Continuous Availability with DB2

Dale McInnis

IBM Canada Ltd.

dmcinnis@ca.ibm.com

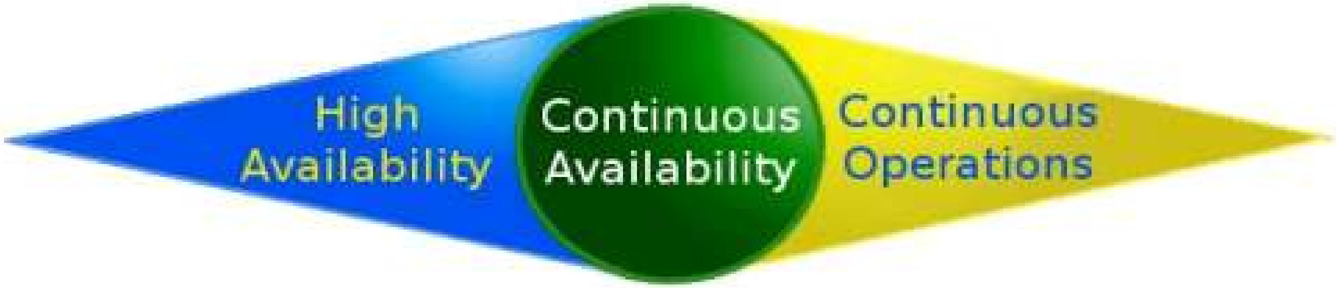
Agenda

- Definitions
- Why is resilience important
- How does DB2 address these availability challenges
- Tips and Techniques
- What are real customers doing

What is Continuous Availability?

- Wikipedia: **Continuous Availability** is an approach to computer system and application design that protects users against downtime, whatever the cause and ensures that users remain connected to their documents, data files and business applications. Continuous availability describes the information technology methods to ensure business continuity.[

High Availability = minimize downtime
Continuous Availability = eliminate downtime



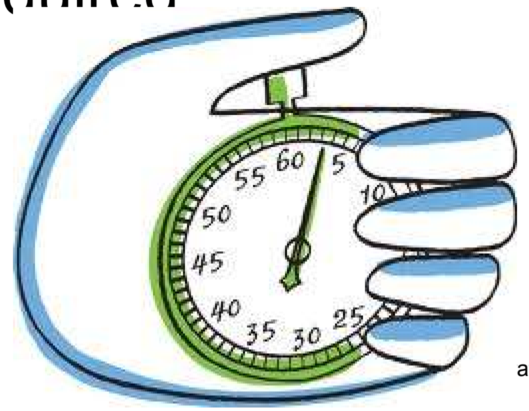
What is DB Resilience?

DB Resiliency is the activity performed by the IT organization to ensure that critical database services will be available when needed.

This will encompass what we traditionally think of as High Availability (HA) as well as Disaster Recovery (DR) and not limited to either.

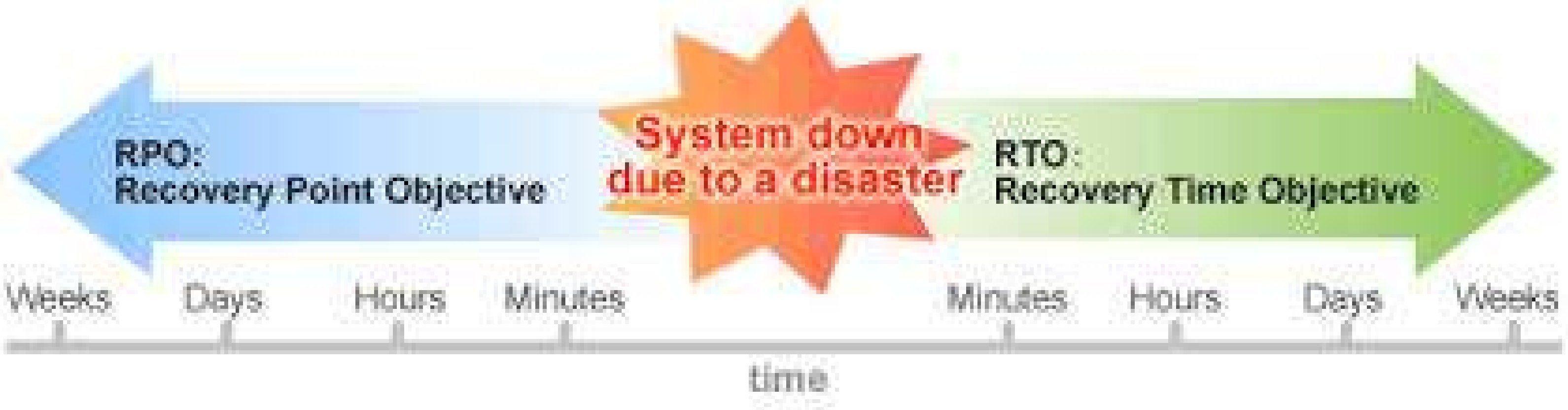
What is RTO?

- The **Recovery Time Objective** (RTO) is the duration of time and a service level within which a business process must be restored after a disaster (or disruption) in order to avoid unacceptable consequences associated with a break in business continuity
- It should be noted that the RTO attaches to the business process and not the resources required to support the process.



What is RPO?

- **Recovery Point Objective** (RPO) describes the acceptable amount of data loss measured in time.
- The Recovery Point Objective (RPO) is the point in time to which you must recover data as defined by your organization. This is generally a definition of what an organization determines is an "acceptable loss" in a disaster situation
- RPO is typically 0 for HA and non-zero for DR



What are customers asking for?

