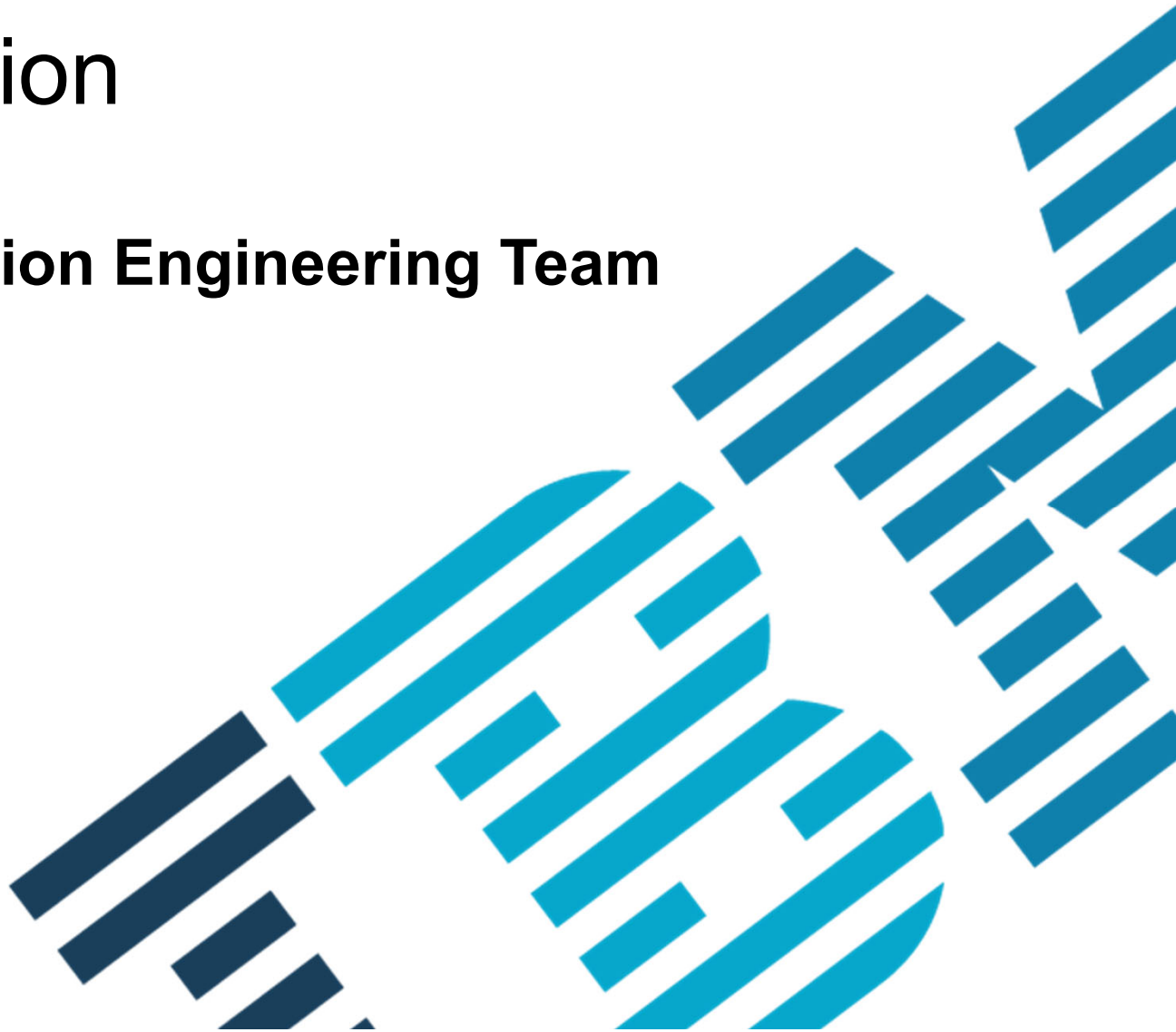
IBM DB2 HADR
State of the Union

Dale McInnis

STSM – Client Adoption Engineering Team

IBM Canada Ltd.



Safe Harbor Statement

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MAY NEVER BE DELIVERED BY I BM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

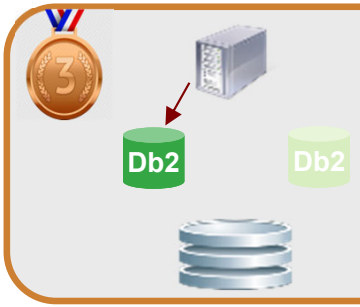
WHO AM I

- Team Lead Db2 Client Adoption Engineering
- 38 years with IBM, 25 years at the Db2 LUW High Availability Architect
- Information Systems Professional (I.S.P) Certified
- Information Technology Certified Professional (ITCP)
- 11 Patents in database resiliency
- Published author
- IDUG Speaker Hall of Fame member
- Ph.D. Student at the Ontario Technical University

Agenda

- **Comparison to other DB2 HA offerings**
- HADR Review
- HADR Sync modes
- HADR Multiple Standby
- HADR Configuration
- HADR Monitoring
- Recent Improvements

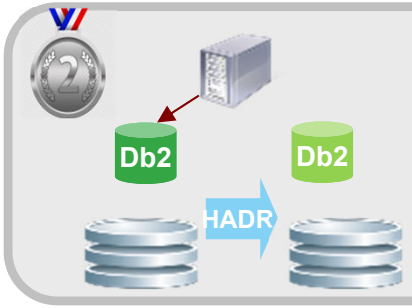
Db2 HA Options : 24x7x365 Zero Data Loss HA for OLTP



Integrated Clustering

99% availability

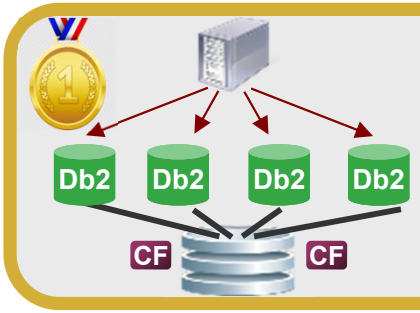
- Active/passive
- Hot/cold, with failover typically in minutes
- Easy to setup
- Db2 ships with integrated TSA failover software
- No additional licensing required



HADR

99.99% availability

- Active/passive or active/active (with Reads on Standby)
- Hot/warm or hot/hot (with RoS), with failover typically less than one minute
- Easy to setup
- Zero data loss failover in < 1 minute
- Db2 ships with integrated TSA
- Minimal licensing (full licensing required if standby is active)
- Perform system and database updates with minimal interruption

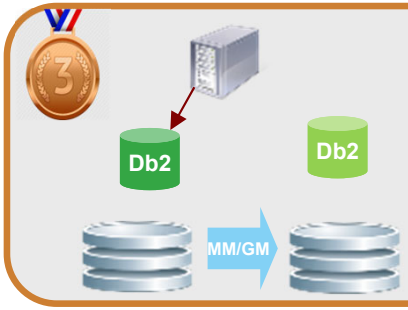


pureScale

99.99+% availability

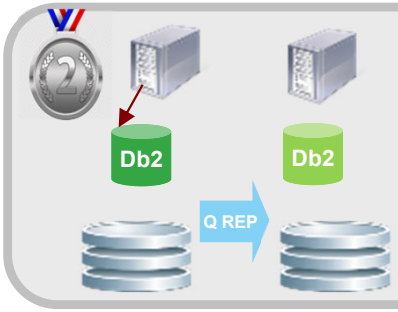
- Active/active
- Hot/hot, with automatic and online failover
- Integrated solution includes CFs, clustering, and shared data access
- Perform system and database updates in rolling online fashion
- Also works with HADR (single target)
- Horizontal scaling

Db2 Disaster Recovery Options



Log Shipping / Storage Based Replication

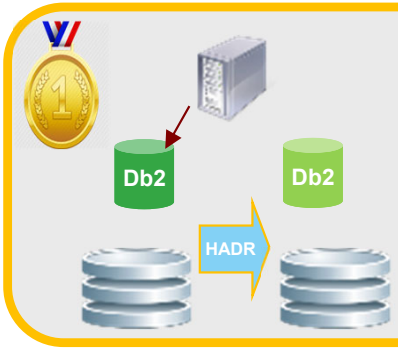
- Active/passive
- Hot/cold, with failover typically in minutes
- Asynchronous
- Complete DB replication only



Logical Replication

- Active/active (updates require conflict resolution / avoidance
- Hot/Hot (Instant failover)
- Asynchronous

- Added flexibility
  - Subsetting
  - Different versions
  - Different topology
  - Multiple standby
  - Time delay
- DDL considerations



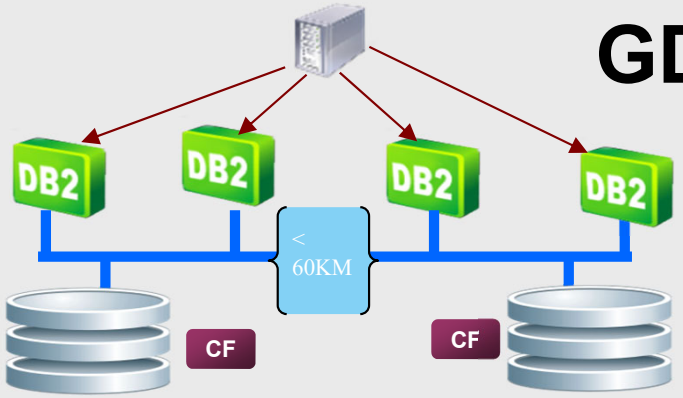
HADR

- Active/passive or active/active (with Reads on Standby)
- Hot/warm or hot/hot (with RoS), with failover typically less than one minute
- Easy to setup

- Complete DB Replication
- Minimal licensing (full licensing required if standby is active)
- Time Delay
- Perform system and database updates minimal interruption

Situational Platinum

GDPC



- Active / active (fully coherent)
- Hot / hot (**online** failover)
- Synchronous
- Complete DB replication
- Continuous testing of DR site
- Distance limitations
- Only available through lab services

DB2 LUW HA Guidelines

- Check out IDUG Technical Content article

https://www.idug.org/news/db2-luw-high-availability-ha-guidelines

Discusses the options available for high availability:

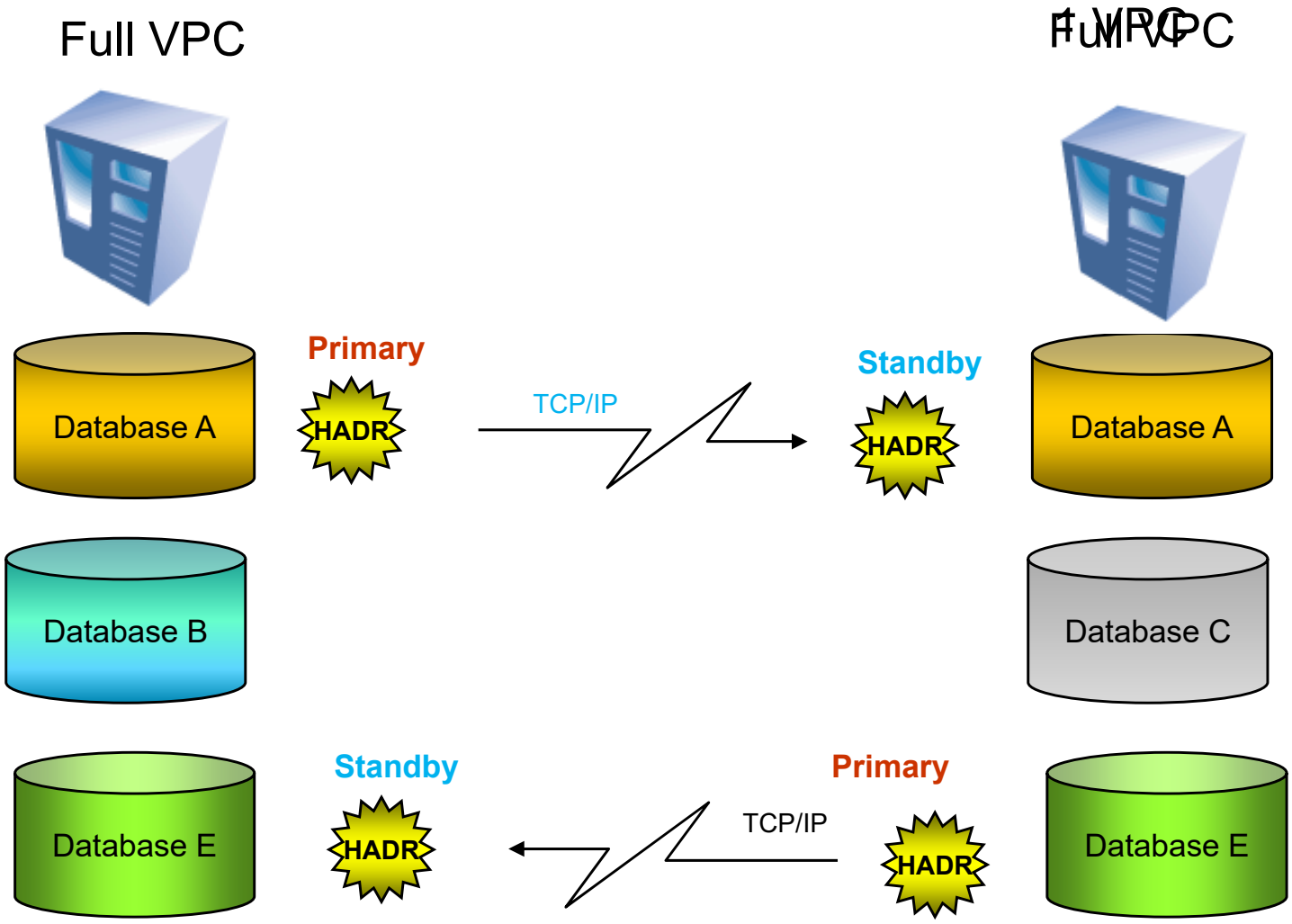
- Clustering with shared storage (active-passive)
- HADR (High Availability Disaster Recovery)
- Db2 pureScale
- Logical replication

Agenda

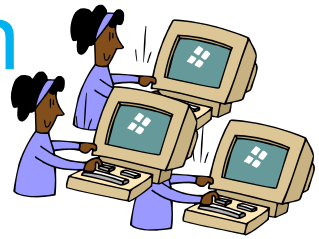
- Comparison to other DB2 HA offerings
- **HADR Review**
- HADR Sync modes
- HADR Multiple Standby
- HADR Configuration
- HADR Monitoring
- Recent Improvements

Scope of Action

HADR replication takes place at the database level.



