock timeouts or deadlocks during batch jobs are observed when a batch job is replicated. That is, NUM_DEADLOCKS in the Q Apply monitor table ASNQ_APPLYMON is not zero.
multi-row-insert=Y	This is the default. Use DB2 multi-row insert (V10.2.1), which requires DB2 10 APAR PM90568/UK97013.
prune_batch_sz=100	Q Apply deletes 100 messages from the receive queue per MQ commit. This helps keeps the receive queue small.

Tune DB2 at the target

If high wait time in the accounting log read I/O or other service task or *not* accounted time, consider changing these settings:

- Increase the DB2 subsystem OUTBUFF parameter for the members that are running the Capture program.
- Use fast disk devices for both the DB2 active logs and the DB2 bootstrap data set (BSDS).
- Reduce log data volume.
  - Use DB2 data compression for insert-intensive tables. The tradeoff is CPU cost for decompression at log read time.
  - Optimize table design to minimize log record size. This is applicable only to DB2 Basic Row Format (BRF).

Tune direct access storage devices

There are higher performance requirements for MQ log data sets:

- Striped I/O
- Separate at different storage controls with lower activity
- Enable zHPF to get more efficient channel use

Dropping secondary unique indexes during restart of Q Capture with old LSN

Table A-1 on page 76 illustrates why Q Replication conflict resolution cannot fix a fuzzy database copy when changes already present in the copy might be resent by the Q Capture program. The table illustrates the problem that can happen if and when the indexes are not dropped. For some sequence of operations, the conflict cannot be resolved, and the Q Apply program terminates, because error_action is defined as 'S' for *stop*.

Table A-1 Unresolvable conflicts with secondary unique indexes when restarting with LSN from the past

Source			Target		
Time	Operations	Result	Operations	Q Apply result	Target copy
to	Start capturing log (capstart subscription or restart with LSN				
t1	LIVE source unload begins				
t2	INSERT (1,B)	(1,B)			
t3	DELETE (1,B)				
t4	INSERT (1,A)	(1,A)			
t5	INSERT (2,B)	(2,B)			
t6	Source unload ends				(1,A),(2,B)
t7	INSERT (3,C)	(1,A),(2,B),(3,C)	Q Apply loads the table	LOAD DONE	(1,A),(2,B)
t8		(1,A),(2,B),(3,C)	INSERT (1,B)	-803 on REPLICATION KEY (1,B) clashes with (1,A) Follow CONFLICT_ACTION=F	(1,A),(2,B)
t9		(1,A),(2,B),(3,C)	Q Apply try to FORCE: UPDATE set C2='B' where C1=1;	-803 SECONDARY INDEX Follow ERROR_ACTION APPLY COMES DOWN	(1,A),(2,B)

Table A-2 on page 77 illustrates how the same conflict is resolved without the secondary unique indexes. Assume that the table named T1 has two unique indexes. C1 is the primary key, which is used as the replication key. C2 is another unique constraint on table T1, with a definition like this one:

```
TABLE T1 (C1 INT PRIMARY KEY, C2 CHAR(1) UNIQUE);
```

Q Apply is defined with `conflict_rule = 'K'`, which checks the replication key values when applying SQL at the target. Using `conflict_action = 'F'` for conflicts that are caused by either a duplicate row or a row not found forces the value from the source by, for instance, changing the insert to an update for a row that is already present at the target.

After the index on column C2 is dropped, rows with the same value for this column are allowed, and no conflict is detected. At time t9 in Table A-2 on page 77, there will be two rows in the table with the value of B, but it is deleted after all changes are replicated, and the target becomes identical to the source.