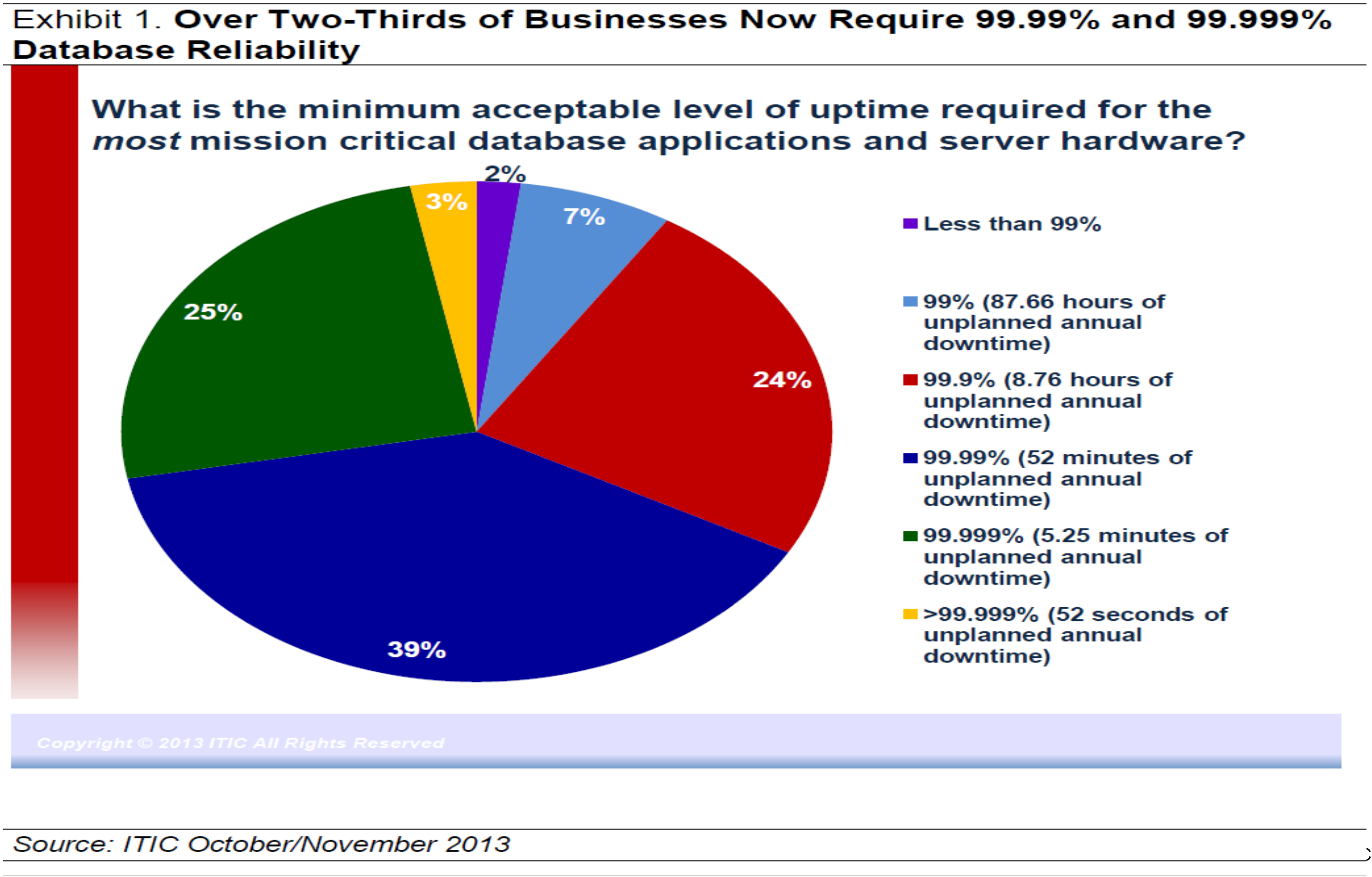


Exhibit 1. **Over Two-Thirds of Businesses Now Require 99.99% and 99.999% Database Reliability**

What is the minimum acceptable level of uptime required for the *most* mission critical database applications and server hardware?

Source: ITIC October/November 2013

Agenda

- Definitions
- Why is resilience important
- How does DB2 address these availability challenges
- Tips and Techniques
- What are real customers doing

Why is continuous availability important?

- Importance of disaster recovery environments are getting a lot of press
  - Retailer lost $25 Million due to a flood
  - A health agency incurred $100 million in additional cost during a 3 day outage
  - **IDC: “90% of SMBs that experience disasters file for bankruptcy within 12 months”**
- IDC: “Good DR is about preparation, planning, and practice, and with a price of downtime ranging from $70,000 for a retailer to over $7 million an hour for a wealth mgmt. firm good DR is just good business”

Business Continuity

k10163024 fotosearch.com

The top Causes of Business Interruption:

1. Planned Maintenance
   - System and Software Upgrades or Reconfigura
   - Database Administration

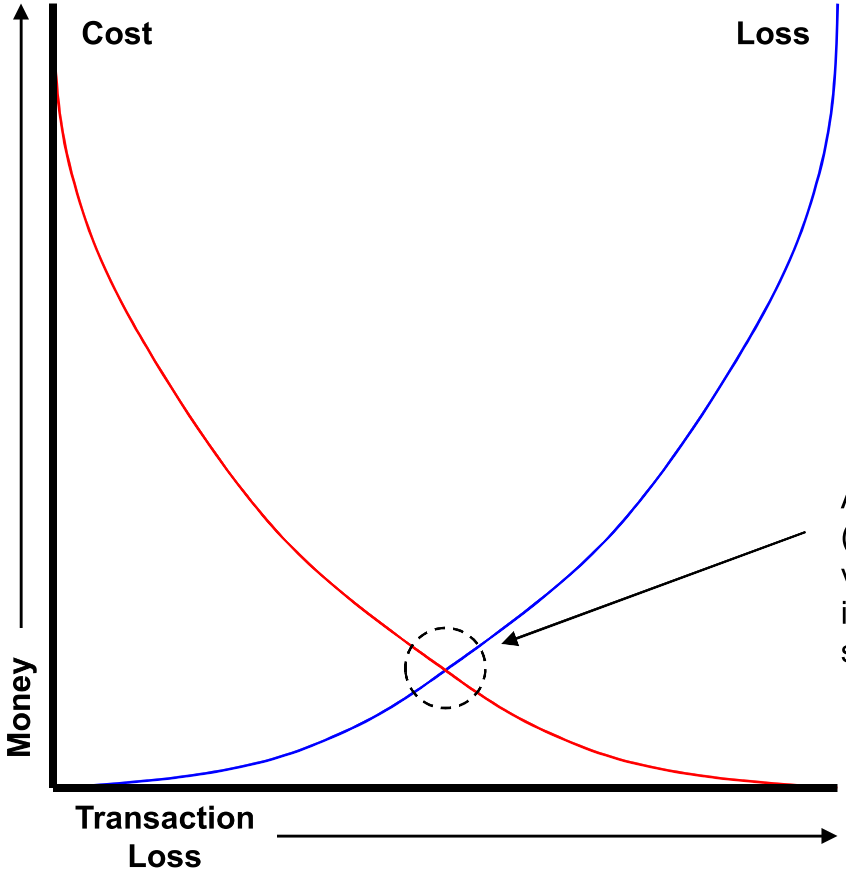
2. Component Failure
   - Operator Errors, Software defects, Disk Failure, Subsystems, Hardware, Power Grid outage
   - *Data is recoverable*
   - But, changes might be *stranded* until component is restored

3. Disaster
   - Flood, Earthquake, Fire, …, Loss of a site
   - *Data is not recoverable*

Addressing Customer Requirements for Business Continuity

- Requires a *shift* in strategy:
  - –From Failover to **Active/Active**
  - –From Local to **Geographically Dispersed**
  - –From a pure Storage play to an **Information Management play focused on the data required for Business Continuity**