HADR Implementation



Primary Connection

Client Reroute

DB2 Engine

PRIMARY SERVER

STANDBY SERVER

DB2 Engine

TCPIP

HADR

Log Pages

Log Pages

HADR

Log Records

Shredder

Replay Master

Log Pages

Log Records

Log Pages

Log Writer

Log Reader

Log Writer

Log Reader

Replay Workers

Tables

Indexes

Logs

Old

Logs

Logs

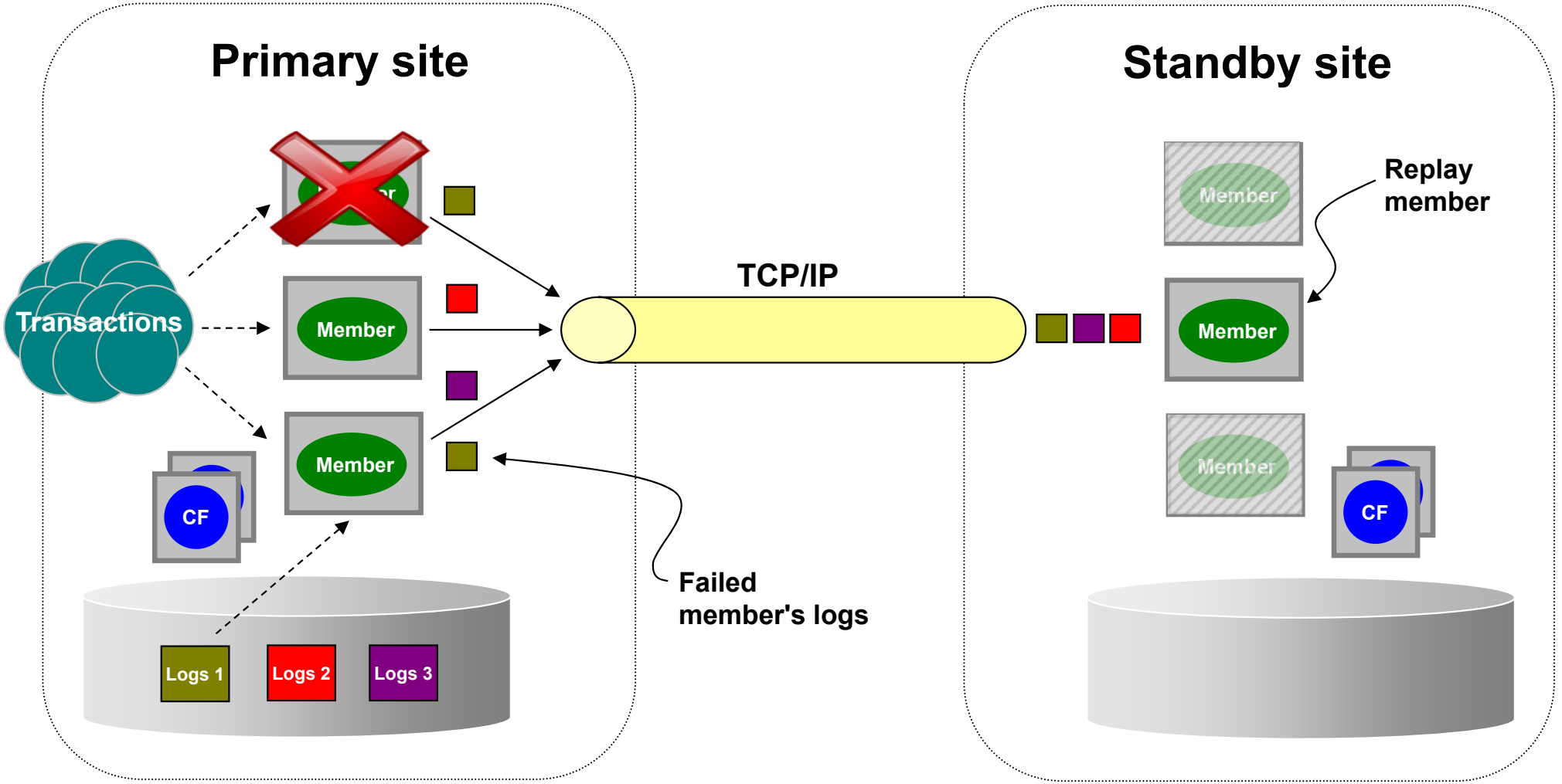
Old

Logs

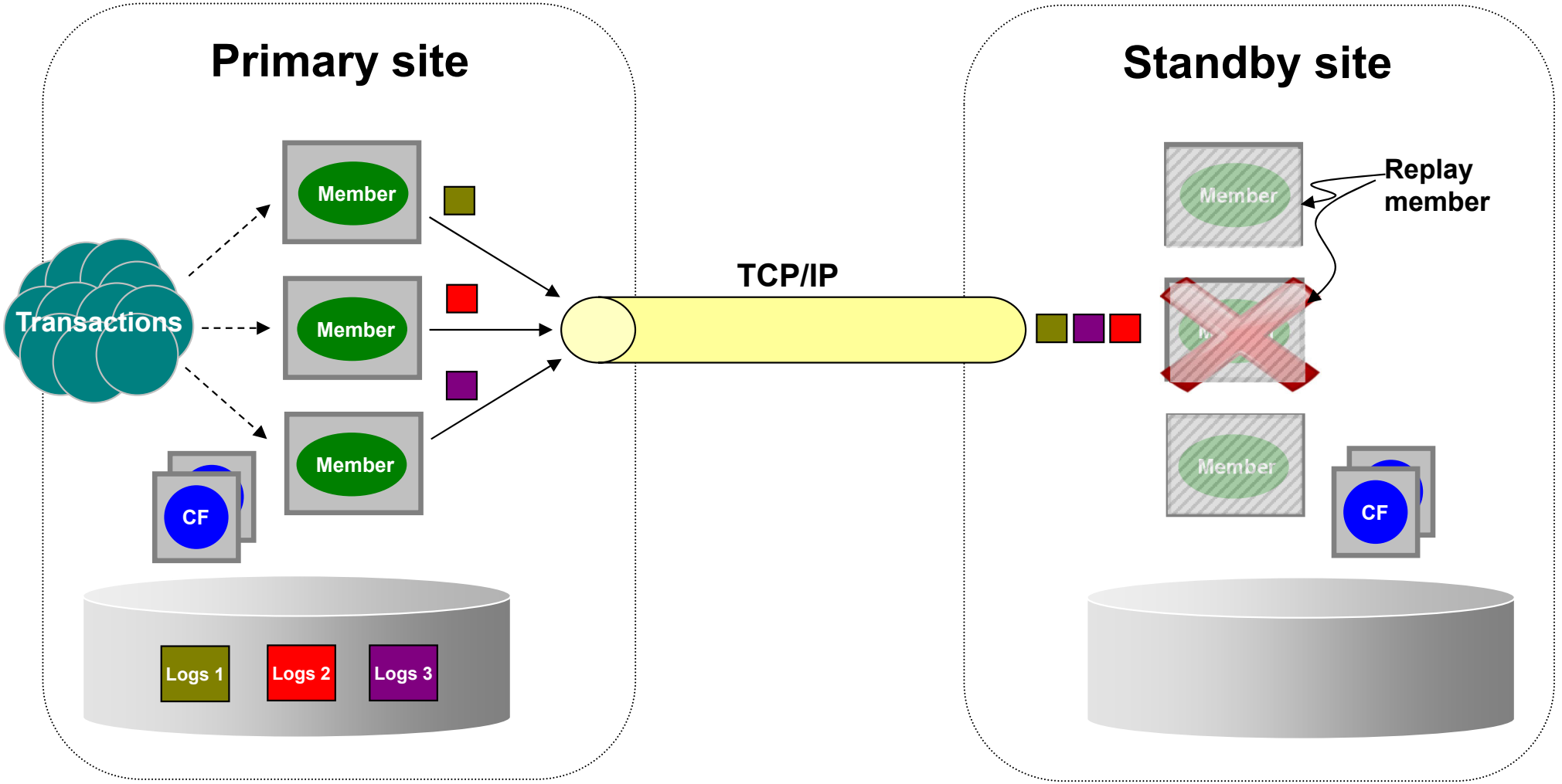
Tables

Indexes

HADR in DB2 pureScale: Example



HADR in DB2 pureScale: Example

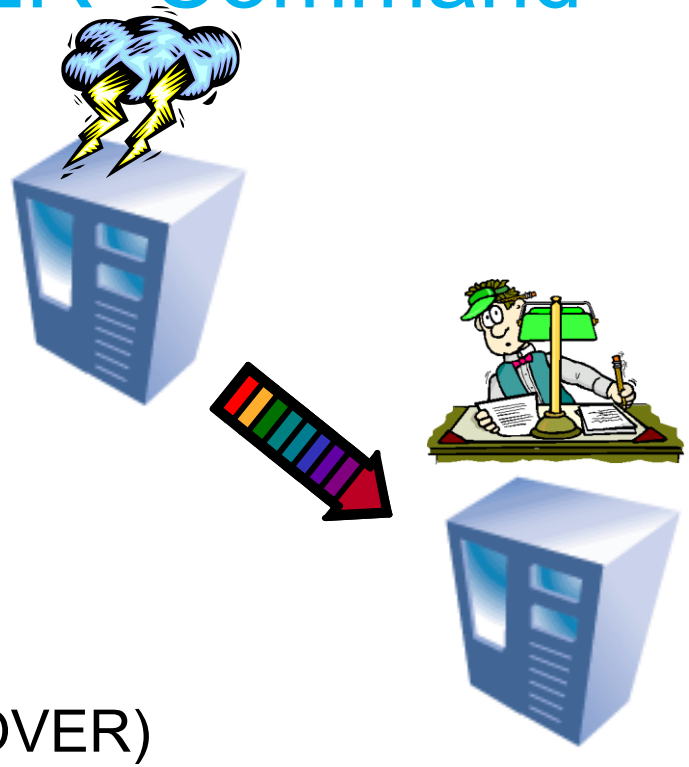


What's replicated, what's not?

- Logged operations are replicated
  - Example: DDL, DML, table space and buffer pool creation/deletion.
- Not logged operations are not replicated.
  - Example: database configuration parameter. not logged initially table, UDF libraries.
- Index pages are not replicated unless LOGINDEXBUILD is enabled
  - Ensure logsecond is increased as index rebuild is a single transaction. 4096 logs of 16 GB max
- How do I prevent non-logged operations?
  - Enable BLOCKNONLOGGED db cfg parameter

Failing Over : Simple “TAKEOVER” Command

- Normal TAKEOVER
  - Primary and standby switch roles as follows:
    1. Standby tells primary that it is taking over.
    2. Primary forces off all client connections and refuses new connections.
    3. Primary rolls back any open transactions and ships remaining log, up to the end of log, to standby.
    4. Standby replays received log, up to end of the log.
    5. Primary becomes new standby.
    6. Standby becomes new primary



- Emergency TAKEOVER (aka ‘Forced’ TAKEOVER)
  - The standby sends a notice asking the primary to shut itself down.
  - The standby does NOT wait for any acknowledgement from the primary to confirm that it has received the takeover notification or that it has shut down
  - The standby stops receiving logs from the primary, finishes replaying the logs it has already received, and then becomes a primary.

```
TAKEOVER HADR ON DATABASE <dbname>
        <USER <username> [USING <password>]]   [BY FORCE] [PEER WINDOW ONLY]
```

Primary Reintegration

- After primary failure and takeover, allow old primary to reintegrate as a standby with the new primary (saves user from having to reinitialize standby from scratch)
- Differentiating feature for DB2 HADR – competitors do not support this
- Reintegration possible if old primary can be made consistent with new primary
- Some conditions to satisfy, e.g. old primary crashed in peer state and had no disk updates that were not logged on old standby; some other details.
- Successful reintegration is most likely in SYNC mode, least likely in ASYNC and SUPERASYNC modes
- Synchronization with tail of the log file

HADR Setup Fits on One Slide

Primary Setup

db2 backup db hadr_db to backup_dir

db2 update db cfg for hadr_db using
HADR_LOCAL_HOST host_a
HADR_REMOTE_HOST host_b
HADR_LOCAL_SVC svc_a
HADR_REMOTE_SVC svc_b
HADR_REMOTE_INST inst_b
HADR_TIMEOUT 120
HADR_SYNCMODE ASYNC

db2 start hadr on database hadr_db as primary

Standby Setup

db2 restore db hadr_db from backup_dir

db2 update db cfg for hadr_db using
HADR_LOCAL_HOST host_b
HADR_REMOTE_HOST host_a
HADR_LOCAL_SVC svc_b
HADR_REMOTE_SVC svc_a
HADR_REMOTE_INST inst_a
HADR_TIMEOUT 120
HADR_SYNCMODE ASYNC

db2 start hadr on database hadr_db as standby

HADR Timeout

- While connected, the Primary and Standby exchange heartbeat messages
- The user can configure a timeout value using the database configuration parameter HADR_TIMEOUT
- If no message is received for the duration of HADR_TIMEOUT seconds, the TCP connection will be closed
  - A standby will then attempt to re-establish the connection by sending a handshake message to the primary
  - A primary will send a redirection message to the standby to probe it to start the handshake protocol
- Heartbeat interval is the minimum of the following:
  - 1/4 of HADR_TIMEOUT
  - 1/4 of HADR_PEER_WINDOW
  - 30 seconds
  - Find the exact heartbeat interval using the monitor element HEARTBEAT_INTERVAL

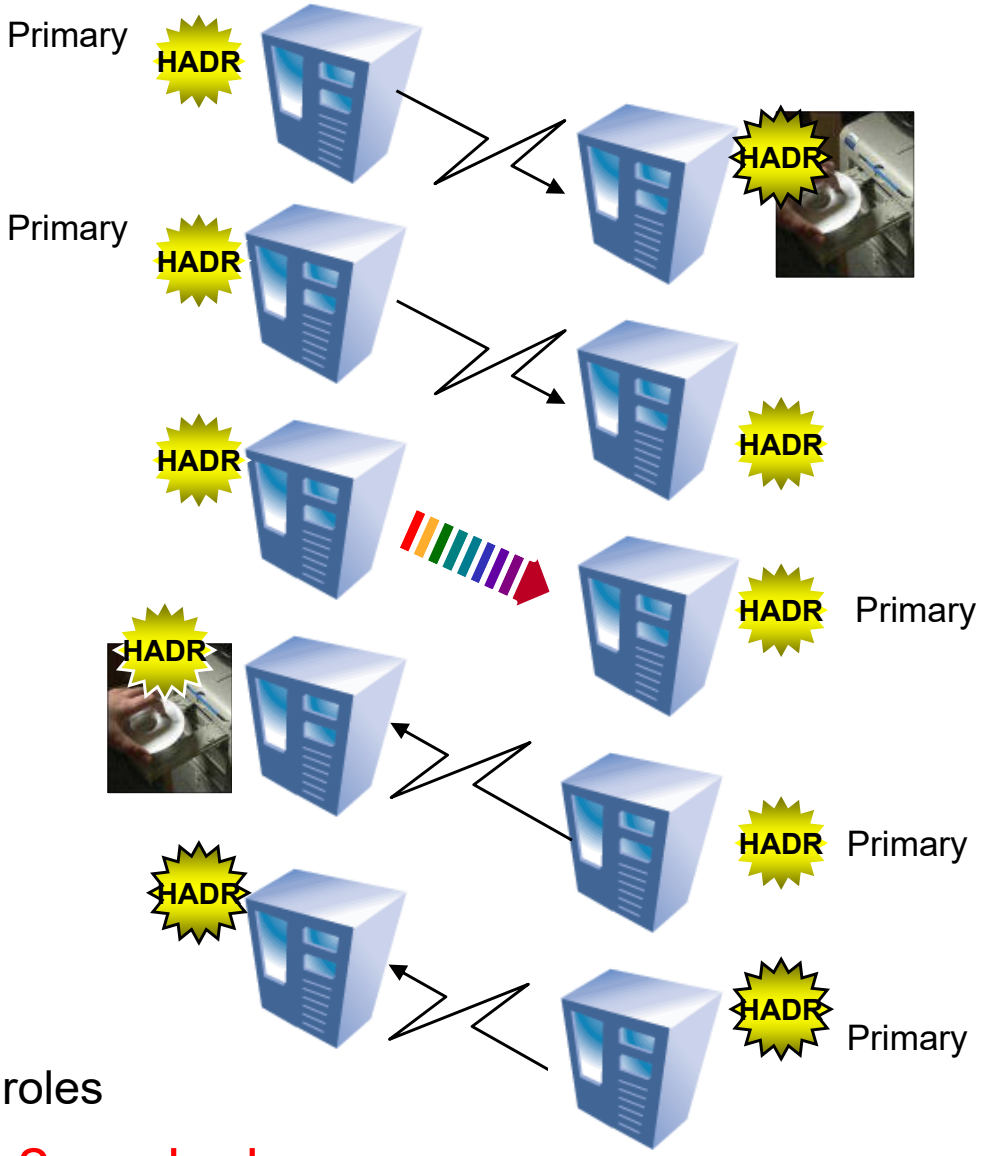
HADR Timeout

- If HADR_TIMEOUT is too long, it will slow detection of a lost connection or a failed standby
  - This may end up blocking transactions on primary
- It HADR_TIMEOUT is too short, HADR may get too many false alarms
  - Resulting in breaking the connection more often than necessary

- BEST PRACTICES:
  - The recommended HADR_TIMEOUT is at least 60 seconds
  - The default is 120 seconds
  - Some customers set HADR_TIMEOUT to very low values in order to avoid ever blocking the primary, at the cost of a disconnection every time the network hiccups

Software updates* on the fly

1. HADR in peer state
2. Deactivate HADR on the Standby
3. Upgrade the standby
4. Start the standby again
   - Let it catch-up with primary
5. Issue a normal TAKEOVER
   - The primary and standby change roles
6. Deactivate the new standby
7. Upgrade the new standby
8. Reactivate the new standby
   - Let it catch-up with primary
9. Optionally, TAKEOVER again
   - The primary and standby play their original roles