Table A-2 How conflict is resolved when restarting with LSN from the past and a fuzzy image

Source			Target		
Time	Operations	Result	Operations	Q Apply result	Target copy
to	Start capturing log (capstart subscription or restart with LSN		Dropping the secondary unique index.		
t1	LIVE source unload begins				
t2	INSERT (1,B)	(1,B)			
t3	DELETE (1,B)				
t4	INSERT (1,A)	(1,A)			
t5	INSERT (2,B)	(1,A),(2,B)			
t6	Source unload ends				(1,A),(2,B)
t7	INSERT (3,C)	(1,A),(2,B),(3,C)	Q Apply loads the table	LOAD DONE	(1,A),(2,B)
t8		(1,A),(2,B),(3,C)	INSERT (1,B)	-803 on REPLICATION KEY CONFLICT_ACTION=F	(1,A),(2,B)
t9		(1,A),(2,B),(3,C)	Q Apply try to FORCE: UPDATE set C2='B' where C1=1;	OK	(1,B),(2,B)
t10		(1,A),(2,B),(3,C)	DELETE (1,B)	OK	(2,B)
t11		(1,A),(2,B),(3,C)	INSERT (1,A)	OK	(1,A),(2,B)
t12		(1,A),(2,B),(3,C)	INSERT (2,B)	-803 on REPLICATION KEY (1,B) clashes with (1,A) Follow CONFLICT_ACTION=F	(1,A),(2,B)
t13		(1,A),(2,B),(3,C)	Q Apply try to FORCE: UPDATE set C2='B' where C1=2;	OK	(1,A),(2,B)
t14		(1,A),(2,B),(3,C)	INSERT (3,C)	OK	(1,A),(2,B), (3,C)

Capturing restart (LSN) time stamp that includes all inflight transactions

In the DB2 DSN6SYSP macro, the DB2 subsystem parameters URCHKTH and CHKFREQ determine the frequency at which uncommitted unit of recovery (UR) information is reported.

CHKFREQ Sets the system checkpoint frequency to the specified number of minutes or log records.

URCHKTH	Determines after a certain number of checkpoint cycles DB2 issues a warning message to the console for an uncommitted unit of recovery (UR).
CHKFREQ* URCHKTH	Determines after how long a long-running UR will be reported.

Follow this procedure to get the LSN for capture restart after copy:

1. Wait for PPRC-XD copy to reach the 99% complete.
2. Get the current time stamp as LSN1.
3. Wait until CHKFREQ*URCHKTH time to be reached.
4. If DSNR035I was not reported, go to next step for Freeze Site A PPRC copy. The LSN1 can be used as the capture restart LSN after copy.
5. If DSNR035I was reported, loop from 1) to 4) until you get the valid restart LSN.

When the PPRC-XD copy reaches 99% complete, record a start time stamp and wait for CHKFREQ*URCHKTH time. If no DSNR035I was issued by then, suspend the PPRC. The recorded start time stamp can be used as the capture restart LSN. If DSNR035I is reported in the time interval that corresponds to CHKFREQ*URCHKTH, wait one more CHKFREQ*URCHKTH time until there is no DSNR035I reported and use the new recorded time stamp.

Alternative method: Use group restart of the LSN at Site B

During Site B DB2 group restart, the UR status message DSNR007I, shown in Example A-1, displays the begin time for all URs that need to be recovered. The oldest inflight transaction begin time can be determined by comparing all DB2 members' DSNR007I messages. The restart LSN is the oldest unit of recovery start time.

Example A-1 DSNR007I message sample

```
21.47.15 SO111488  DSNR004I  -PB11 RESTART...UR STATUS COUNTS  372
    372             IN COMMIT=0, INDOUBT=0, INFLIGHT=2, IN ABORT=0, POSTPONED ABORT=0
 21.47.15 SO111488  DSNR007I  -PB11 RESTART...STATUS TABLE  373
    373             T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
    373             - -------- ------------ -------- -------- - ------------ --- --------
    373             B BATCH    OLRPTSH      CBODBAT  CBODPBPD F 0371B30EC997 017 17:46:50
    373             S CI11PA11 ENTRTC020046 CBODUSER CBODPOPD F 0371B3137A98 017 17:46:50
```

Glossary

A

AOM. asynchronous operations manager.

application system. A system made up of one or more host systems that perform the main set of functions for an establishment. This is the system that updates the primary disk volumes that are being copied by a copy services function.

asynchronous operation. A type of operation in which the remote copy XRC function copies updates to the secondary volume of an XRC pair at some time after the primary volume is updated. Contrast with synchronous operation.