Cost vs. Availability

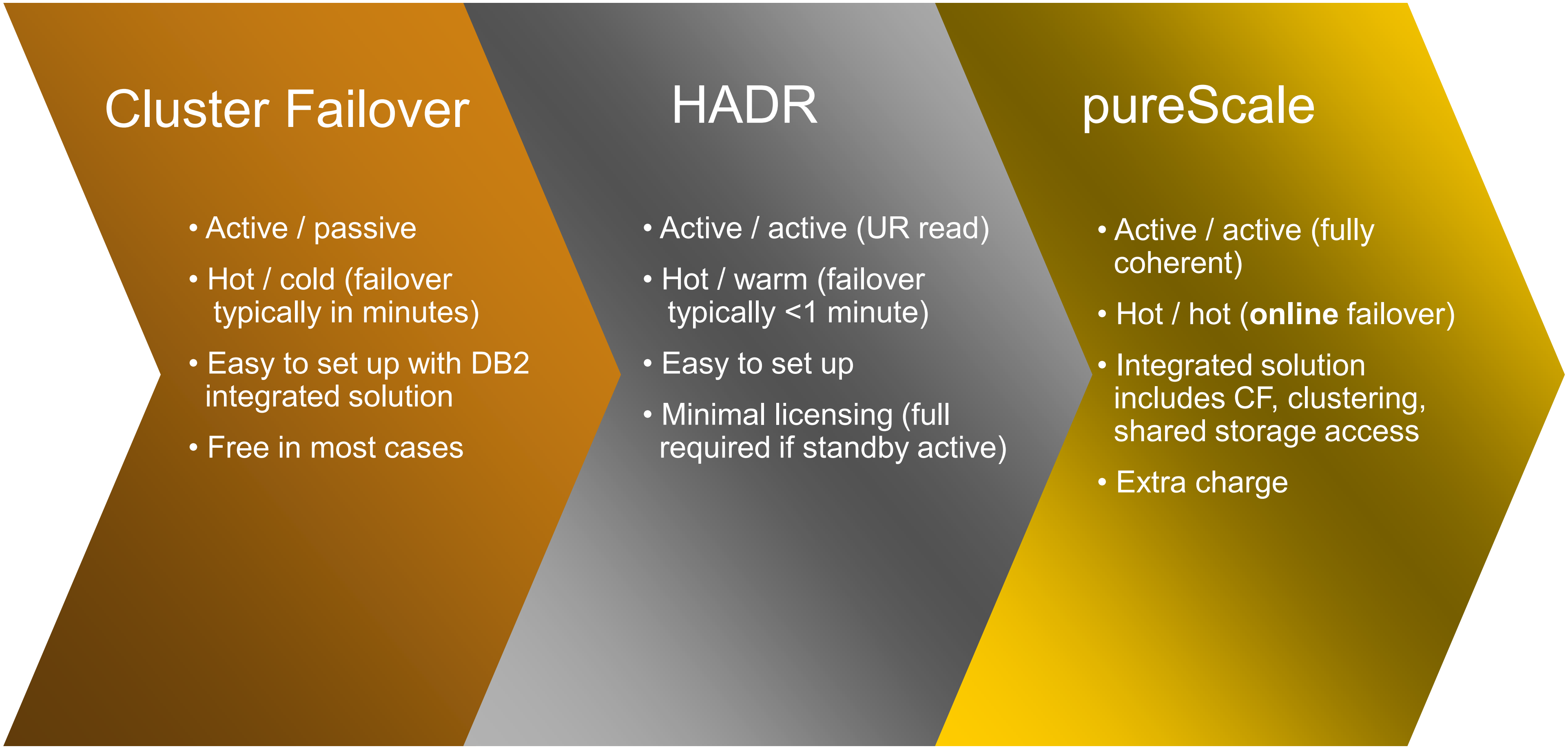


Acceptable transaction loss (both real and potential) versus the cost of implementing an HA strategy

Agenda

- Definitions
- Why is resilience important
- How does DB2 address these availability challenges
- Tips and Techniques
- What are real customers doing

DB2 Local Availability Solutions



Cluster Failover

- Active / passive
- Hot / cold (failover typically in minutes)
- Easy to set up with DB2 integrated solution
- Free in most cases

Bronze

HADR

- Active / active (UR read)
- Hot / warm (failover typically <1 minute)
- Easy to set up
- Minimal licensing (full required if standby active)

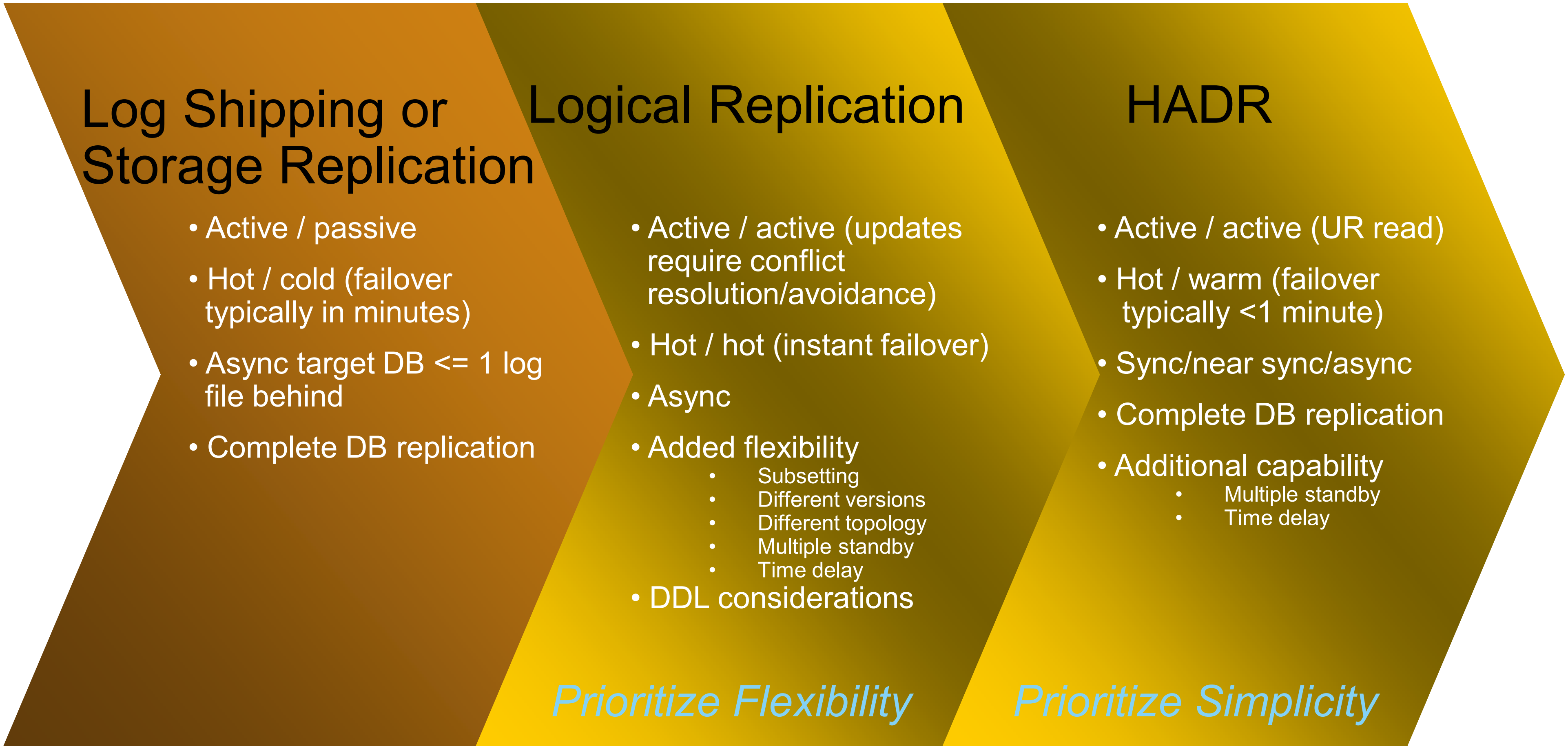
Silver

pureScale

- Active / active (fully coherent)
- Hot / hot (**online** failover)
- Integrated solution includes CF, clustering, shared storage access
- Extra charge

Gold

DB2 Disaster Recovery Solutions



Log Shipping or Storage Replication

- Active / passive
- Hot / cold (failover typically in minutes)
- Async target DB <= 1 log file behind
- Complete DB replication

Logical Replication

- Active / active (updates require conflict resolution/avoidance)
- Hot / hot (instant failover)
- Async
- Added flexibility
  - Subsetting
  - Different versions
  - Different topology
  - Multiple standby
  - Time delay
- DDL considerations

Prioritize Flexibility

HADR

- Active / active (UR read)
- Hot / warm (failover typically <1 minute)
- Sync/near sync/async
- Complete DB replication
- Additional capability
  - Multiple standby
  - Time delay