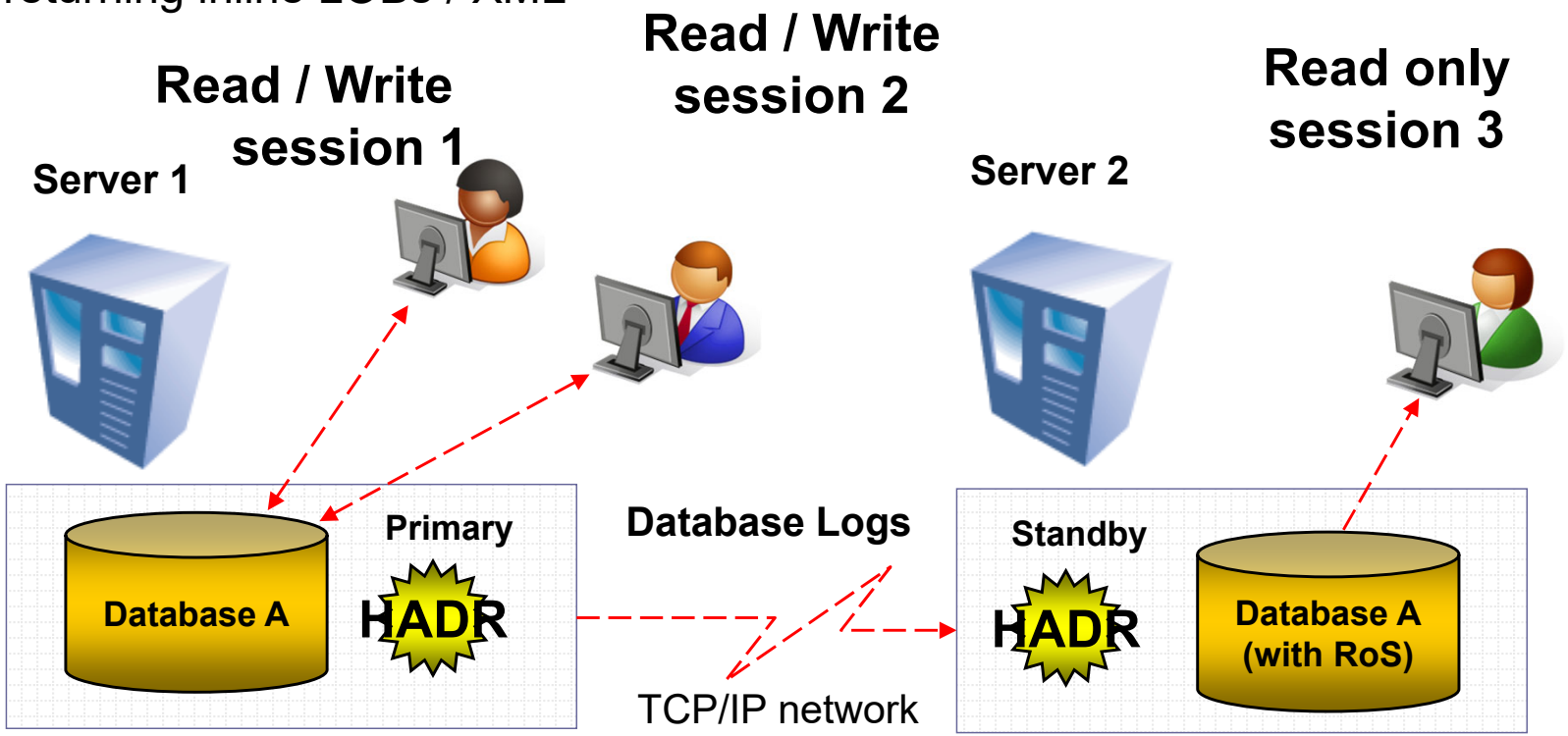
*os patch, h/w upgrades/replacement, Db2 modpak

Diagnostics

- db2diag.log is the most important diagnostic tool.
  - Look for hdrSetHdrState() trace points. All HADR state transition goes through this function.
  - You can also search for all messages produced by the HADR EDU.
  - If there are multiple HADR databases in the instances, be sure to distinguish messages from different databases.
  - What about the “This operation is not allowed on a standby database” message?
    - It indicates that a client (possibly an admin tool) is trying to connect to the standby database without enabling Reads On Standby.

HADR Read On Standby (RoS)

- Reads on Standby provides high availability, disaster recovery and allows reporting workloads.
- Improve resource utilization on your HA or DR hardware
- Offload reporting work from your primary, Increase capacity of HADR system
- Maximize Return on Investment and decrease Total Cost of Ownership
- supports returning inline LOBs / XML



HADR Reads on Standby
Overview

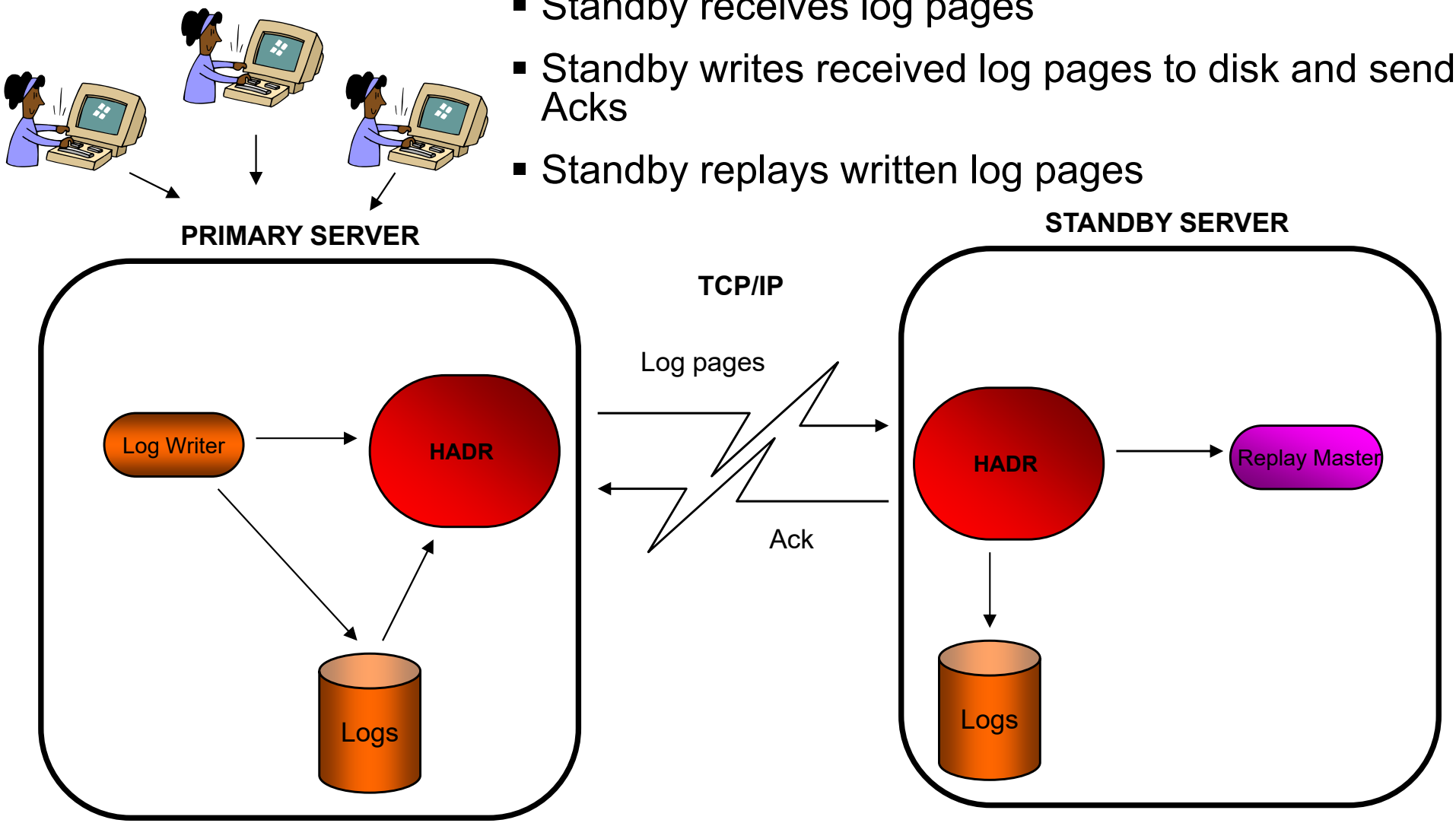
- RoS is enabled by registry variable – DB2_HADR_ROS
- Concurrent replay of logs and allow readers in all sync modes of HADR.
- Readers are allowed in all states of HADR except Local catchup.
- Support all type of complex read queries including:
  - joins
  - nested queries
  - index scans
  - cursors
- Support usage of internal temp tables for read queries.
- Auditing and security supported on Standby.
- Support WLM on Standby – New WLM definitions should be driven from Primary.

Agenda

- Comparison to other DB2 HA offerings
- HADR Review
- **HADR Sync modes**
- HADR Multiple Standby
- HADR Configuration
- HADR Monitoring
- Recent Improvements

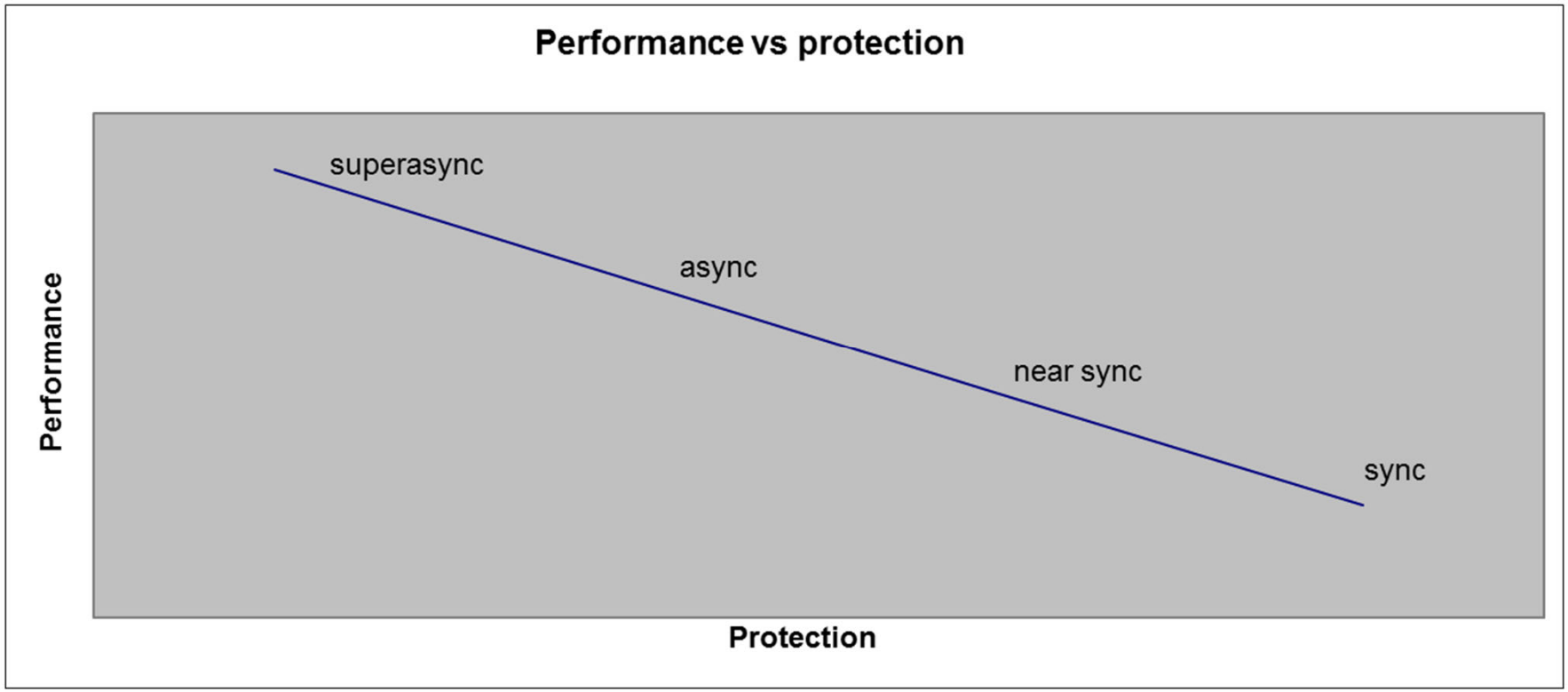
HADR replicate changes from the primary to the standby

- Transactions generate log records on the primary
- Primary sends log pages to the standby
- Standby receives log pages
- Standby writes received log pages to disk and sends Acks
- Standby replays written log pages



- Delay in the operations on the critical path can impact transactions on the primary

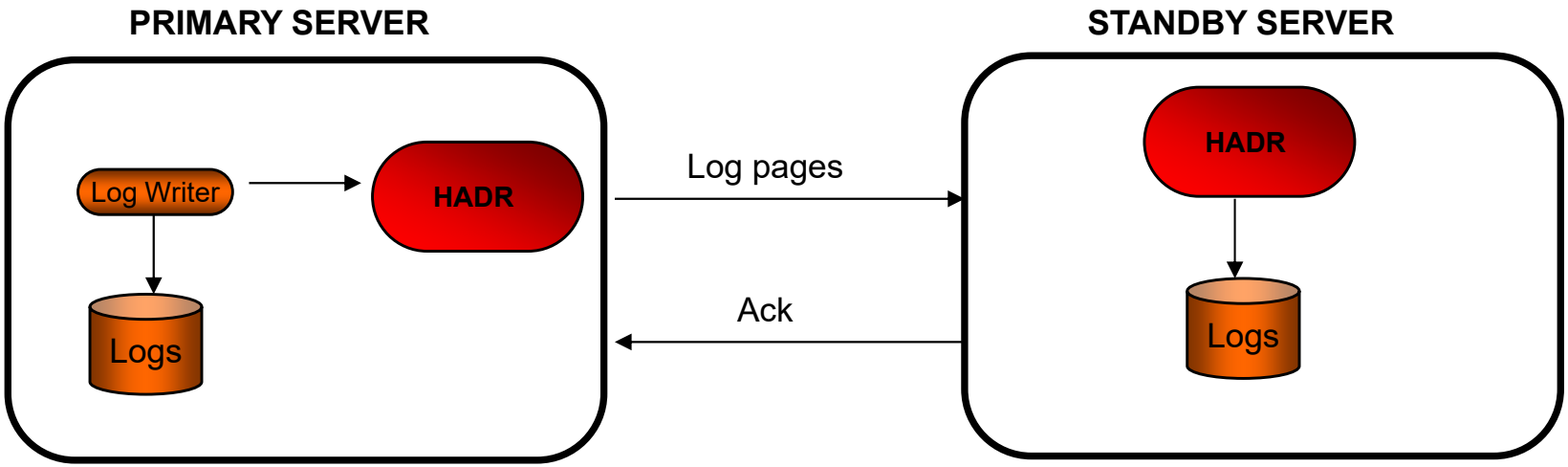
Performance Overhead varies with sync mode



- HADR overhead on primary database logging varies depending on the synchronization mode
  - Stronger sync mode provides more HA and DR protection
  - Weaker sync mode has less impact on the primary database

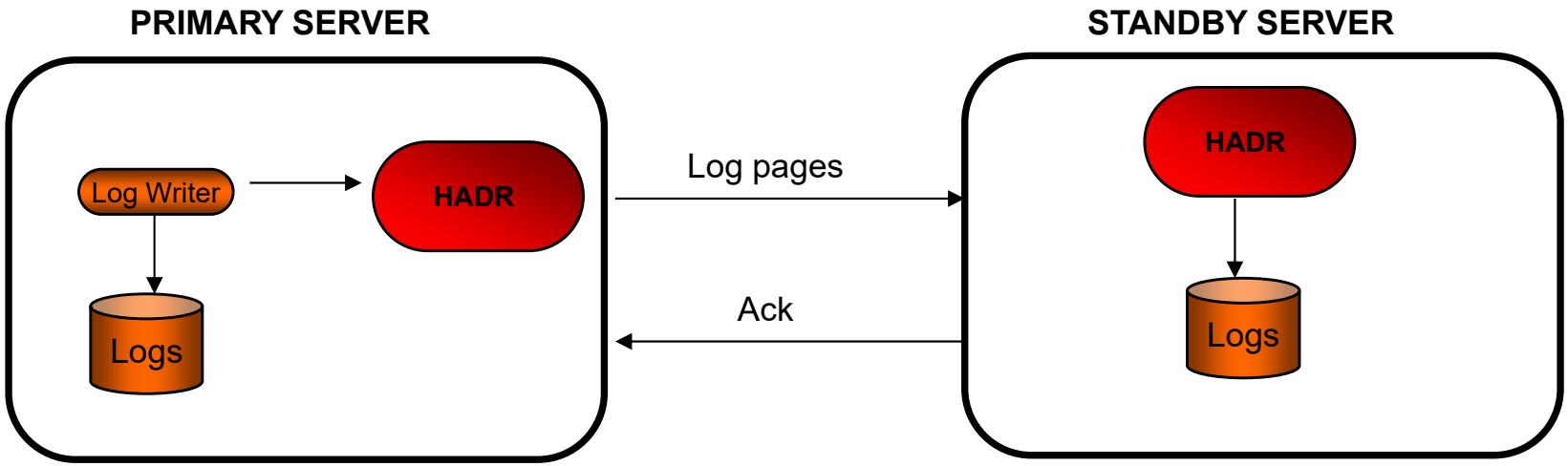
Synchronization Modes

- SYNC mode:
  - Logs are first written to the primary and are only then sent to standby (Serially)
  - Two on-disk copies of the log data are required for transaction commit
  - Total log write time = primary log write + log send + standby log write + ack message
  - Best data protection
  - But the cost of replication is higher than all other modes