B

backup. The process of creating a copy of data to ensure against accidental loss.

bidirectional replication. A replication configuration where tables at a source node are also the target of a replication solution; this configuration is required when changes made to replicated tables can occur at two nodes and must be reflected at both nodes.

C

cache. A random access electronic storage in selected storage controls used to retain frequently used data for faster access by the channel.

Capacity BackUp. Capacity Backup (CBU) is a hardware feature that allows the temporary activation of central processors (CPs) on IBM servers. CBU provides the ability to concurrently increment the capacity of your processor, using Licensed Internal Code (LIC), in the event of an unforeseen loss of substantial System z computing capacity at one or more of your eligible sites. Central processors can be dynamically added to the agreed System z model for a 90 days period with no system power-down and no associated re-IML/IPLs.

CBU. See Capacity Backup.

central processor complex (CPC). The unit within a cluster that provides the management function for the storage server. It consists of cluster processors, cluster memory, and related logic.

channel connection address (CCA). The input/output (I/O) address that uniquely identifies an I/O device to the channel during an I/O operation.

channel interface. The circuitry in a storage control that attaches storage paths to a host channel.

consistency group time. The time, expressed as a primary application system time-of-day (TOD) value, to which XRC secondary volumes have been updated. This term was previously referred to as “consistency time”.

continuous availability. Undisrupted access to business critical applications 24 hours, 7 days a week.

consistent copy. A copy of a data entity (for example a logical volume) that contains the contents of the entire data entity from a single instant in time.

control unit address. The high-order bits of the storage control address, used to identify the storage control to the host system.

D

dark fibre. A dedicated fibre link between two sites that is dedicated to use by one client.

DASD. direct access storage device.

data in transit. The update data on application system DASD volumes that is being sent to the recovery system for writing to DASD volumes on the recovery system.

data mover. See system data mover.

device address. The ESA/390 term for the field of an ESCON device-level frame that selects a specific device on a control unit image. The one or two leftmost digits are the address of the channel to which the device is attached. The two rightmost digits represent the unit address.

device number. The ESA/390 term for a four-hexadecimal-character identifier, for example 13A0, that you associate with a device to facilitate communication between the program and the host operator. The device number that you associate with a subchannel.

Device Support Facilities program (ICKDSF). A program used to initialize DASD at installation and perform media maintenance.

DFDSS. Data Facility Data Set Services is an IBM licensed program to copy, move, dump, and restore data sets and volumes.

DFSMSdss. A functional component of DFSMS/MVS used to copy, dump, move, and restore data sets and volumes.

disaster recovery (DR). Recovery after a disaster, such as a fire, that destroys or otherwise disables a system. Disaster recovery techniques typically involve restoring data to a second (recovery) system, then using the recovery system in place of the destroyed or disabled application system. See also recovery, backup, and recovery system.

dual copy. A high availability function made possible by the nonvolatile storage in cached IBM storage controls. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical storage subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

duplex pair. A volume composed of two physical devices within the same or different storage subsystems that are defined as a pair by a dual copy, PPRC, or XRC operation, and are not in suspended or pending state. The operation records the same data onto each volume.

DWDM. Dense Wavelength Division Multiplexor. A technique used to transmit several independent bit streams over a single fiber link.

E

extended remote copy (XRC). A hardware-based and software-based remote copy service option that provides an asynchronous volume copy across storage subsystems for Disaster Recovery, device migration, and workload migration.

F

fixed utility volume. A simplex volume assigned by the storage administrator to a logical storage subsystem to serve as working storage for XRC functions on that storage subsystem.

FlashCopy. A point-in-time copy services function that can quickly copy data from a source location to a target location.

floating utility volume. Any volume of a pool of simplex volumes assigned by the storage administrator to a logical storage subsystem to serve as dynamic storage for XRC functions on that storage subsystem.

J

journal. A checkpoint data set that contains work to be done. For XRC, the work to be done consists of all changed records from the primary volumes. Changed records are collected and formed into a “consistency group”, and then the group of updates is applied to the secondary volumes.