Prioritize Simplicity

Bronze **Gold** **Gold**

DB2 Disaster Recovery Solutions : Continued

GDPC

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

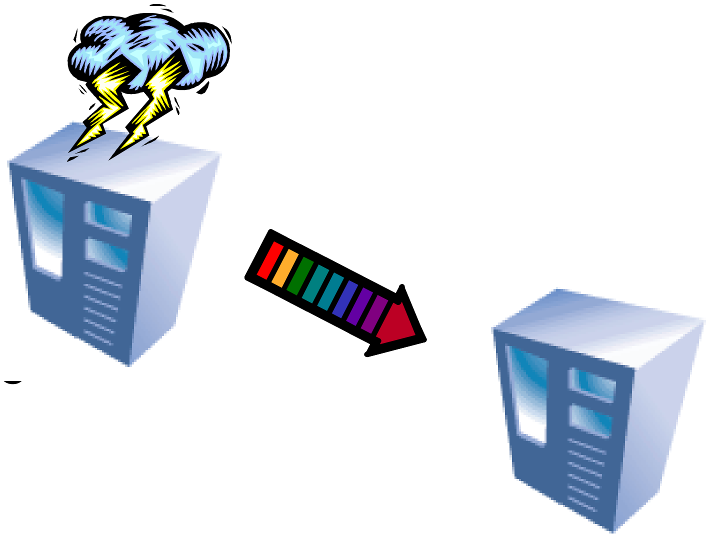
Situational Platinum

DB2 Continuous Availability Features

- There are four major features which provide relief for outages, namely:
  - HADR
  - PureScale
  - GDPC
  - Logical Replication

Basic Principles of HADR

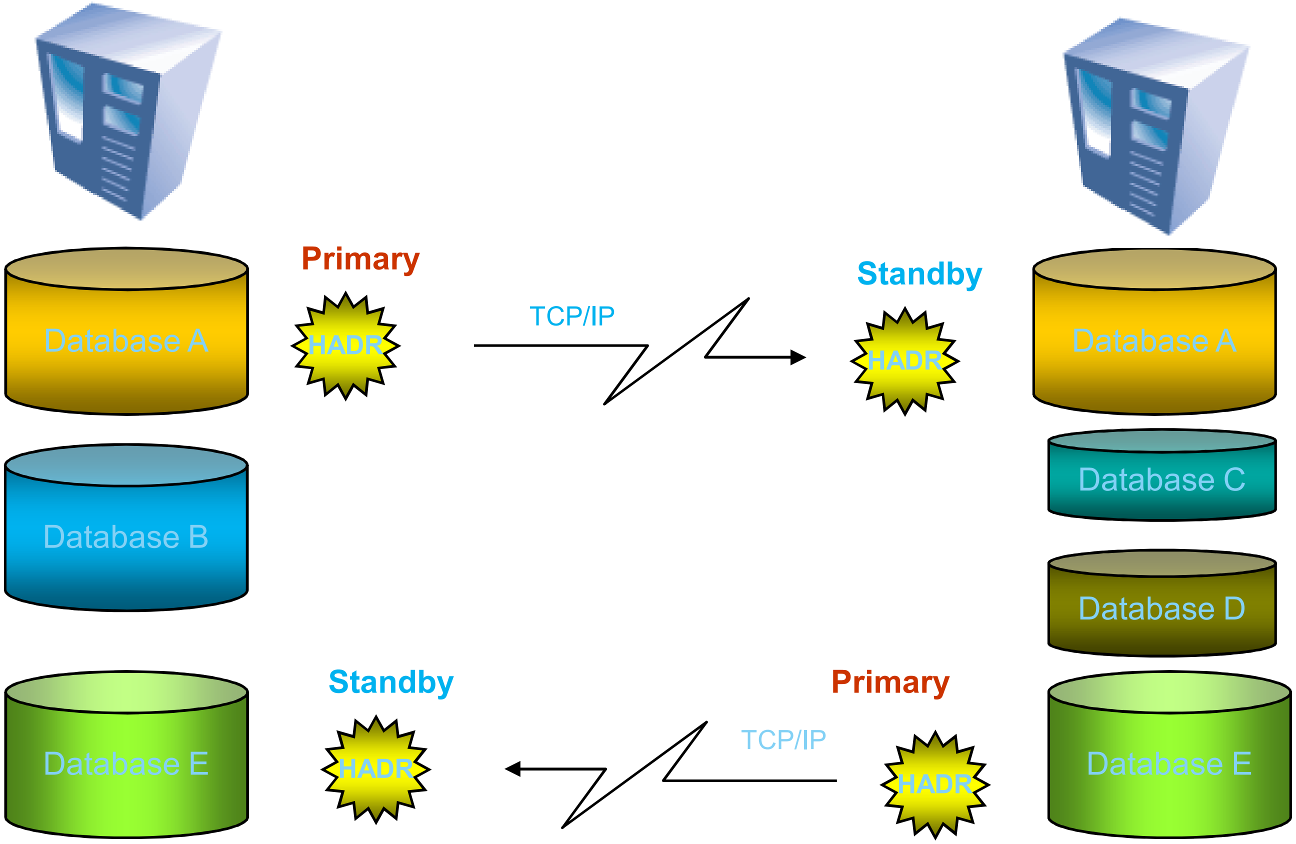
- Two active machines
  - Primary
    - Processes transactions
    - Ships log entries to the other machine
  - Standby
    - Cloned from the primary
    - Receives and stores log entries from the primary
    - Re-applies the transactions



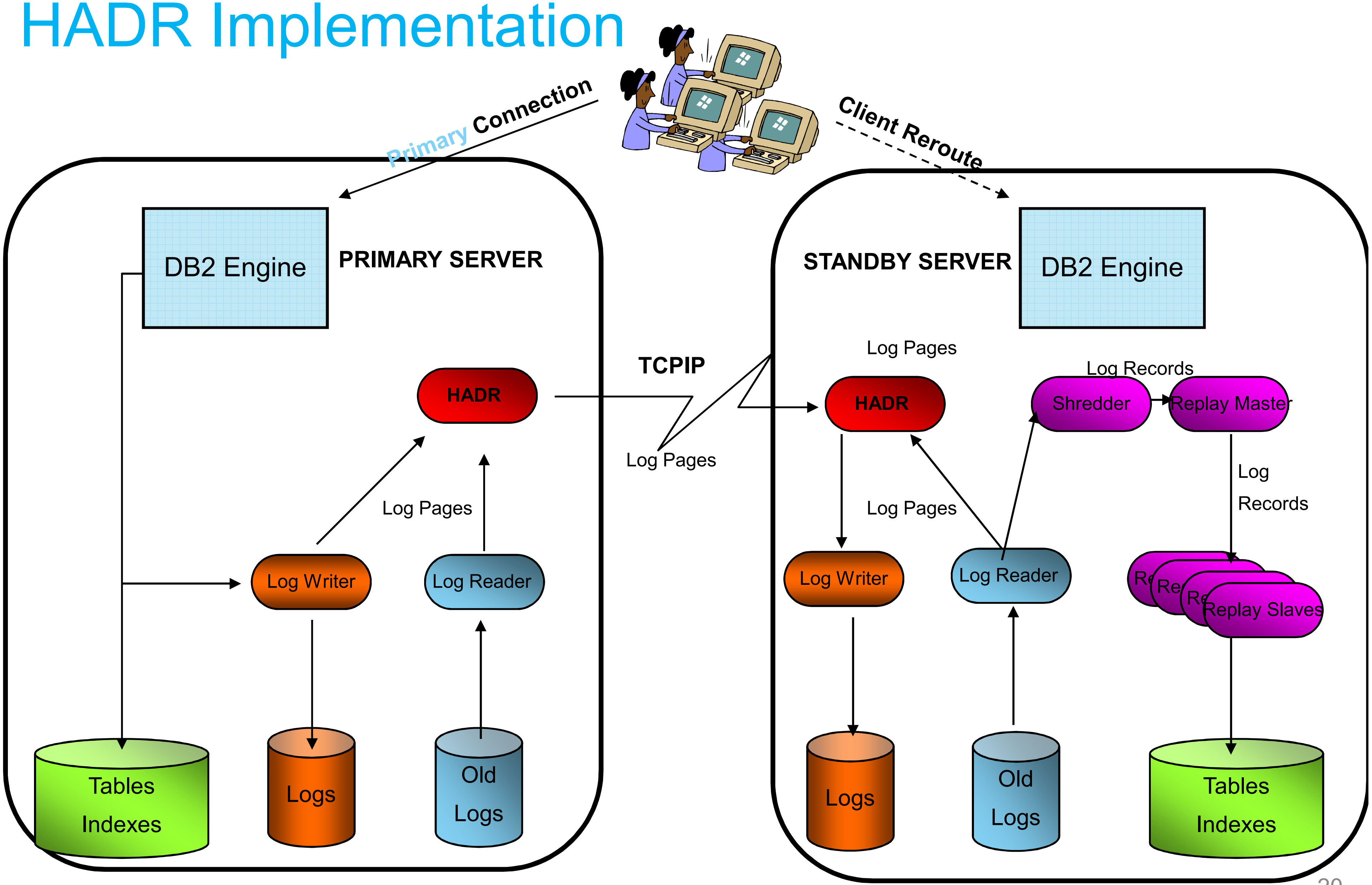
- If the primary fails, the standby can take over the transactional workload
  - The standby becomes the new primary
- If the failed machine becomes available again, it can be resynchronized
  - The old primary becomes the new standby