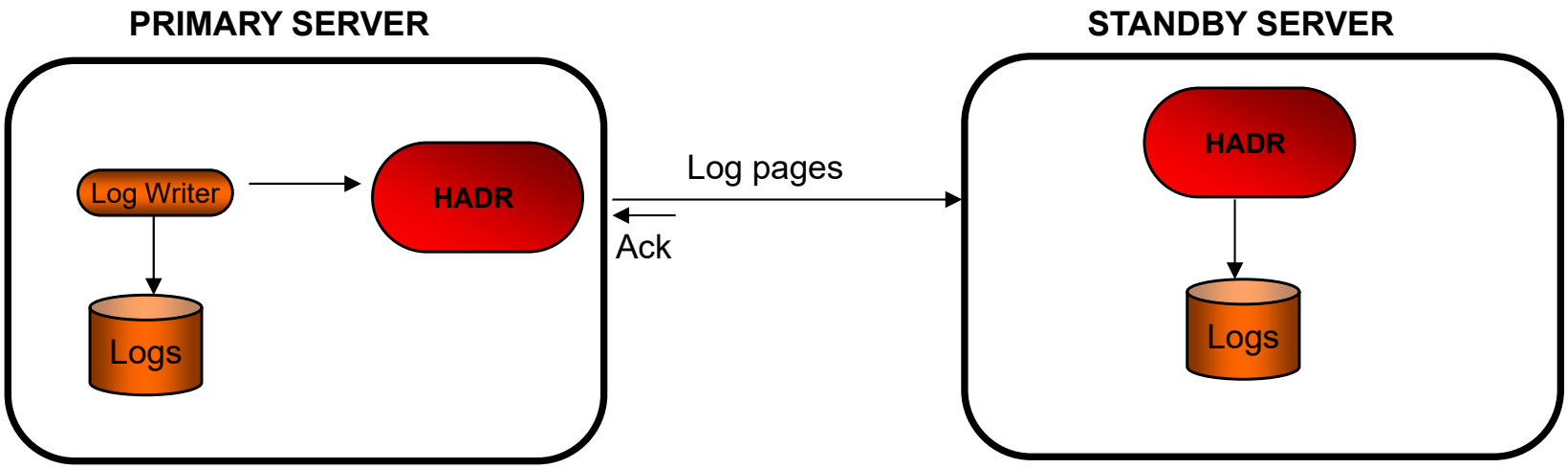
Synchronization Modes

- NEARSYNC mode:
  - Writing logs on the primary and sending logs to standby are done in parallel
  - Standby sends ack message as soon as it receives the logs
  - On a fast network, log replication results in little or no overhead to primary
  - Total log write time = Max (primary log write, log send + ack message)
  - Exposure to the relatively rare 'double failure' scenario
    - primary fails and the standby fails before it has a chance to write received logs to disk
  - Good choice for many applications
  - providing near synch protection at lower performance cost



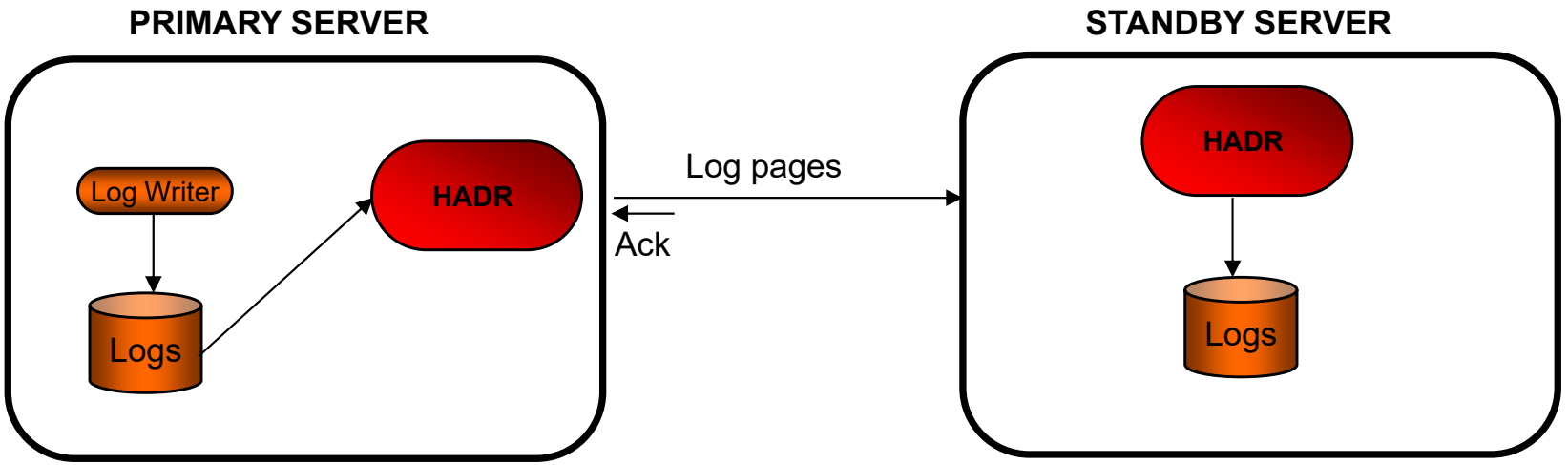
Synchronization Modes

- ASYNC mode
  - Writing logs on the primary and sending logs to standby are done in parallel
  - Does not wait for ack messages from the standby
    - Just the ack that the message has been sent
  - If the primary database fails, there is a higher chance that logs in transit are lost
  - Total log write time = Max (Primary log write rate, Submit log for sending)
  - Well suited for WAN application since network transmission delay does not impact performance



Synchronization Modes

- SUPERASYNC mode
  - Log writing and log shipping are completely independent
  - HADR remains in remote catchup state and never enters peer state
  - Zero impact on Log writing: Total log write time = Primary log write
  - But the log gap between the primary and the standby can grow
    - In a failover, data in the gap will be lost.
  - This mode has the least impact on primary, at the cost of the lowest data protection



Which Sync Mode to Use?

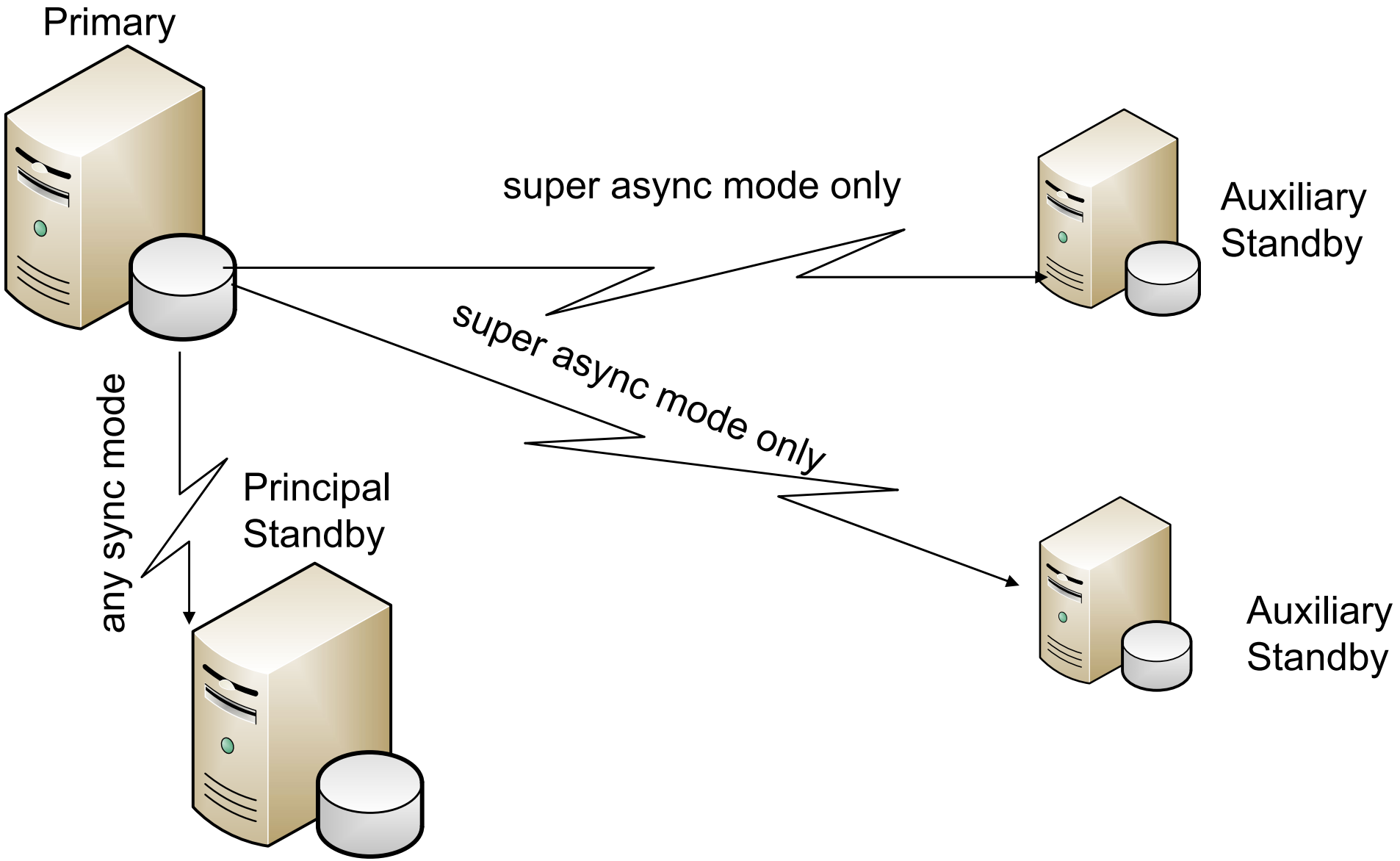
- Use SYNC or NEARSYNC mode:
  - Intended for HA
  - On a faster network
  - When you need the highest protection

- Use ASYNC or SUPERASYNC :
  - Intended for DR
  - On a slower network

Agenda

- Comparison to other DB2 HA offerings
- HADR Review
- HADR Sync modes
- **HADR Multiple Standby**
- HADR Configuration
- HADR Monitoring
- Recent Improvements

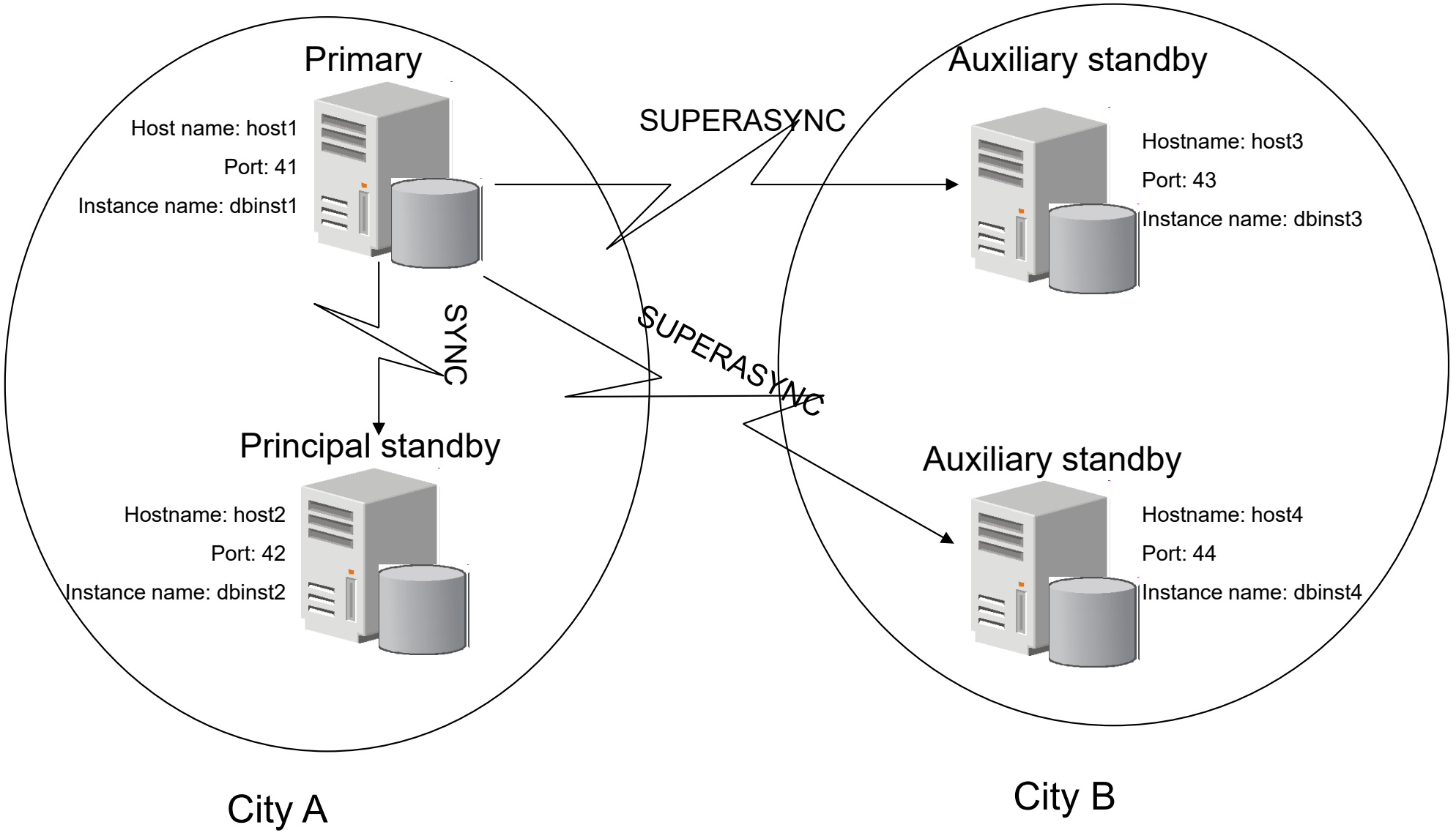
HADR Multiple Standby Overview



HADR Multiple Standby Features

- Principal Standby (PS) equivalent to standby today
  - PS supports any sync mode
  - Can automate takeover using integrated TSA
- Support for up to two(2) Auxiliary Standbys (AS)
  - AS supports super async mode only
  - No automated takeover supported
  - Always feed from the current primary
  - Can be added dynamically

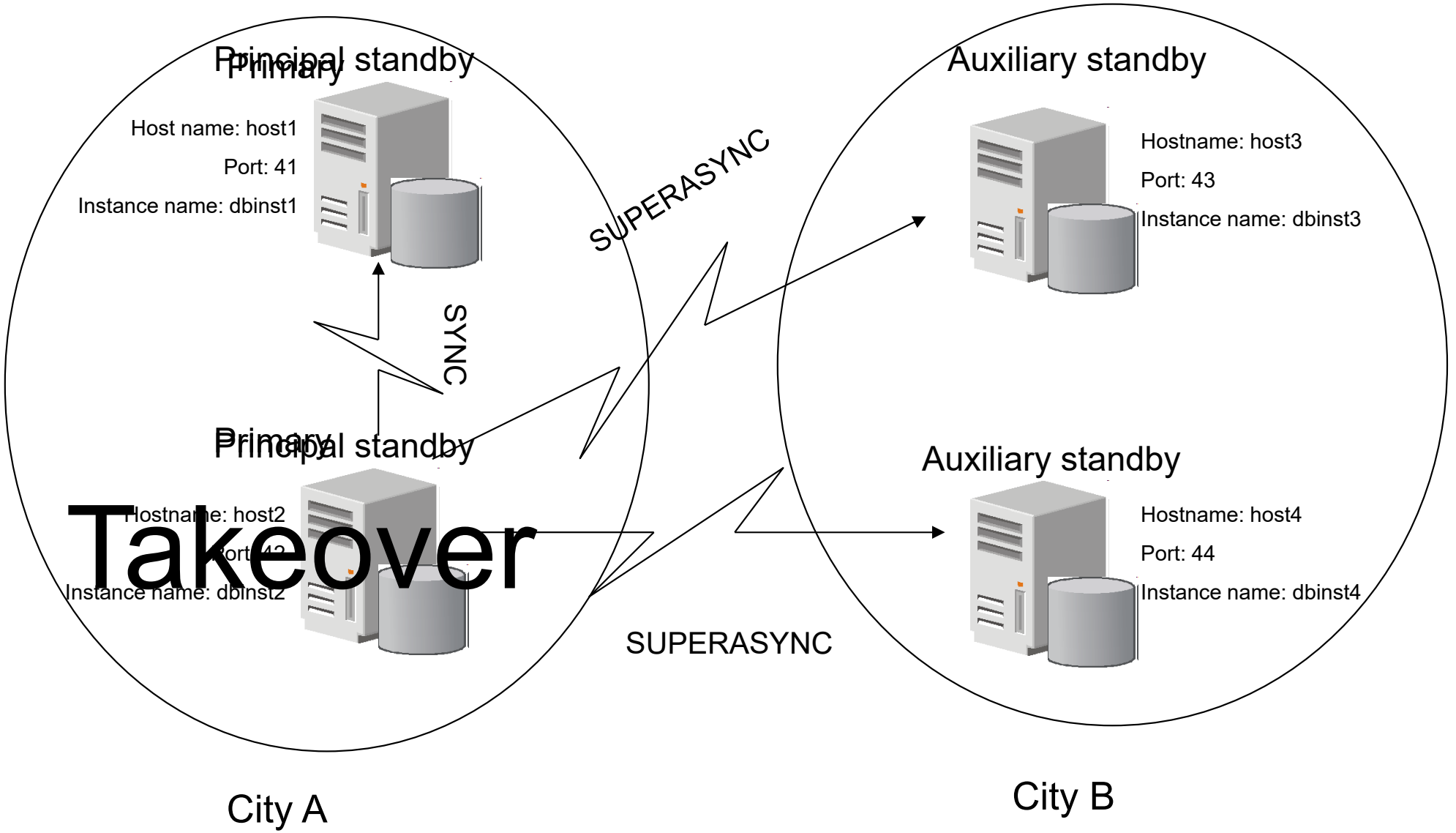
HADR Multiple Standby Example



Configuration values for each host

Configuration parameter	Host1	Host2	Host3	Host4
Hadr_target_list	host2:42 | host3:43 | host4:44	host1:42 | host3:43 | host4:44	Host2:42 | host1:42 | host4:44	host3:43 | host1:41 | host2:42
Hadr_remote_host	host2	host1	host1	host1
Hadr_remote_svc	42	41	41	41
Hadr_remote_inst	dbinst2	dbinst1	dbinst1	dbinst1
Hadr_local_host	host1	host2	host3	host4
Hadr_local_svc	41	42	43	44
Operational Hadr_syncmode	sync	sync	Near sync	Async
Effective Hadr_syncmode	N/A	sync	superasync	superasync