K

km. kilometer.

L

Licensed Internal Code (LIC). Microcode that IBM does not sell as part of a machine, but licenses to the customer. LIC is implemented in a part of storage that is not addressable by user programs. Some IBM products use it to implement functions as an alternative to hard-wired circuitry.

link address. On an ESCON interface, the portion of a source or destination address in a frame that ESCON uses to route a frame through an ESCON director. ESCON associates the link address with a specific switch port that is on the ESCON director. Equivalently, it associates the link address with the channel subsystem or controller link-level functions that are attached to the switch port.

l**og sequence number**. The log sequence number (LSN) is assigned by DB2 for each data change and is used by SQL Replication programs to keep data synchronized between source and target.

logical partition (LPAR). The ESA/390 term for a set of functions that create the programming environment that is defined by the ESA/390 architecture. ESA/390 architecture uses this term when more than one LPAR is established on a processor. An LPAR is conceptually similar to a virtual machine environment except that the LPAR is a function of the processor. Also, the LPAR does not depend on an operating system to create the virtual machine environment.

LSN. See log sequence number.

logical subsystem (LSS). The logical functions of a storage controller that allow one or more host I/O interfaces to access a set of devices. The controller aggregates the devices according to the addressing mechanisms of the associated I/O interfaces. One or more logical subsystems exist on a storage controller. In general, the controller associates a given set of devices with only one logical subsystem.

N

Node. A source or target DB2 for z/OS subsystem that participates in a replication configuration; changes made to tables at a source node are replicated to tables that reside at the target node.

O

orphan data. Data that occurs between the last, safe backup for a recovery system and the time when the application system experiences a disaster. This data is lost when either the application system becomes available for use or when the recovery system is used in place of the application system.

P

peer-to-peer remote copy (PPRC). A hardware-based remote copy option that provides a synchronous volume copy across storage subsystems for Disaster Recovery, device migration, and workload migration. It was renamed Metro Mirror by IBM.

pending. The initial state of a defined volume pair, before it becomes a duplex pair. During this state, the contents of the primary volume are copied to the secondary volume.

PPRC. See peer-to-peer remote copy.

PPRC dynamic address switching (P/DAS). A software function that provides the ability to dynamically redirect all application I/O from one PPRC volume to another PPRC volume.

PPRC-XD. Peer-to-peer remote copy extended distance.

primary device. One device of a dual copy or remote copy volume pair. All channel commands to the copy logical volume are directed to the primary device. The data on the primary device is duplicated on the secondary device. See also secondary device.

PTF. program temporary fix.

Q

query offloading. A workload balancing strategy that offloads some of the query activity to a DB2 for z/OS that is separate from the online transaction processing DB2 for z/OS.

R

IBM RACF®. Resource Access Control Facility.

recovery point objective (RPO). The maximum tolerable period in which data might be lost from an IT service due to a major incident.

recovery system. A system that is used in place of a primary application system that is no longer available for use. Data from the application system must be available for use on the recovery system. This is usually accomplished through backup and recovery techniques, or through various DASD copying techniques, such as remote copy.

recovery time objective (RTO). The amount of time the business can be without the service, without incurring significant risks or significant losses

remote copy. A storage-based Disaster Recovery and workload migration function that can copy data in real time to a remote location. Two options of remote copy are available. See peer-to-peer remote copy and extended remote copy.

resynchronization. A track image copy from the primary volume to the secondary volume of only the tracks that have changed since the volume was last in duplex mode.

RPO. See recovery point objective.

RTO. See recovery time objective.