Scope of Action

HADR replication takes place at the database level.



HADR Implementation



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_LOCAL_SVC     svc_a
    HADR_TARGET_LIST
      host_b:svc_b
    HADR_REMOTE_INST   inst_b
    HADR_TIMEOUT       120
    HADR_SYNCMODE      ASYNC


db2 start hadr on database hadr_db
  as primary
```

Standby Setup

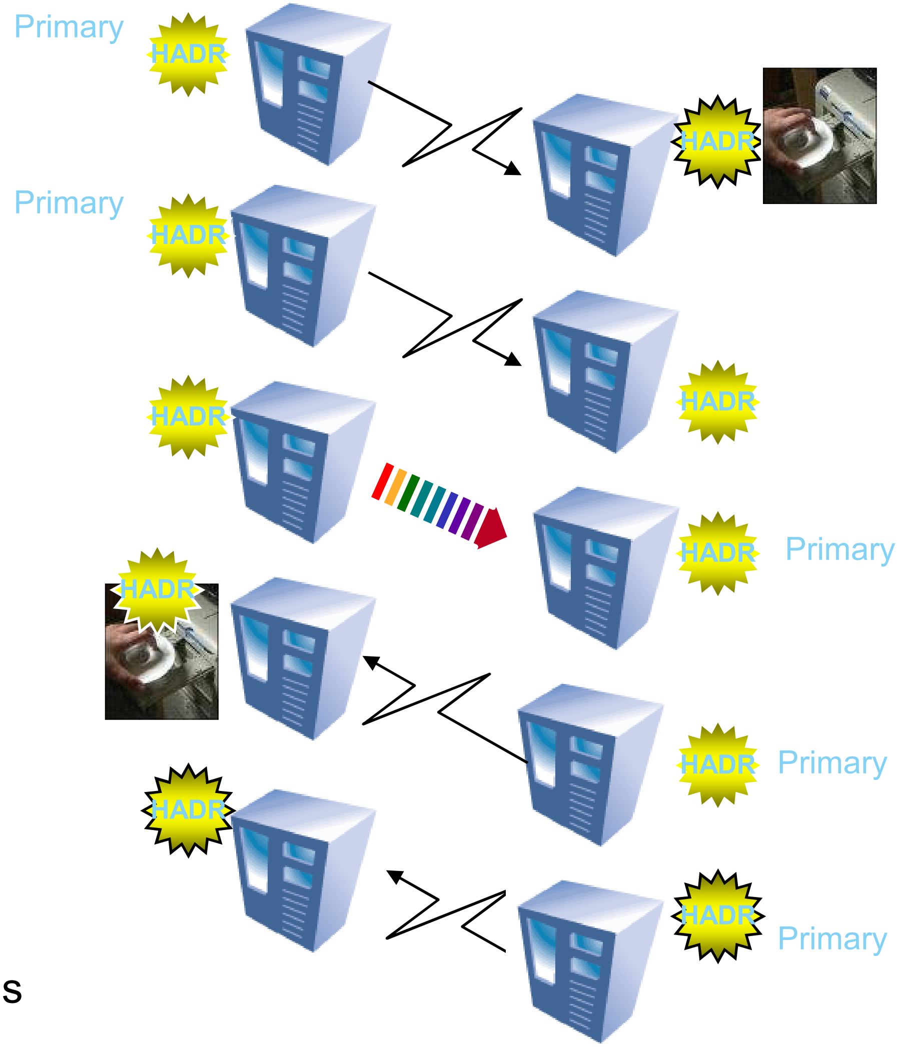
```
db2 restore db hadr_db from
  backup_dir