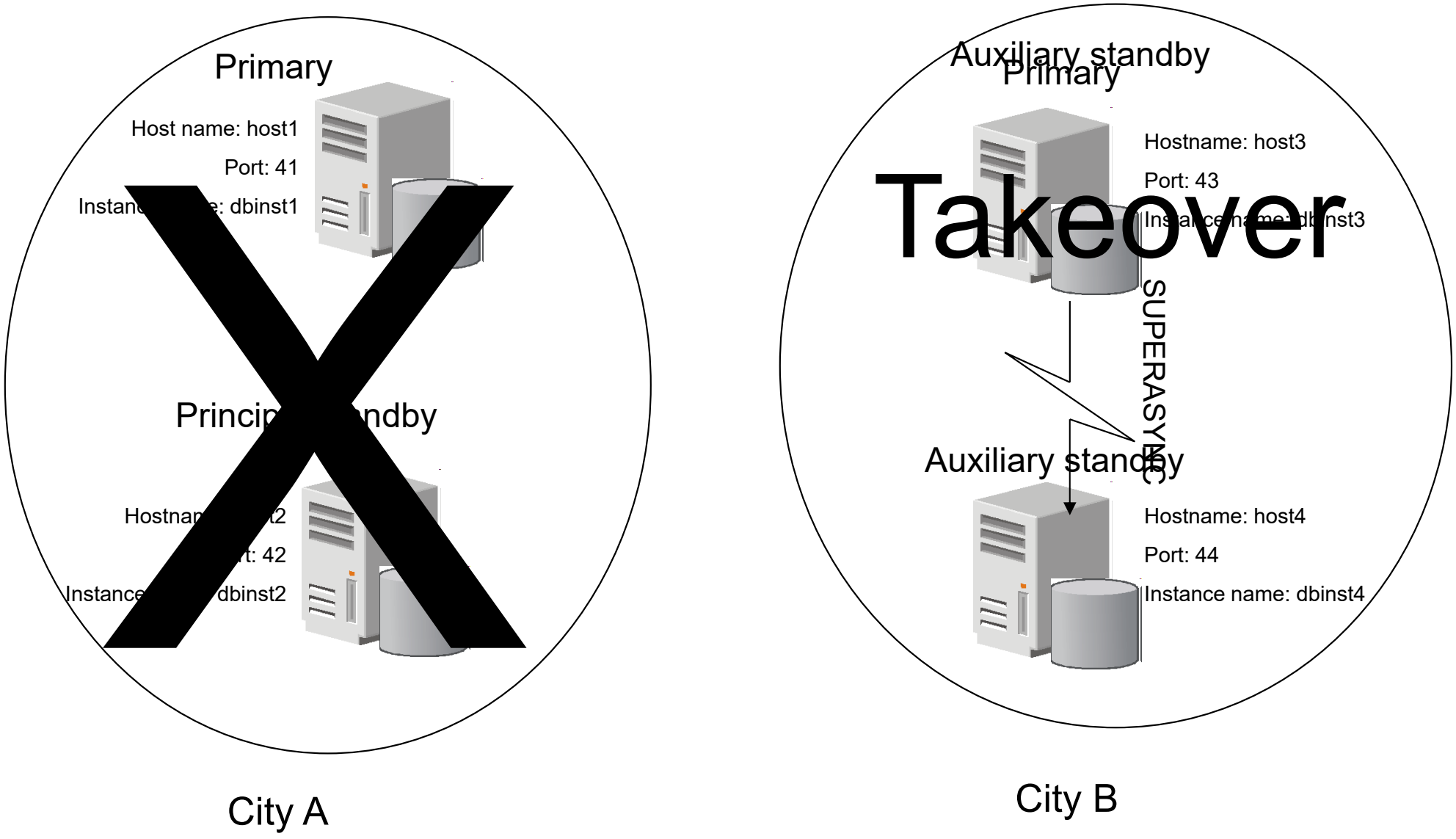
HADR Multiple Standby Role Reversal Example



After issuing takeover on host2 (auto reconfigured)

Configuration parameter	Host1	Host2	Host3	Host4
Hadr_target_list	host2:42 | host3:43 | host4:44	host1:41 | host3:43 | host4:44	Host2:42 | host1:41| host4:44	host3:43 | host1:41 | host2:42
Hadr_remote_host	host2	host1	host2	host2
Hadr_remote_svc	40	10	42	42
Hadr_remote_inst	dbinst2	dbinst1	dbinst2	dbinst2
Hadr_local_host	host1	host2	host3	host4
Hadr_local_svc	41	42	43	44
Operational Hadr_syncmode	sync	sync	Near sync	Async
Effective Hadr_syncmode	sync	N/A	superasync	superasync

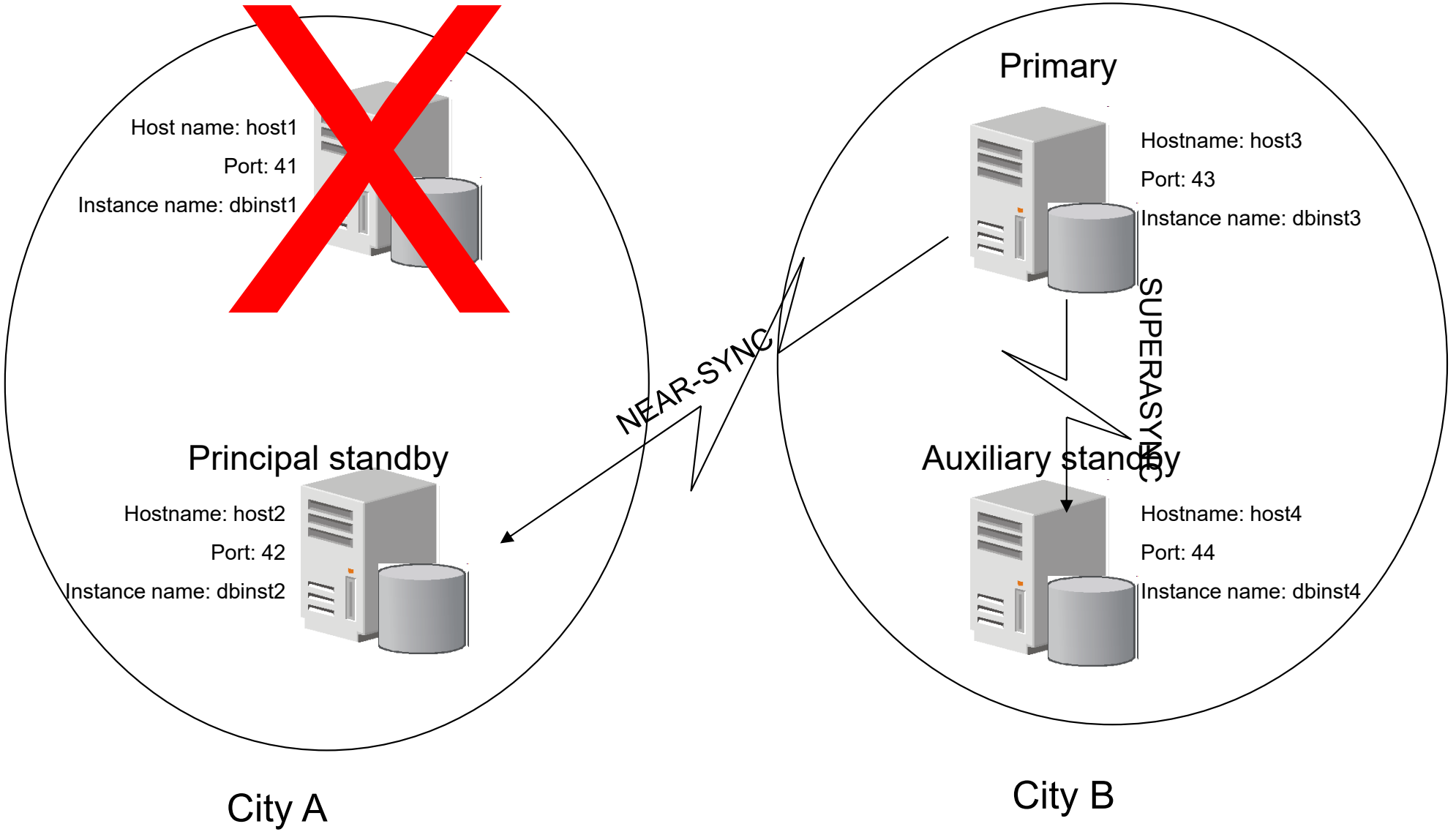
HADR Multiple Standby Forced Takeover Example



After issuing takeover on host3 (host 1+2 are down)

Configuration parameter	Host1	Host2	Host3	Host4
Hadr_target_list	hos hos hos	host1 host3 host4	host2:42 | host1:41 | host4:44	host3:43 | host1:41 | host2:42
Hadr_remote_host	hos	host1	host4	Host3
Hadr_remote_svc	40	10	42	43
Hadr_remote_inst	dbir	dbins	dbinst2	dbinst3
Hadr_local_host	hos	host2	host3	host4
Hadr_local_svc	41	42	43	44
Operational Hadr_syncmode	sync	sync	Near sync	Async
Effective Hadr_syncmode	N/A	sync	n/a	superasync

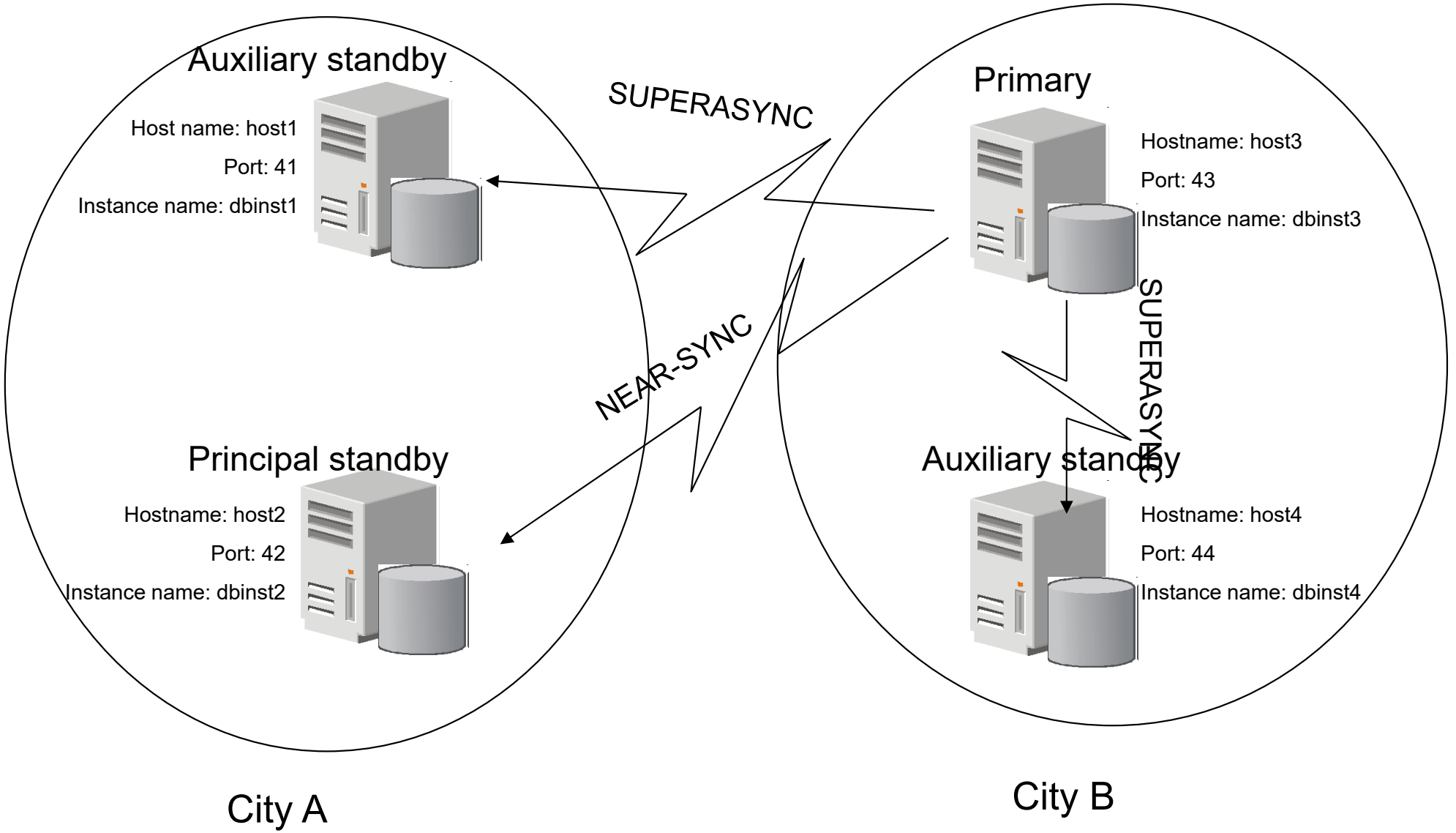
HADR Multiple Standby Example



After issuing takeover on host3 (host 2 online)

Configuration parameter	Host1	Host2	Host3	Host4
Hadr_target_list	host2 host3 host4	host1:41 | host3:43 | host4:44	host2:42 | host1:41 | host4:44	host3:43 | host1:41 | host2:42
Hadr_remote_host	host2	host3	host2	host3
Hadr_remote_svc	40	43	42	43
Hadr_remote_inst	dbinst	dbinst3	dbinst2	dbinst3
Hadr_local_host	host1	host2	host3	host4
Hadr_local_svc	41	42	43	44
Operational Hadr_syncmode	sync	sync	Near sync	Async
Effective Hadr_syncmode	N/A	Nearsync	n/a	superasync

HADR Multiple Standby Example

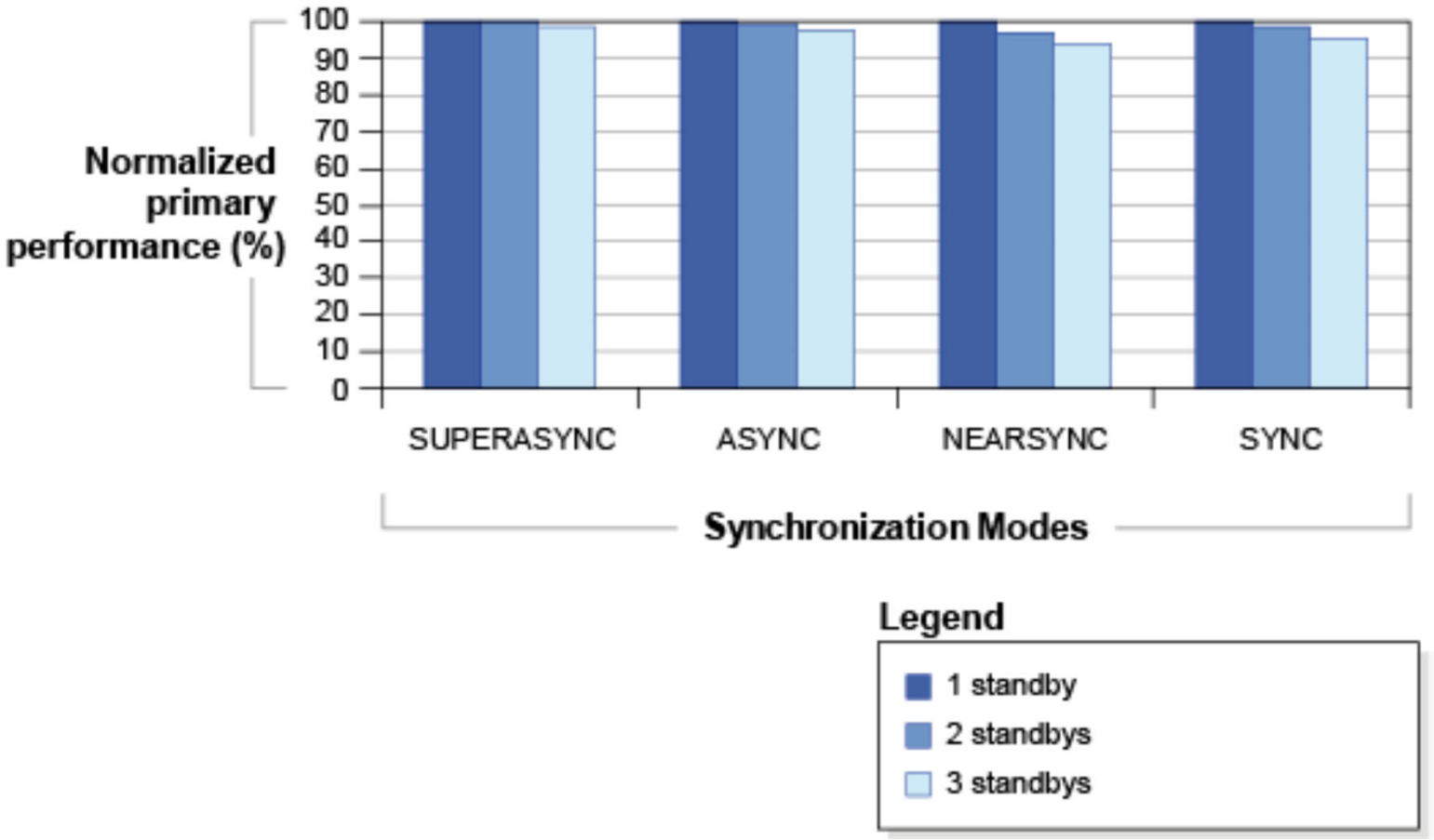


After issuing takeover on host3 (host 1 online)