S

secondary device. One of the devices in a dual copy or remote copy logical volume pair that contains a duplicate of the data on the primary device. Unlike the primary device, the secondary device may accept only a limited subset of channel commands.

sidefile. A storage area used to maintain copies of tracks within a concurrent copy domain. A concurrent copy operation maintains a sidefile in storage control cache and another in processor storage.

simplex state. A volume is in the simplex state if it is not part of a dual copy or a remote copy volume pair. Ending a volume pair returns the two devices to the simplex state. In this case, there is no longer any capability for either automatic updates of the secondary device or for logging changes, as would be the case in a suspended state.

site. A server where applications run; includes key resources like DB2 for z/OS and IBM MQ for z/OS

site table. Entity within GDPS that is created from information in the GEOPLEX DOMAINS. It contains a list of all the systems in the GDPS environment.

suspended state. When only one of the devices in a dual copy or remote copy volume pair is being updated because of either a permanent error condition or an authorized user command. All writes to the remaining functional device are logged. This allows for automatic resynchronization of both volumes when the volume pair is reset to the active duplex state.

synchronization. An initial volume copy. This is a track image copy of each primary track on the volume to the secondary volume.

synchronous operation. A type of operation in which the remote copy PPRC function copies updates to the secondary volume of a PPRC pair at the same time that the primary volume is updated. Contrast with asynchronous operation.

system data mover. A system that interacts with storage controls that have attached XRC primary volumes. The system data mover copies updates made to the XRC primary volumes to a set of XRC-managed secondary volumes.

T

timeout. The time in seconds that the storage control remains in a “long busy” condition before physical sessions are ended.

U

Unidirectional replication. A replication configuration where tables at a source node are never the target of a replication configuration

Related publications

The publications listed in this section are considered particularly suitable for more detailed information about the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information. Some publications referenced in this list might be available in softcopy only.

- *Understanding and Using Q Replication for High Availability Solutions on the IBM z/OS Platform, SG24-8154*
- *InfoSphere Data Replication for DB2 for z/OS and WebSphere Message Queue for z/OS: Performance Lessons, REDP-4947*
- *GDPS Family: An Introduction to Concepts and Capabilities*, SG24-6374
- *Always On: Assess, Design, Implement, and Manage Continuous Availability*, REDP-5109
- *Always On: Business Considerations for Continuous Availability*, REDP-5090
- *GDPS/Active-Active Overview and Planning*, SG24-8241

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft, and additional materials, on the IBM Redbooks web page:

ibm.com/redbooks

Other publications

This publication is also relevant for further information:

IBM Multi-site Workload Lifeline V2.0 User's Guide Version 2.0, SC27-4653

Online resources

These websites are also helpful:

- IBM GDPS website

  http://www.ibm.com/systems/z/advantages/gdps/getstarted/gdpsaa.html
- IBM InfoSphere Data Replication Version 10.2.1 section of the IBM Knowledge Center

  http://ibm.co/1soJ16o
- IBM Multi-site Workload Lifeline

  http://www.ibm.com/software/products/en/network-lifeline

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

The Value of Active-Active Sites with Q Replication for IBM DB2 for z/OS

An Innovative IBM Client's Experience

Redpaper™

Learn best practices for continuous availability with active-active sites

Deploy sites using IBM InfoSphere Q Replication and Lifeline

Study an IBM client's solution

Any business interruption is a potential loss of revenue. Achieving business continuity involves a tradeoff between the cost of an outage or data loss with the investment required for achieving the recovery point objective (RPO) and recovery time objective (RTO).

Continuous system availability requires scalability, as well as failover capability for maintenance, outages, and disasters. It also requires a shift from standby to active-active systems. *Active-active* sites are geographically distant transaction processing centers, each with the infrastructure to run business operations and with data synchronized by using database replication, such as the Q Replication technology that is part of IBM InfoSphere Data Replication software.

This IBM Redbooks publication describes preferred practices and introduces an architecture for continuous availability and disaster recovery that is used by a very large business institution that runs its core business on IBM DB2 for z/OS databases. This paper explains the technologies and procedures that are required for the implementation of an active-active sites architecture. It also explains an innovative procedure for major IT upgrades that uses Q Replication for DB2 on z/OS, Multi-site Workload Lifeline, and Peer-to-Peer Remote Copy/Extended Distance (PPRC-XD).

This paper is of value to decision makers, such as executive and IT architects, and to database administrators who are responsible for design and implementation of the solution.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-5140-00