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


db2 start hadr on database hadr_db
  as standby
```

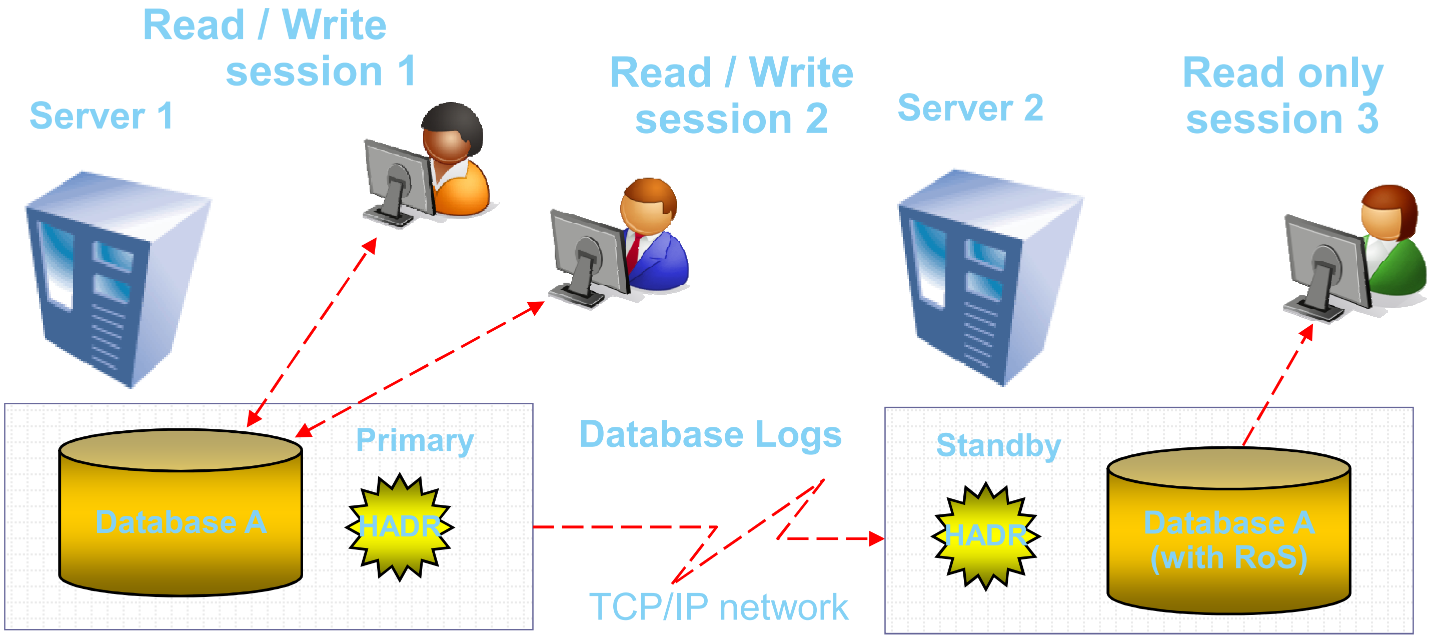
Software upgrades on the fly

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. Suspend 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



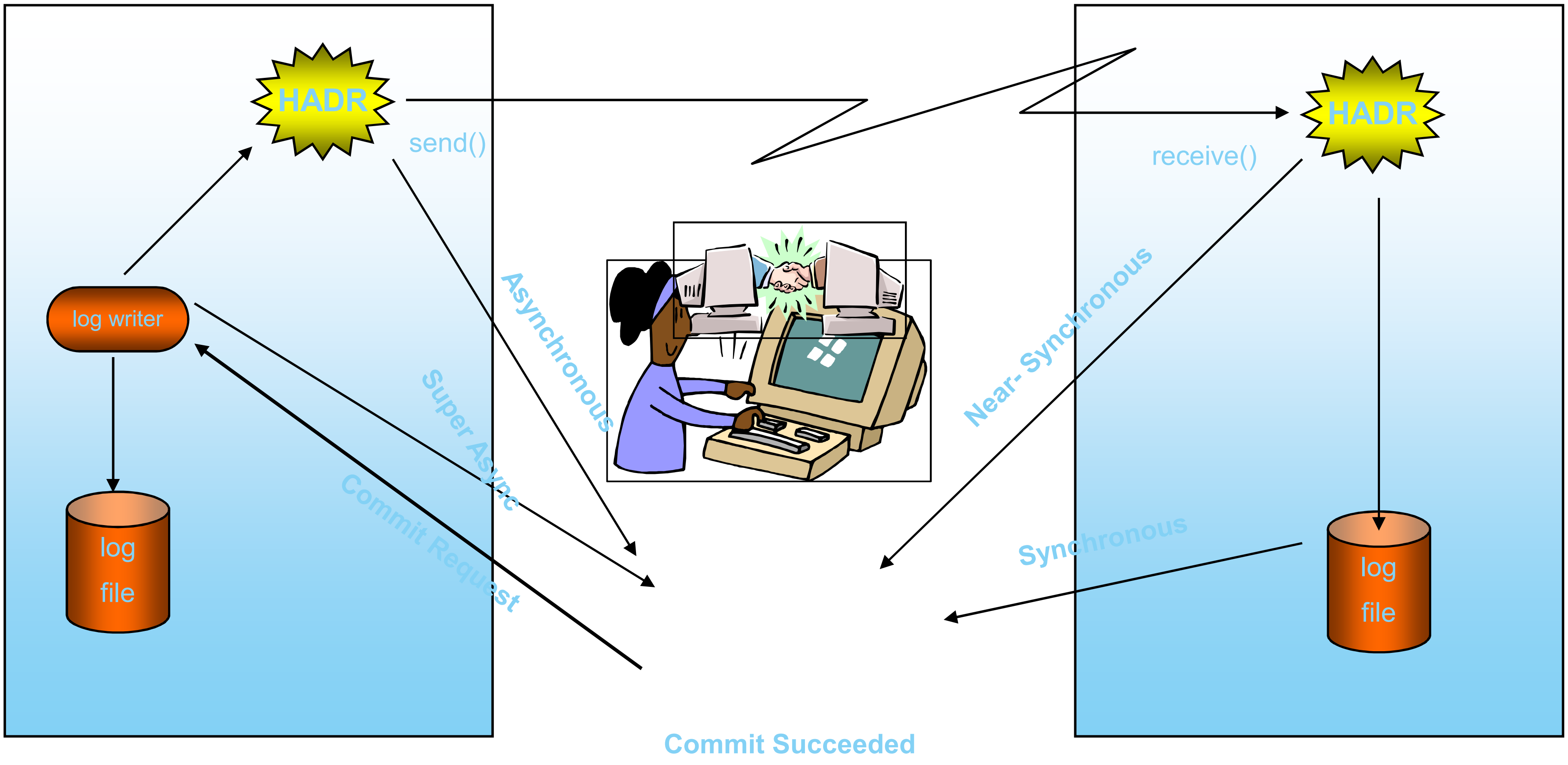
HADR Read On Standby (RoS)

- 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
- **V 9.7 FP5 now supports returning inline LOBs / XML**



Synchronization modes

Sync, Near Sync, Async, Super Async



HADR Standby Log Spooling

- When enabled, the log spooling feature will allow the standby to spool log records arriving from the primary

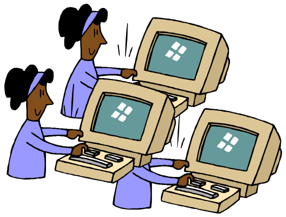
- This decouples log replay on the standby from receiving of the log data from the primary

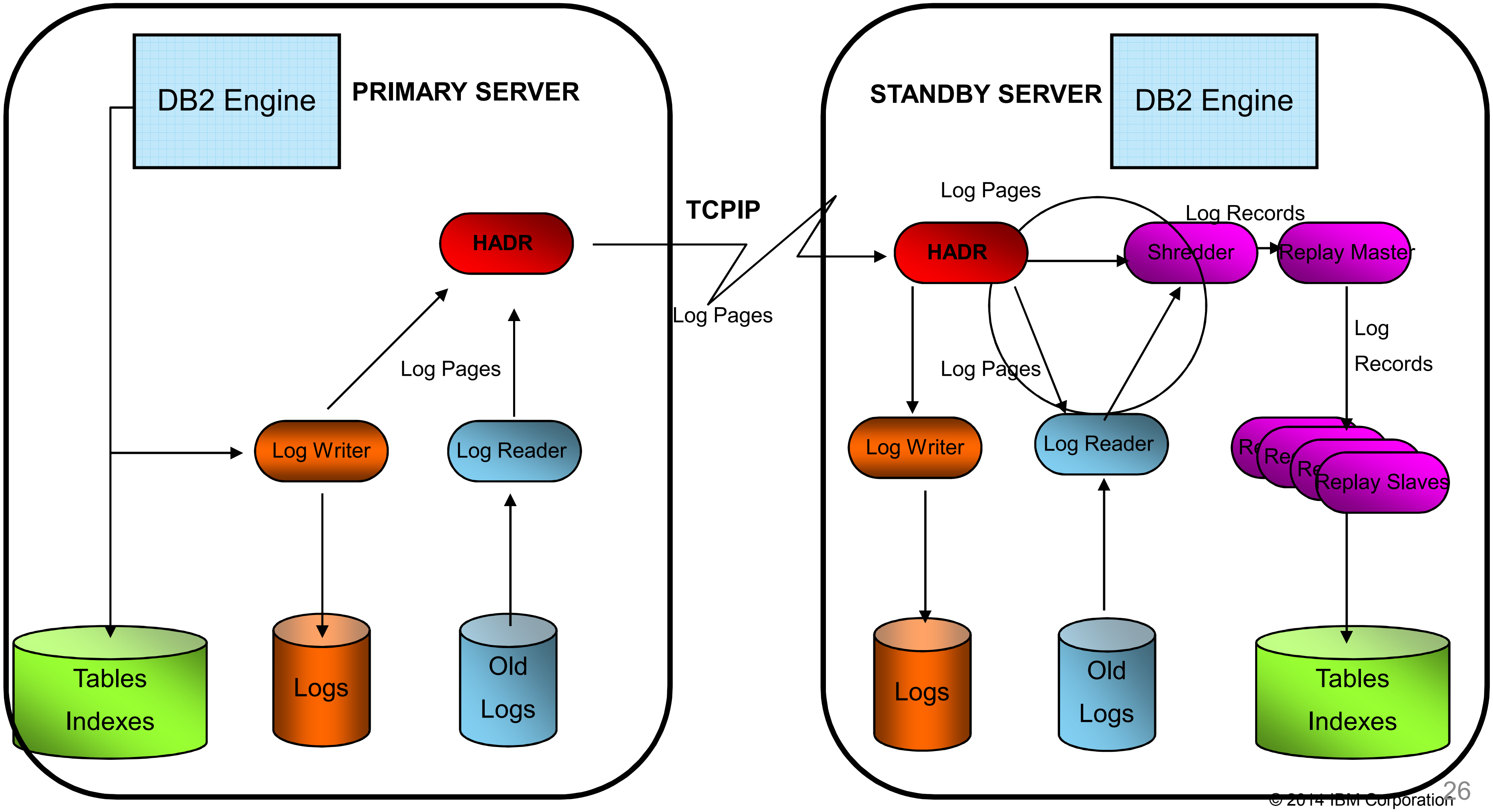
- Logs will be spooled in the standby DB’s active log path

- Set through a new DB CFG parameter, HADR_SPOOL_LIMIT

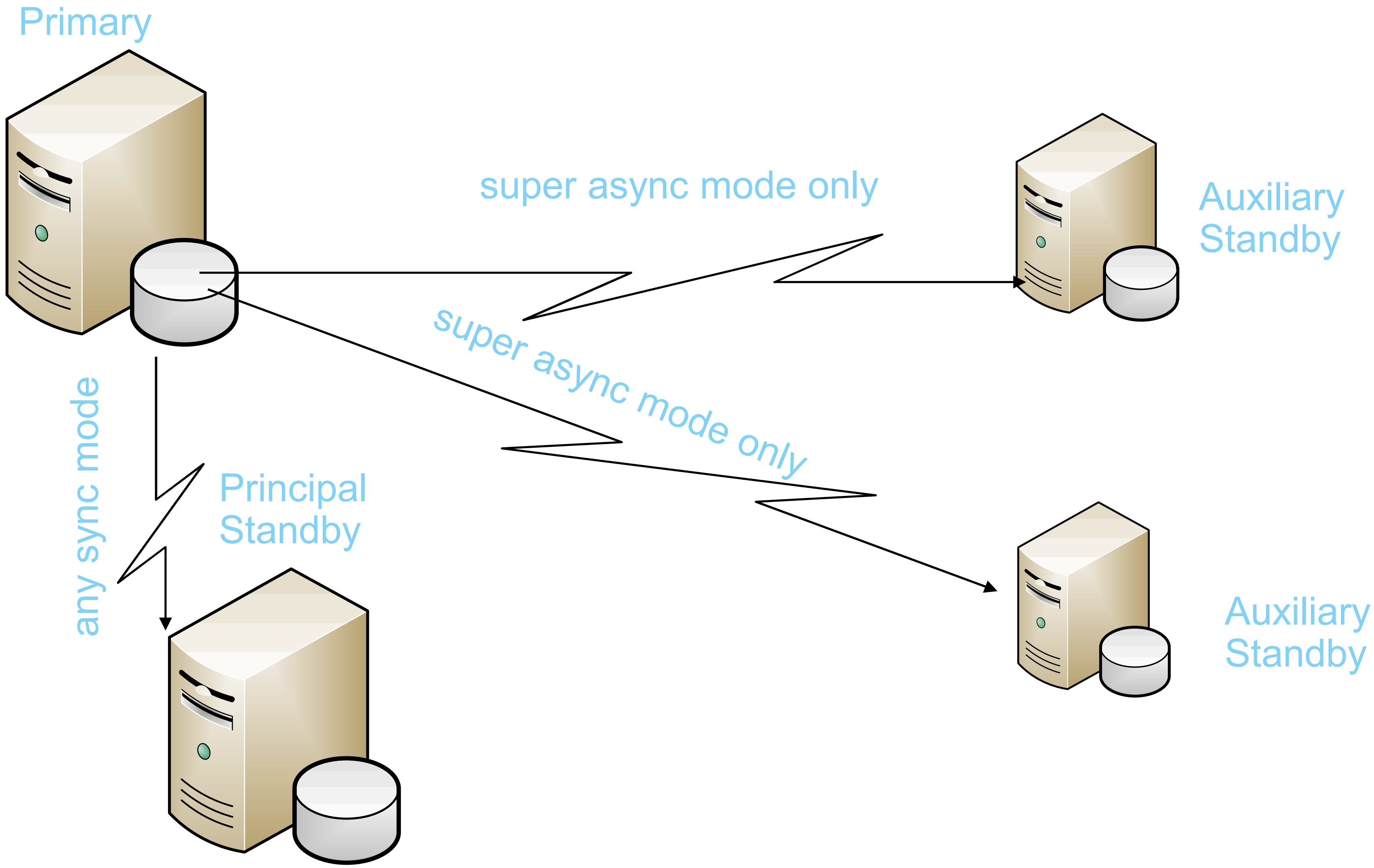
- Can limit the amount of space allocate to the log spool by specifying the maximum amount of disk space to use