Configuration parameter	Host1	Host2	Host3	Host4
Hadr_target_list	host2:42 | host3:43 | host4:44	host1:41 | host3:43 | host4:44	host2:42 | host1:41 | host4:44	host3:43 | host1:41 | host2:42
Hadr_remote_host	host3	host3	Host2	host3
Hadr_remote_svc	43	43	42	43
Hadr_remote_inst	dbinst3	dbinst3	dbinst2	dbinst3
Hadr_local_host	host1	host2	Host3	host4
Hadr_local_svc	41	42	43	44
Operational Hadr_syncmode	sync	sync	Near sync	Async
Effective Hadr_syncmode	superasync	Nearsync	n/a	superasync

Multiple standby Performance impact

- The performance impact on the primary of adding additional standby targets is approximately 3%



Accelerate a standby catchup phase

- Use the most recent backup from the production server
- Provide access to archived logs
- When starting a standby will look at
  - Active log path
  - Mirrored log path
  - Overflow log path
  - Logarchmeth1
  - Logarchmeth2
  - Request log data from the primary

Agenda

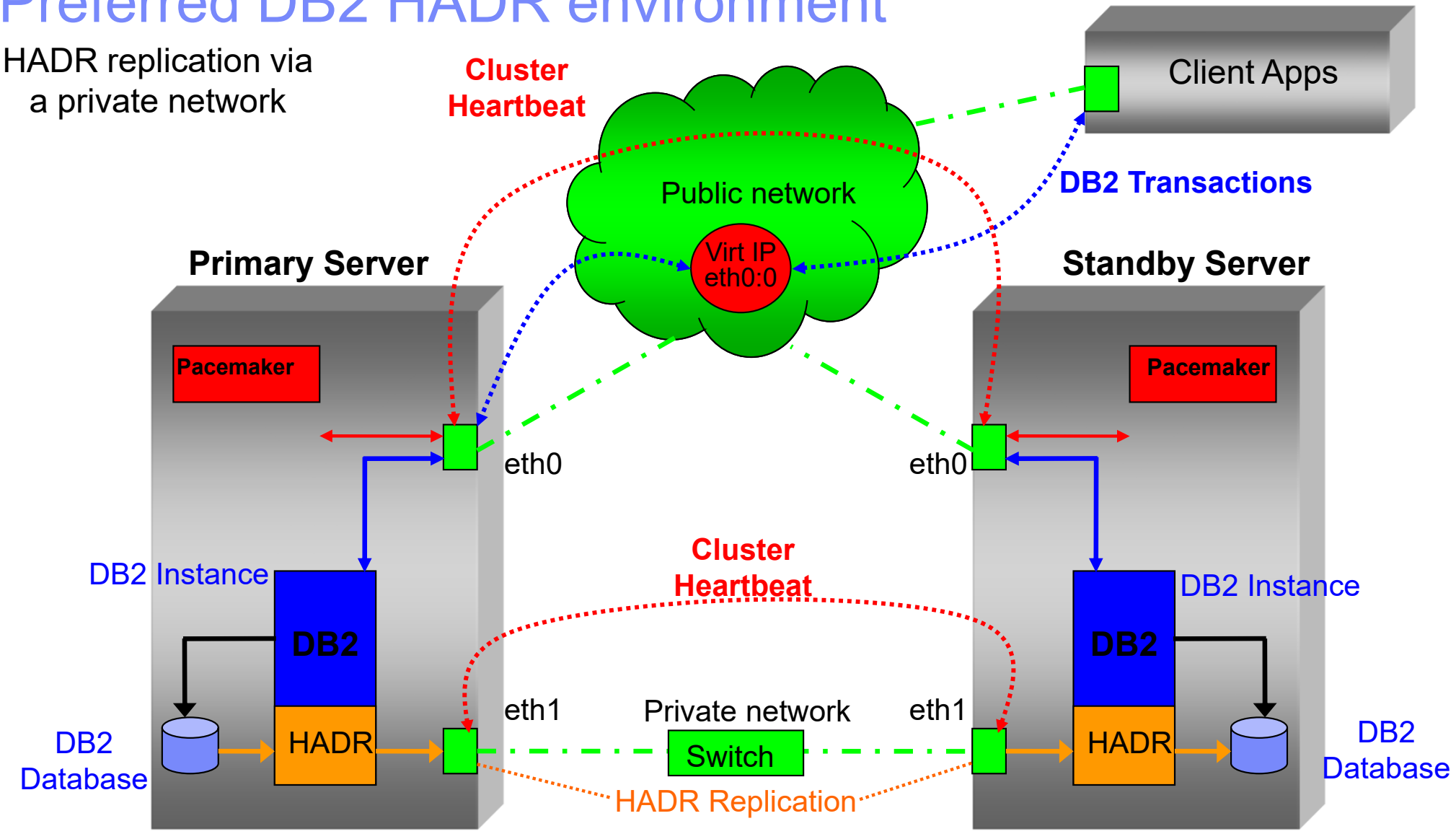
- Comparison to other DB2 HA offerings
- HADR Review
- HADR Sync modes
- HADR Multiple Standby
- **HADR Configuration**
- HADR Monitoring
- Recent Improvements

HADR Configuration Parameters Updates

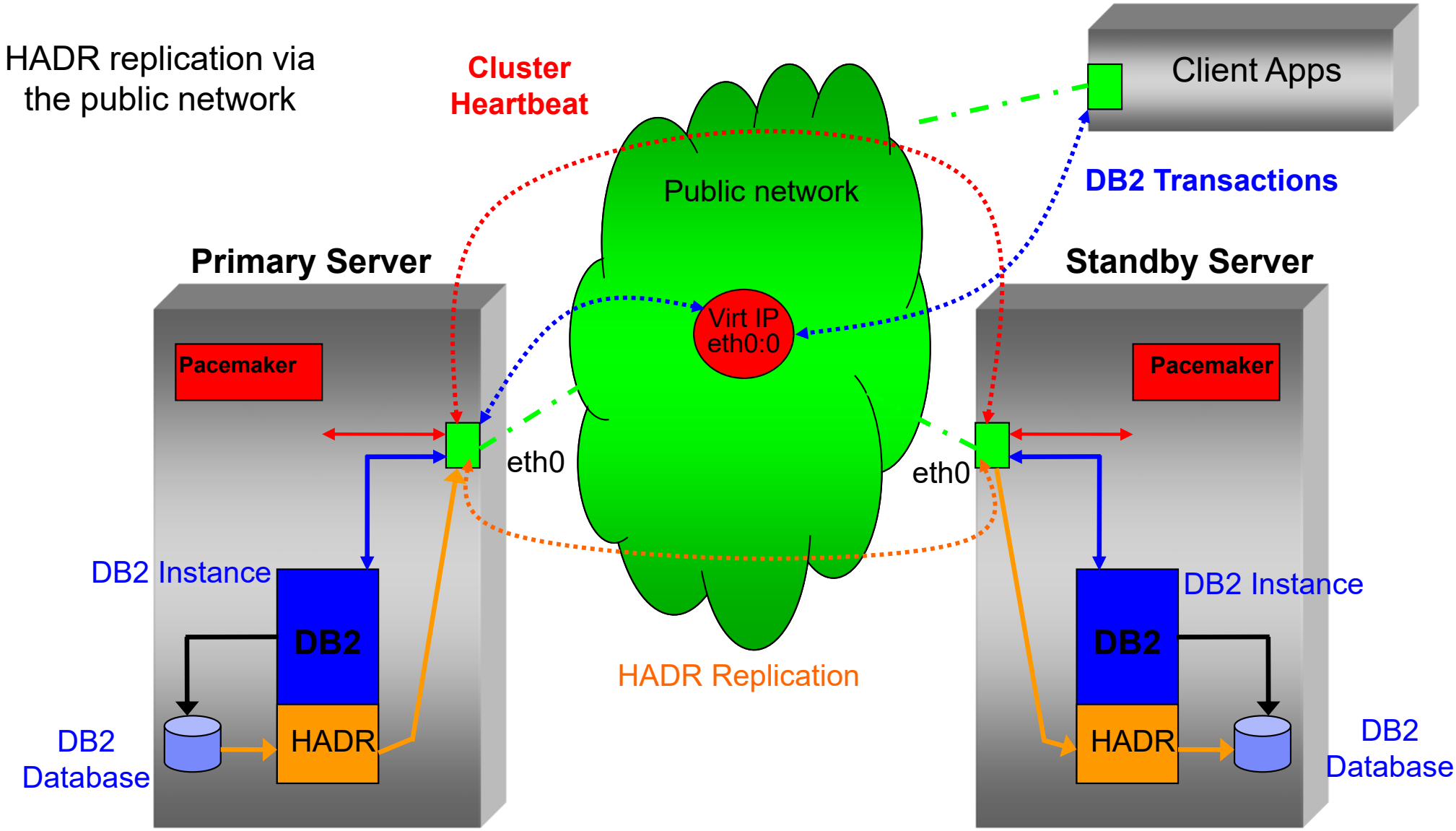
- you need only stop and start HADR for updates to some HADR configuration parameters for the primary database to take effect. You do not have to deactivate and reactivate the database. This dynamic capability affects only the primary database because stopping HADR deactivates any standby database.
- The affected configuration parameters are as follows:
  - **hadr_local_host**
  - **hadr_local_svc**
  - **hadr_peer_window**
  - **hadr_remote_host**
  - **hadr_remote_inst**
  - **hadr_remote_svc**
  - **hadr_replay_delay**
  - **hadr_ssl_host_val**
  - **hadr_ssl_label**
  - **hadr_spool_limit**
  - **hadr_syncmode**
  - **hadr_target_list**
  - **hadr_timeout**

Preferred DB2 HADR environment

HADR replication via
a private network



Alternate example of a DB2 HADR environment

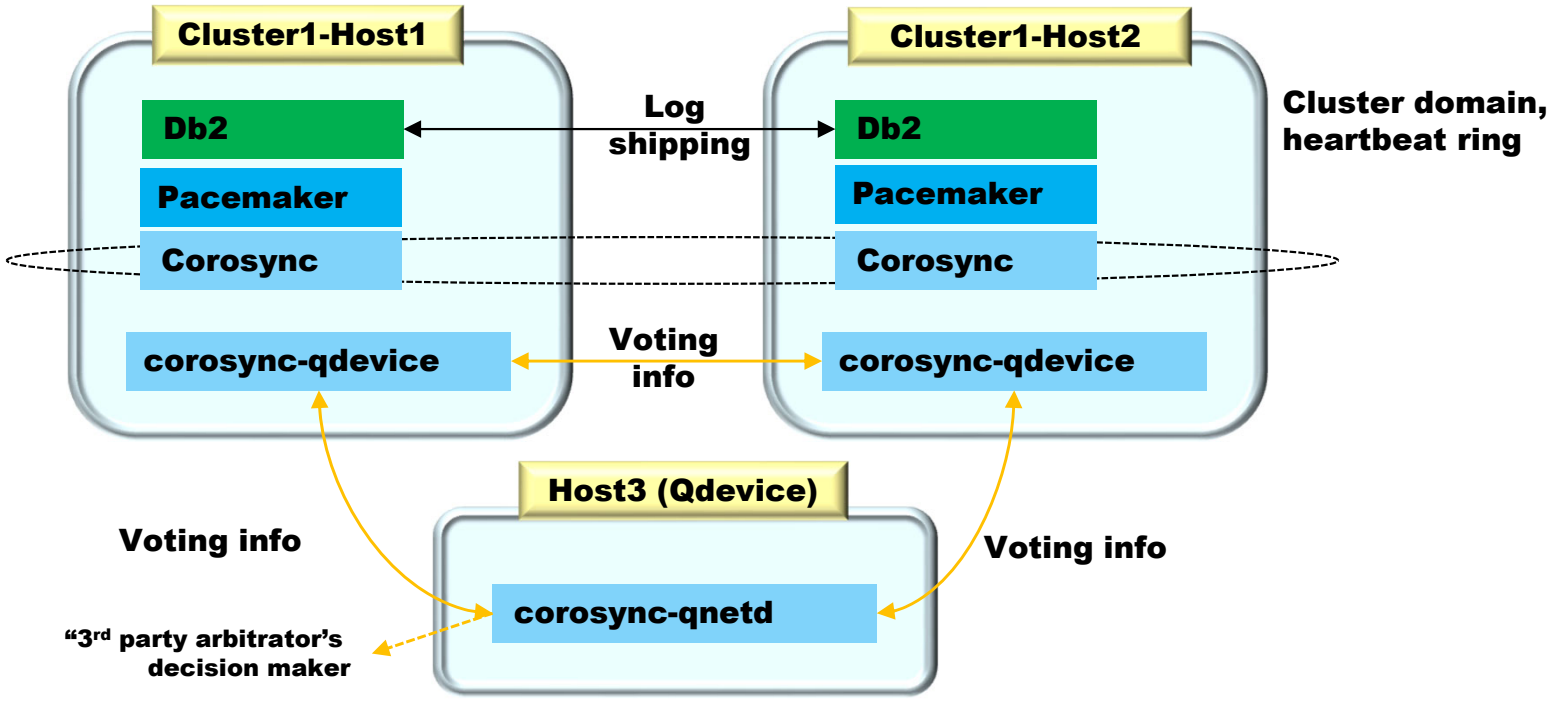


HADR Pacemaker Documentation

- Extensive changes to Db2 Knowledge Center to document how to setup and configure pacemaker to enable HADR Automation
- https://www.ibm.com/docs/en/db2/12.1.x?topic=db2cm-hadr-pacemaker-option

- Configuring HADR Pacemaker in IBM Cloud VPC
  - https://www.ibm.com/docs/en/db2/12.1.x?topic=pacemaker-vpc
- Configuring HADR Pacemaker in AWS
  - https://www.ibm.com/docs/en/db2/12.1.x?topic=pacemaker-aws
- Configuring HADR Pacemaker in Azure
  - https://www.ibm.com/docs/en/db2/12.1.x?topic=azure-hadr-configurations
- Configuring HADR Pacemaker in GCP
  - https://www.ibm.com/docs/en/db2/12.1.x?topic=pacemaker-two-host-hadr-google-cloud

Q Device – NOT A SPOF
Best practice “2+1” node configuration



Host Failure	Host1 (ID:1)	Host2 (ID:1)	Host3 (Qdevice)	Cluster Quorum State	Scenario
1	Offline	Online	Online	Has Quorum	Host 1 fails, quorum is maintained between host 2 and host 3. Any HADR Primary or MF primary fails over to host 2 and the cluster continues to operate.
2	Online	Offline	Online	Has Quorum	Host 2 fails, quorum is maintained between host 1 and host 3. Any HADR Primary or MF primary fails over to host 1 and the cluster continues to operate.
3	Online	Online	Offline	Has Quorum	Host 3 fails, quorum is maintained between host 1 and host 2. There is no impact to Db2.
1&2	Offline	Offline	Online	No Quorum	Host 1 & 2 fail. The Pacemaker cluster is down, there is nothing Host 3 can do.
1&3	Offline	Online	Offline	No Quorum	Host 1 and Host 3 fail. Host 2 must shutdown Db2 as it has lost quorum.
2&3	Online	Offline	Offline	No Quorum	Host 2 and Host 3 fail. Host 1 must shutdown Db2 as it has lost quorum.

The first 3 scenarios are single host failures.

The last 3 scenarios are double host failures where Db2 will go offline.

- Should be very rare assuming customers have built their clusters for high availability correctly.
- i.e., No shared hardware, network switch, power source, etc which can become a SPOF.

Agenda

- Comparison to other DB2 HA offerings
- HADR Review
- HADR Sync modes
- HADR Multiple Standby
- HADR Configuration
- **HADR Monitoring**
- Recent Improvements

Monitoring HADR Performance

- Two interfaces:
  - MON_GET_HADR table function
    - Available on the primary
    - Available on the standby when reads on standby is enabled
  - db2pd –hadr
    - Available on both the primary and the standby
- The older snapshot interface has been deprecated
- Information for a remote database can be slightly out of date
  - For up to date information, query the source database directly

Table Function Example

Example: 3 member clusters; Primary member 0 is assisting member 1; Standby member 0 is replay member

```
select LOG_STREAM_ID, PRIMARY_MEMBER, STANDBY_MEMBER, HADR_STATE, HADR_FLAGS
from table (mon_get_hadr(0))
```

```
LOG_STREAM_ID PRIMARY_MEMBER STANDBY_MEMBER HADR_STATE           HADR_FLAGS
------------- -------------- -------------- ------------------- ----------
0             0              0              PEER
1             0              0              REMOTE_CATCHUP      ASSISTED_REMOTE_CATCHUP
```

```
select LOG_STREAM_ID, PRIMARY_MEMBER, STANDBY_MEMBER, HADR_STATE, HADR_FLAGS
from table (mon_get_hadr(1))
```

```
LOG_STREAM_ID PRIMARY_MEMBER STANDBY_MEMBER HADR_STATE           HADR_FLAGS
------------- -------------- -------------- ------------------- ----------
1             1              0              DISCONNECTED
```

```
select LOG_STREAM_ID, PRIMARY_MEMBER, STANDBY_MEMBER, HADR_STATE, HADR_FLAGS
from table (mon_get_hadr(2))
```

```
LOG_STREAM_ID PRIMARY_MEMBER STANDBY_MEMBER HADR_STATE           HADR_FLAGS
------------- -------------- -------------- ------------------- ----------
2             2              0              PEER
```

```
select LOG_STREAM_ID, PRIMARY_MEMBER, STANDBY_MEMBER, HADR_STATE, HADR_FLAGS
from table (mon_get_hadr(-2))
```

```
LOG_STREAM_ID PRIMARY_MEMBER STANDBY_MEMBER HADR_STATE           HADR_FLAGS
------------- -------------- -------------- ------------------- ----------
0             0              0              PEER
1             0              0              REMOTE_CATCHUP      ASSISTED_REMOTE_CATCHUP
2             2              0              PEER
```

ration

```
Database Member 0 -- Database SAMPLE -- Active -- Up 0 days 00:03:40 -- Date 2026-05-21-14.08.50.802167

                           HADR_ROLE = PRIMARY
                       REPLAY_TYPE = PHYSICAL
                     HADR_SYNCMODE = NEARSYNC
                        STANDBY_ID = 1
                     LOG_STREAM_ID = 0
                        HADR_STATE = PEER
                        HADR_FLAGS = TCP_PROTOCOL
                PRIMARY_MEMBER_HOST = db2hadrnode1-priv
                    PRIMARY_INSTANCE = db2inst1
                      PRIMARY_MEMBER = 0
                STANDBY_MEMBER_HOST = db2hadrnode2-priv
                    STANDBY_INSTANCE = db2inst1
                      STANDBY_MEMBER = 0
                HADR_CONNECT_STATUS = CONNECTED
           HADR_CONNECT_STATUS_TIME = 05/21/2026 14:05:38.246083 (1779390338)
         HEARTBEAT_INTERVAL(seconds) = 5
                    HEARTBEAT_MISSED = 11
                  HEARTBEAT_EXPECTED = 38
               HADR_TIMEOUT(seconds) = 120
       TIME_SINCE_LAST_RECV(seconds) = 2
            PEER_WAIT_LIMIT(seconds) = 0
          LOG_HADR_WAIT_CUR(seconds) = 0.000
    LOG_HADR_WAIT_RECENT_AVG(seconds) = 0.000000
   LOG_HADR_WAIT_ACCUMULATED(seconds) = 0.000
                  LOG_HADR_WAIT_COUNT = 0
SOCK_SEND_BUF_REQUESTED,ACTUAL(bytes) = 0, 87040
SOCK_RECV_BUF_REQUESTED,ACTUAL(bytes) = 0, 131072
           PRIMARY_LOG_FILE,PAGE,POS = S0000021.LOG, 862, 150250396
           STANDBY_LOG_FILE,PAGE,POS = S0000021.LOG, 862, 150250396
                 HADR_LOG_GAP(bytes) = 0
     STANDBY_REPLAY_LOG_FILE,PAGE,POS = S0000021.LOG, 862, 150250396
       STANDBY_RECV_REPLAY_GAP(bytes) = 0
                  PRIMARY_LOG_TIME = 05/07/2026 12:23:13.000000 (1778174593)
                  STANDBY_LOG_TIME = 05/07/2026 12:23:13.000000 (1778174593)
            STANDBY_REPLAY_LOG_TIME = 05/07/2026 12:23:13.000000 (1778174593)
         STANDBY_RECV_BUF_SIZE(pages) = 512
             STANDBY_RECV_BUF_PERCENT = 0
           STANDBY_SPOOL_LIMIT(pages) = 13000
                STANDBY_SPOOL_PERCENT = 0
                   STANDBY_ERROR_TIME = NULL
                PEER_WINDOW(seconds) = 120
                     PEER_WINDOW_END = 05/21/2026 14:10:47.000000 (1779390647)
            READS_ON_STANDBY_ENABLED = N
             HADR_LAST_TAKEOVER_TIME = 06/02/2025 05:59:50.000000 (1748861990)
```

Worth Paying Attention To

HADR_STATE

- Verify the primary and standby are in connected and in PEER mode (or REMOTE_CATCHUP for superasync sync mode)

HADR_CONNECT_STATUS

- SQLM_HADR_CONN_CONNECTED - the good
- SQLM_HADR_CONN_DISCONNECTED - the bad
- SQLM_HADR_CONN_CONGESTED - and the ugly
- If you see congestion, either there are network issues or the standby is not keeping up.
- Use Spooling if the standby is not keeping up

HADR_LOG_GAP

- Use this element to determine the gap between the primary and standby HADR database logs
- It should be kept to a minimum
- An increasing number indicates a standby that is not keeping up

HEARTBEAT_EXPECTED vs HEARTBEAT_MISSED

- Number of heartbeat messages expected vs those that were not received, accumulated since database startup on the local member

Worth Paying Attention To

PRIMARY_LOG_FILE, PAGE. POS vs STANDBY_LOG_FILE, PAGE. POS

- The name of the current log file on this log stream on the primary/standby HADR database.
- Increasing difference between the two above reflects either a network or I/O bottleneck

STANDBY_REPLAY_LOG_FILE,PAGE,POS

- standby replay log position. The page number is relative to the log file.

LOG_HADR_WAIT_CUR

- Provides current logger waiting time on an HADR log shipping request
- Helps identify a primary that is being blocked by log shipping to the standby
- Should never reach more than 5 seconds
- Can serve as an early indication of network or standby problems
- You can cap this using the registry variable DB2_HADR_PEER_WAIT_LIMIT

Worth Paying Attention To

HADR_FLAGS – Bit flag, more than 1 flag can be enabled

- ASSISTED_REMOTE_CATCHUP: The stream is in assisted remote catchup state.
- ASSISTED_MEMBER_ACTIVE: During assisted remote catchup, the member on primary that is being assisted is active. This is an abnormal condition because an active member is expected to connect to the standby database directly.
- STANDBY_LOG_RETRIEVAL: The standby database is interacting with the log archive device to retrieve log files.
- STANDBY_RECV_BLOCKED :The standby database temporarily cannot receive logs.
- STANDBY_LOG_DEVICE_FULL: The standby log device is full. This condition blocks receiving of logs until space is released as replay proceeds.
- STANDBY_REPLAY_NOT_ON_PREFERRED :The current replay member on the standby is not the preferred replay member.
- STANDBY_KEY_ROTATION_ERROR: The standby database encountered a master key rotation error. No logs are received until the error is corrected. The system shuts down if the error is not corrected within the timeout period (30 minutes).
- STANDBY_TABLESPACE_ERROR: The standby database has a table space in an invalid error state and can no longer replay transactions that affect it. Replay of transactions on other valid table spaces continues.
- TCP_PROTOCOL: Communication between the primary and standby databases is configured with TCP/IP protocol.
- SSL_PROTOCOL: Communication between the primary and standby databases is configured with secure sockets layer (SSL) protocol

Tools available to measure HADR related performance

- Check out https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/DB2HADR/page/Perf%20Tuning

The following tools are now included as part of the Db2 engine installation

- HADR Simulator to measure network and disk speed
- DB2 Log Scanner to analyze database workload
- HADR Calculator to estimate performance of various HADR sync modes

4 steps to Db2 HADR Performance Tuning

- Step 1: Know Your Workload
  - Db2 log scanner
- Step 2: Know Your Disks
  - Db2 HADR Simulator
- Step 3: Know Your Network
  - Ping + DB2 HADR Simulator
- Step 4: Know Your Sync Modes
  - Db2 HADR Calculator

Step 1: Know Your Workload

DB2 log scanner – analyze the commit patterns in your transactions logs

Located in sqllib/bin directory

db2logscan S0003158.LOG S0003159.LOG ... S0003214.LOG > daily.scan

To scan a large number of files, you can create a file with the file names in it, one per line, then feed the file to scanner via the -f option. If some files need to be retrieved from archive, you can use the -retrieve option.

Please save all output files from the scanner. We will need them in later steps.

Step 2: Know Your Disks

- DB2 HADR simulator to measure performance of your logging disks. Disk speed is modeled as:

  write_time = per_write_overhead + data_amount / transfer_rate

- Command for small write (single page write to calculate write_overhead):

  simhadr -write <logPath>/testFile -flushSize 1

  Sample output:

  Total 9409 writes in 4.000160 seconds, 0.000425 sec/write, 1 pages/write
  Total 38.539264 MBytes written in 4.000160 seconds. 9.634431 MBytes/sec

  This means per_write_overhead is 0.000425 second (from the highlighted "sec/write" field).

  Command for big write (1000 pages per write to calculate transfer rate):

  simhadr-write <logPath>/testFile -flushSize 1000

  Sample output:

  Total 174 writes in 4.014913 seconds, 0.023074 sec/write, 1000 pages/write
  Total 712.704000 MBytes written in 4.014913 seconds. 177.514183 MBytes/sec

  This means transfer rate is 177.514183 MBytes/sec (from the highlighted "MBytes/sec" field).

Step 3: Know Your Network

- Measure HADR primary/standby network speed. Network speed is specified by two numbers: send rate and round trip time. Round trip time can be easily measured by the "ping" command. Send rate may be sensitive to socket buffer size. A special procedure is used for the measurement.
- See Using HADR simulator to determine optimal TCP window size for detailed instructions. This step only gives the tentative socket buffer size based on TCP requirement. Final socket buffer size will also depend on the workload and flush size. It will be determined in step 4. Remember to configure DB2 with the chosen socket buffer size. Otherwise, DB2 won't get the send rate from the simulator test. See TCP Tuning for more info.
- In step 4, use "-network <send_rate> <round_trip_time>" for network speed option for HADR calculator.

Step 4: Know Your Sync Modes

HADR calculator

Usage: hadrCalculator.pl [options] <inputFile1> <inputFile2> ... <inputFileN>

HADR calculator annotates db2logscan output with theoretical HADR data rate.

-syncmode <s> Specify one or more HADR sync modes. Modes are
SYNC, NEARSYNC, ASYNC, or SUPERASYNC (case insensitive).
Multiple modes can be specified as comma delimited list.
Default "SYNC,NEARSYNC,ASYNC".

-network <f1> <f2> Specify primary-standby network speed as

<f1> MBytes/sec

with round trip time of <f2> second.

-disk <f1> <f2> Specify disk write speed as

<f1> MBytes/sec

with overhead of <f2> second per write.

hadrCalculator.pl -disk 200 0.001 -network 10 0.1 <scanOutFile1> ... <scanOutFileN> > daily.scan.calc

Step 4: Know Your Sync Modes

Now look at the calculator output. Look for question mark ("?") in the output. Calculator uses question marks ?, ??, and ??? to indicate small, medium, and heavy impact to applications when HADR is enabled. If no question mark is found, then all is good (expect no impact to applications at all).

Once you have chosen a sync mode, find the max flush size section at end of calculator output:

SYNC	Max flush size: predicted 360 pages, workload max 1126 pages
NEARSYNC	Max flush size: predicted 360 pages, workload max 1126 pages
ASYNC	Max flush size: predicted 17 pages, workload max 1126 pages

HADR socket buffer size should be at least the "predicted" size of your chosen sync mode. This is the expected flush size for the given sync mode. Due to variations in workload, a larger size (such as workload max, which happens when log writing is slower than predicted) is preferred. Both predicted and workload max flush sizes are computed from average transaction size of log rate sample intervals. Actual workload may have peak size above the average. Thus a number even higher than workload max is preferred, as long as the size is reasonable (no more than 32MB). In the above example, a 2000 page (8MB) socket buffer size is recommended.

For verification, run HADR simulator using the chosen sync mode, socket buffer size, disk speed, and predicted flush size to confirm that the system will perform as expected (compare the throughput from simulator to the line with the same flush size in calculator output). Even though this is simulation, the network is stressed with real data. A simulator run with the final parameters is strongly recommended before applying the parameters to the database.

Output is very detailed

```
+ simhadr -role primary -lhost brant -lport 46000 -rhost auklet -rport 46000 -sockSndBuf 64000 -sockRcvBuf 64000 -flushSize 32

Measured sleep overhead: 0.003608 second, using spin time 0.004329 second.
…
NEARSYNC: Total 410255360 bytes in 4.000821 seconds, 102.542793 MBytes/sec
Total 3130 flushes, 0.001278 sec/flush, 32 pages (131072 bytes)/flush

Total 410255360 bytes sent in 4.000821 seconds. 102.542793 MBytes/sec
Total 10893 send calls, 37.662 KBytes/send,
Total 3136 congestions, 0.848714 seconds, 0.000270 second/congestion

Total 150240 bytes recv in 4.000821 seconds. 0.037552 MBytes/sec
Total 3130 recv calls, 0.048 KBytes/recv
…
```

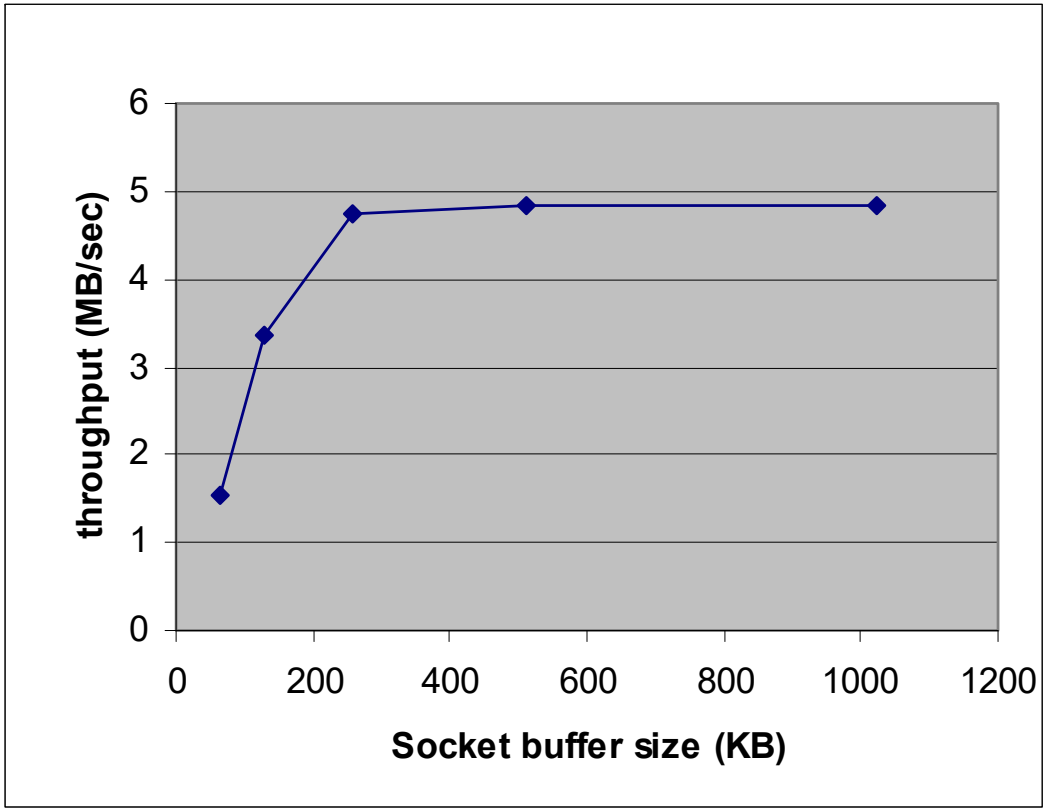
- The throughput is 102 MB/second, on a gigabit network (raw bandwidth = 125 MB/s)
- NEARSYNC mode, where round trip messaging is required for each flush (primary needs an ack message from standby).
- Simhadr reports average log shipping time as 1.3 millisecond. Compared to .1 millisecond round trip time on this network (measured by "ping"), the overhead of ack message is insignificant.
- Thus NEARSYNC mode throughput is not too far behind the raw bandwidth of the network