  - Value of 0 disables spooling (default in V 10.1)
  - Value of -1 defines the spool to be unlimited (limited by file system free space)
  - Value of -2 (automatic) defines the spool to be LOGFILESIZ*(LOGPRIMARY+LOGSECOND) and is the default in V 10.5

HADR Log Spooling





HADR Multiple Standby Overview



HADR Multiple Standby Enablement

- HADR_TARGET_LIST is used to specify all standbys, both auxiliary as well as the principal standby

- HADR_TARGET_LIST uses a hostname or IP Address and port number format with the “|” character as a delimiter
  - E.g. host1.ibm.com:4000|host2.ibm.com:hadr_service|9.47.73.34:5000

- On each standby the HADR_REMOTE_HOST, HADR_REMOTE_INST, HADR_REMOTE_SVC must point to the current primary

- Primary will validate hostname and port number upon handshake from AS

- Existing single standby installations need no configuration change

HADR Configuration Parameters Updates

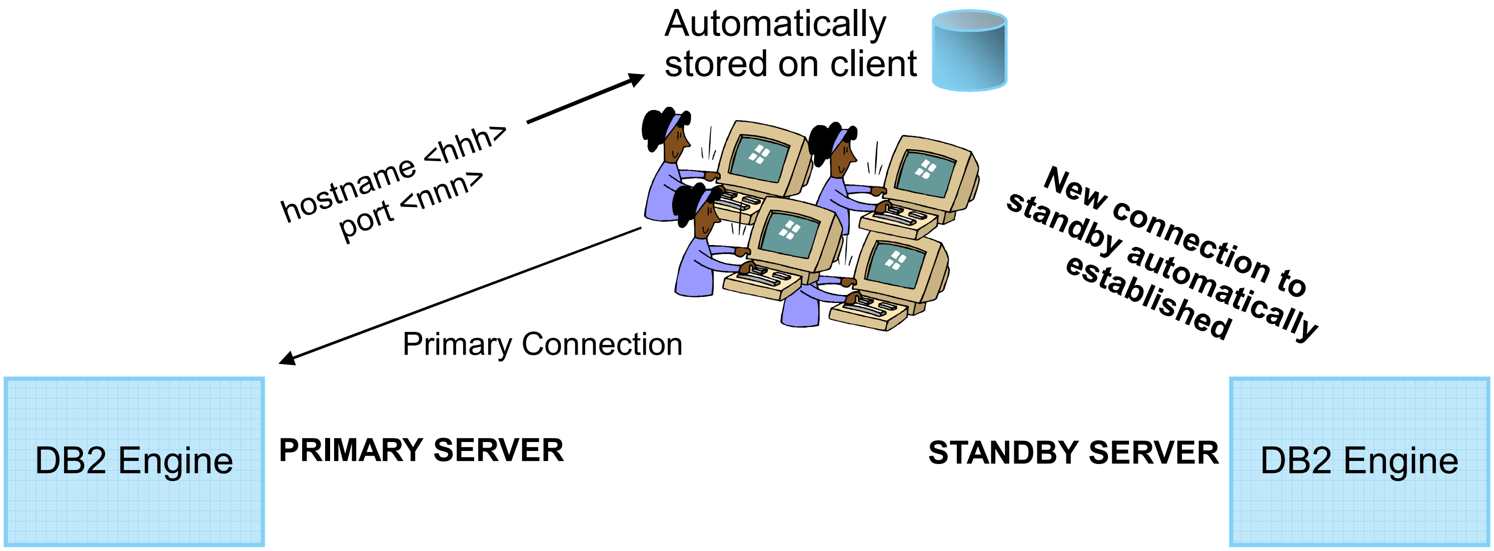
- you need only stop and start HADR for updates to some HADR configuration parameters for the primary database to take effect. You do not have to deactivate and reactivate the database. This dynamic capability affects only the primary database because stopping HADR deactivates any standby database.