Using the HADR simulator

- Below is actual result between IBM labs in Portland, Oregon, and Silicon Valley, California
- The physical distance is about 1000km
- Buffer Throughput results:

  64KB 1.554048 MBytes/sec
  128KB 3.347476 MBytes/sec
  256KB 4.734673 MBytes/sec
  512KB 4.834047 MBytes/sec
  1MB 4.821998 MBytes/sec

- Throughput peaks at 4.8 MB/sec with buffer size at around 256KB
- Set DB2_HADR_SOSNDBUF and DB2_HADR_SORCVBUF to 256K



Agenda

- Comparison to other DB2 HA offerings
- HADR Review
- HADR Sync modes
- HADR Multiple Standby
- HADR Configuration
- HADR Monitoring
- **Recent Improvements**

Monitoring HADR Tablespace Status

- Tablespace Error State
  - When a tablespaces is in an invalid or error state on the Standby database, the `HADR_FLAGS` field will display the value `STANDBY_TABLESPACE_ERROR`
- Monitoring with db2pd
  - The `HADR_FLAGS` field can be monitored by using the `db2pd -hadr` command on the Primary or Standby database

```
db2pd -db HADRDB1 -hadr
… HADR_FLAGS = STANDBY_TABLESPACE_ERROR TCP_PROTOCOL …
```

- Monitoring with Table Function
  - The `MON_GET_HADR()` table function will display the current status on either the Primary database or Standby database with Reads on Standby enabled

```
SELECT STANDBY_ID, HADR_FLAGS FROM
       TABLE(MON_GET_HADR(NULL))
STANDBY_ID
---------- ------------------------------------
         1 STANDBY_TABLESPACE_ERROR TCP_PROTOCOL
```

Db2 HADR recent enhancements

V 11.5.6

- Database connection hang detection for automated HADR
  - export DB2_HADR_HANG_DETECTION=CONNECT
  - export DB2_HADR_HANG_SQL_BYPASS=SQL1040N
- Db2U support for Multiple Standbys

V 11.5.8

- Log replay performance improvements

V 11.5.9

- Improved HADR standby replay performance

V 12.1

- Hostname Validation On By Default between primary and standby

Db2 HADR recent enhancements con't

V 12.1

- a new HADR upgrade procedure provides read only access to the HADR standby database while upgrading the primary database.
- DB2_HADR_BLOCK_ON_DISKFULL registry variable controls the behavior of an HADR standby that is running into a disk full condition in the active log path.
- HADR_SSL_LABEL and HADR_SSL_HOST_VAL database configuration parameters are now supported in pureScale environment

V 12.1.3

- Mixed member topology support for Db2 pureScale HADR configurations
- pureScale Drop member support

V 12.1.5

- POSSIBLY additional standby support

Combining HADR and Storage Replication

Provide continuous availability,

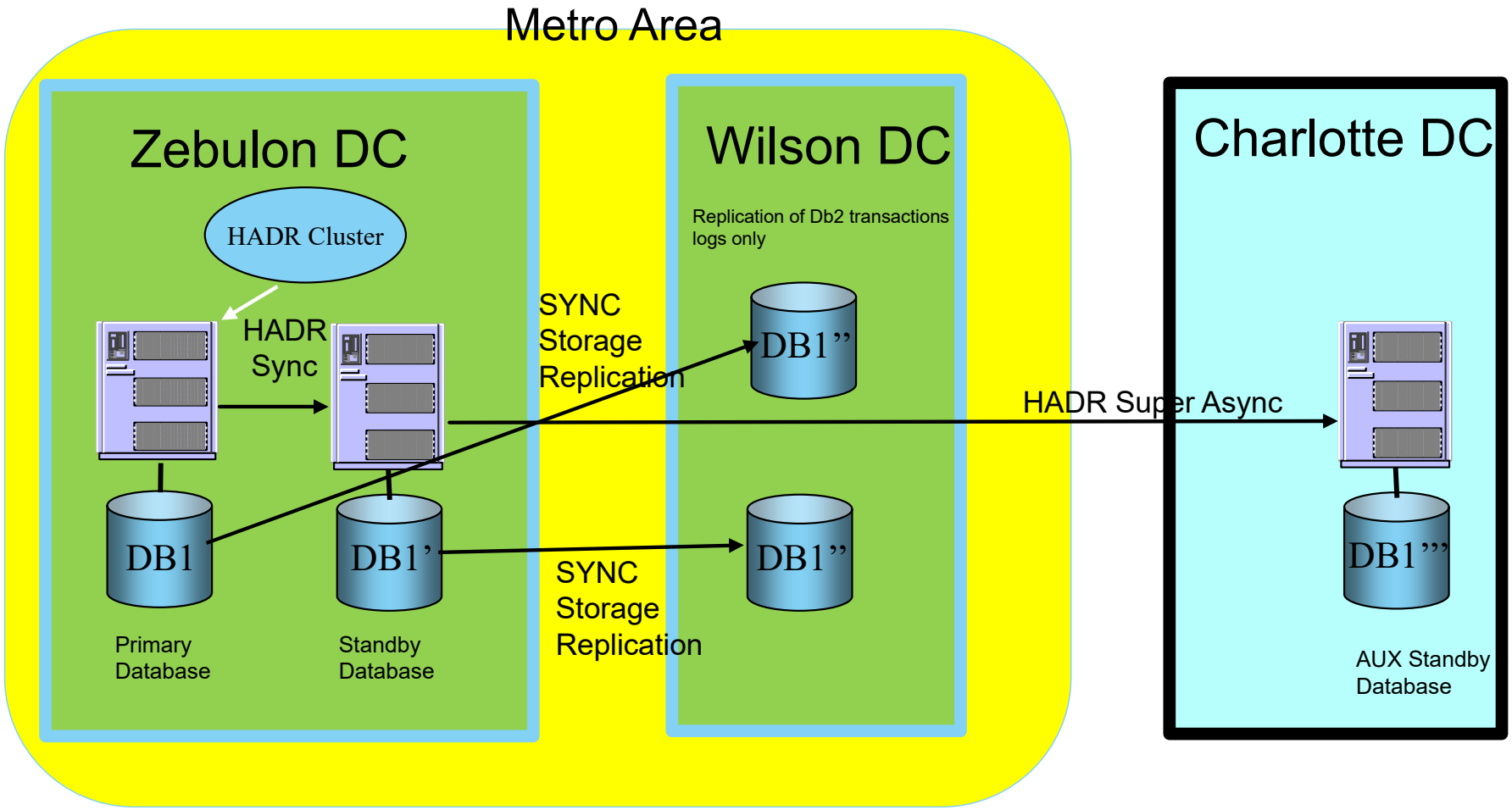
Protect from localized and site wide failures

Zero data loss in the case of a DR

RPO must be 0 whereas RTO can be in minutes

Ability to exploit a “bunker site” near the primary data center

Recommended Configuration



Combining HADR and Q Repl

Provide continuous availability,

Protect from localized and site wide failures

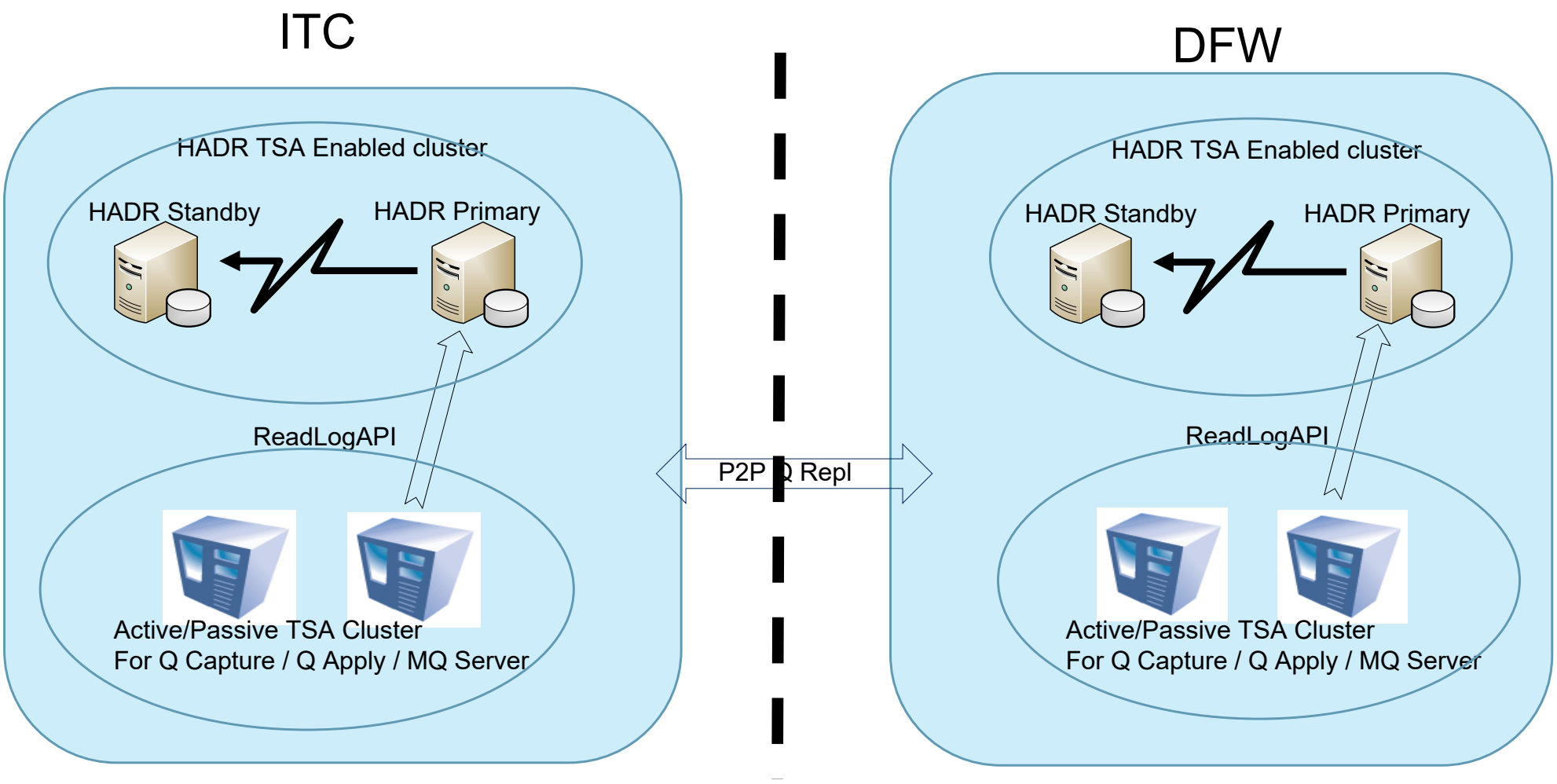
Provide ability to roll out upgrades with zero downtime

Application changes including schema changes must be handled

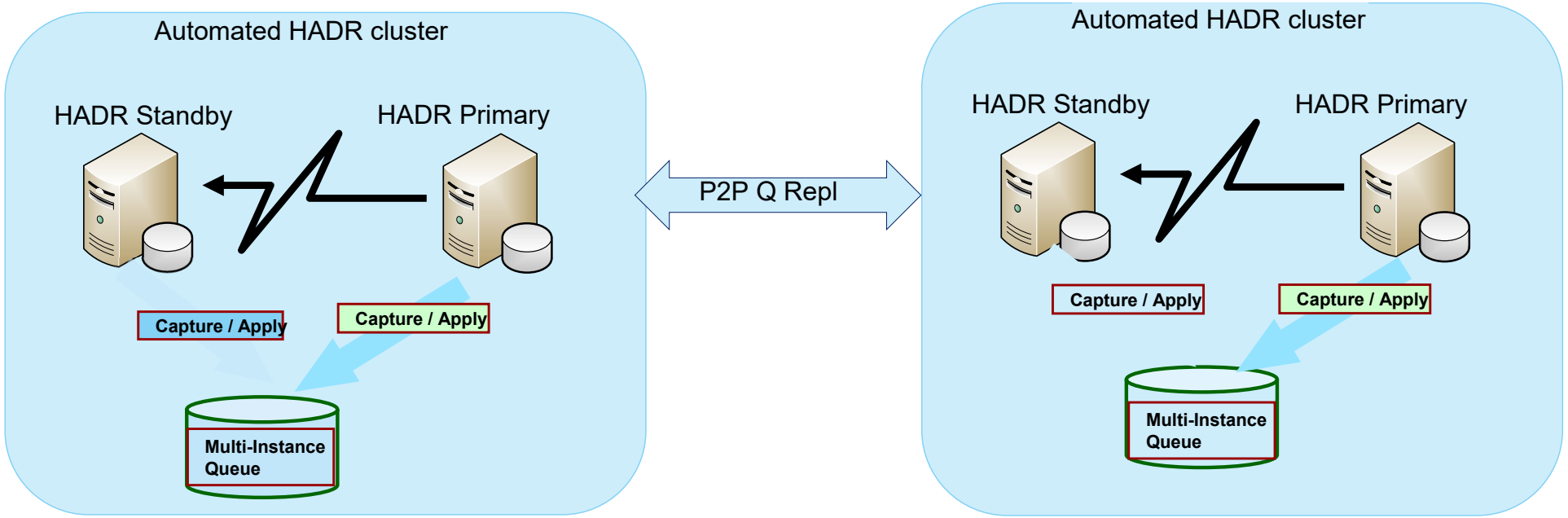
Ability to split workload to eliminate conflicts

Combination of Akamai and F5 routing

Active-Active Design



Local Active-Active Design Modified



Requests for Enhancements 1/3

Home / All ideas / DB24LUW-I-1432 Add a new idea Subscribed

12

VOTED

Status Future consideration

Workspace Db2

Components High Availability and/or Disaster Recovery

Created by Guest

Created on Jan 25, 2022

Backup on HADR Standby

See this idea on ideas.ibm.com

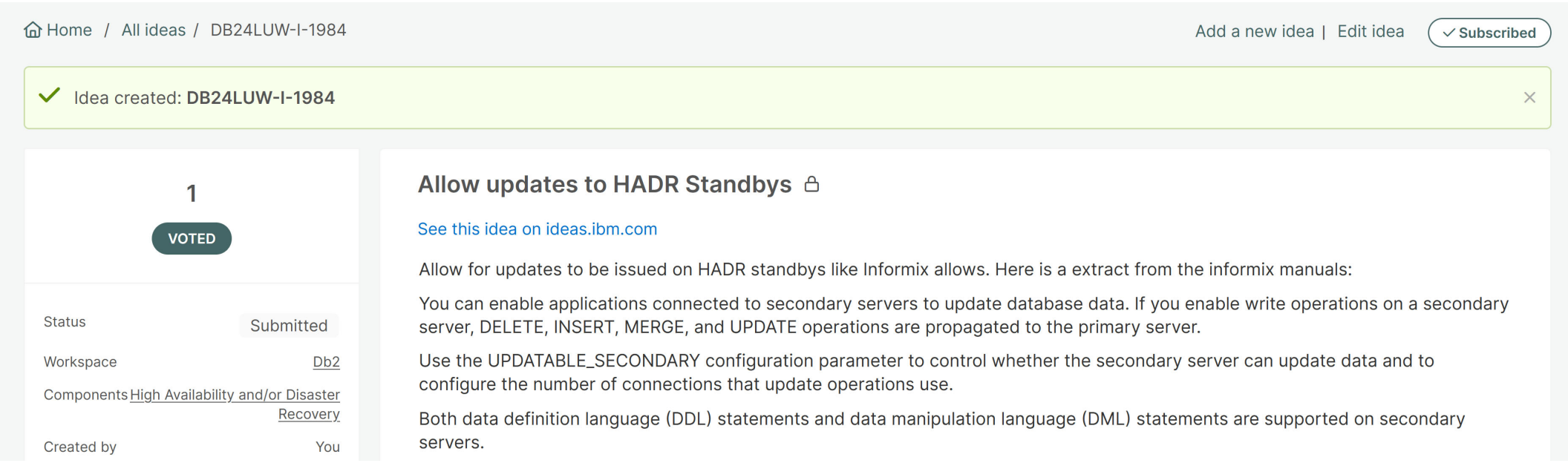
Backup consume resources in primary. If we offload backup in Standby then Primary can be easily served for application use alone . Currently we were able to take backup only in primary and all workloads along with Backup utilise CPU resources .If we have capability or feature to enable which allows Backup to run Standby then it will be easy for Primary server to foucs on application workloads alone.

Needed By Week

COMMENTS 0 MERGED IDEAS 1

Requests for Enhancements 2/3

Requests for Enhancements 3/3



Dale McInnis

IBM Canada Ltd.

dmcinnis@ca.ibm.com

Session IBM DB2 HADR State of the Union