- 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_spool_limit**
  - **hadr_syncmode**
  - **hadr_target_list**
  - **hadr_timeout**

Automatic Client Reroute

- Automatic, transparent connection to alternate server when primary connection fails
  - If there is a currently executing SQL statement, it will fail with sqlcode -30108
  - Transaction can then be re-driven without re-establishing a connection

- Alternate information Stored on client
  - System database directory
  - alternateDataSource property (Java Type 4 driver)

- Works with HADR, EE/ESE, EEE/DPF, Replication



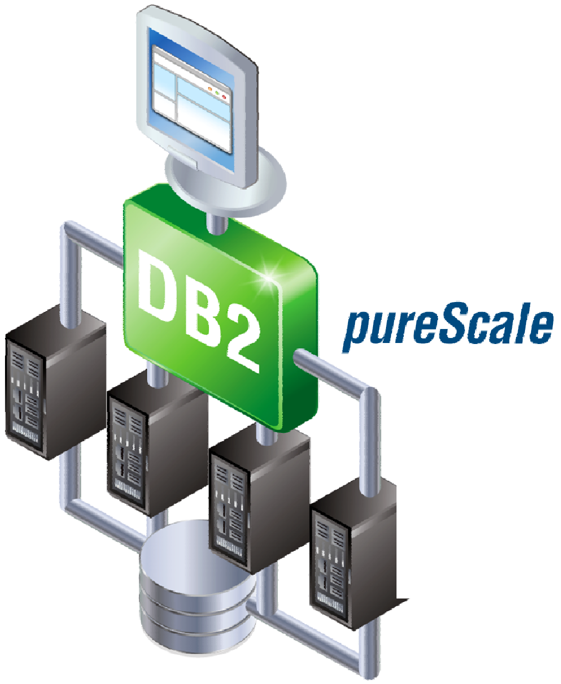
db2 update alternate server for database <dbname> using hostname <hhh> port <nnn>

DB2 Continuous Availability Features

- There are four major features which provide relief for outages, namely:
  - HADR
  - PureScale
  - GDPC
  - Logical Replication

DB2 10.5 Delivers 'Always Available' Transactions
Optimized for OLTP Workloads

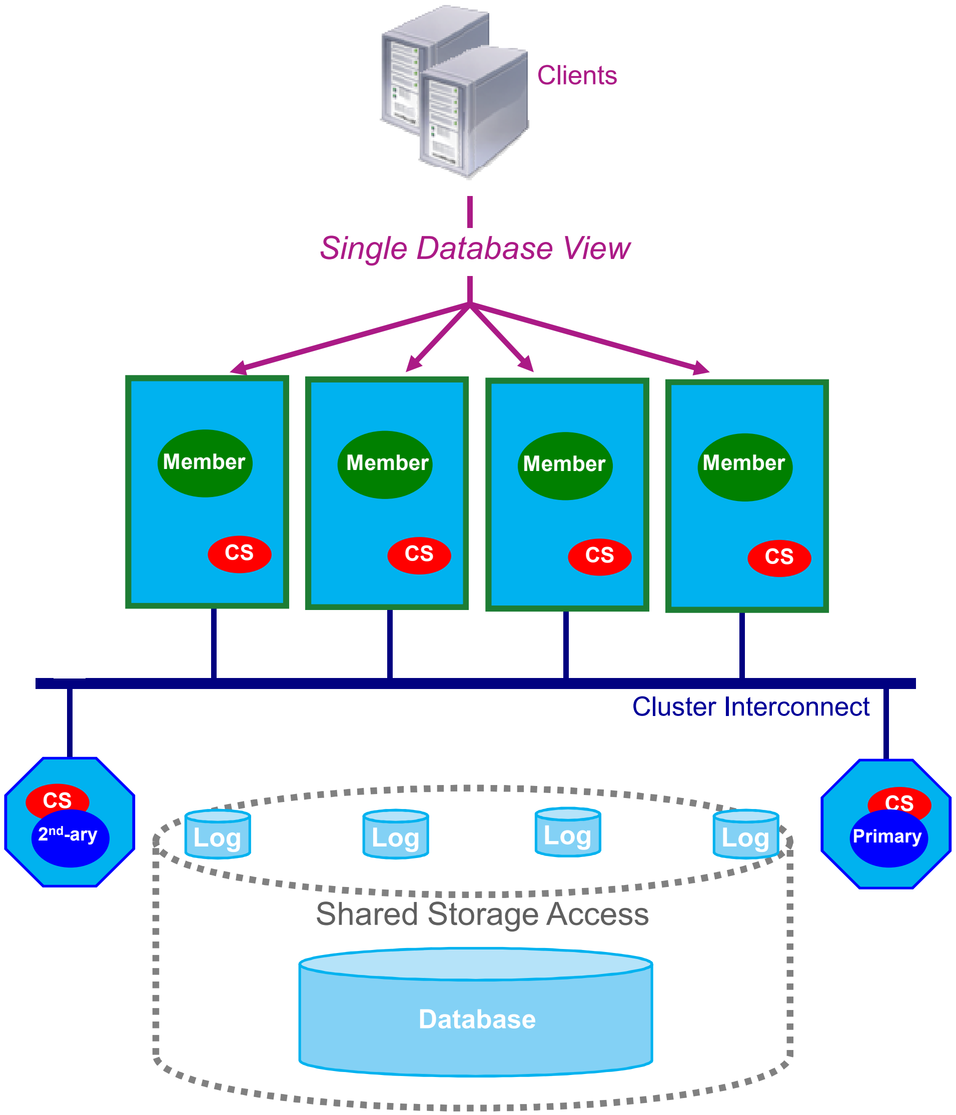
- DB2 pureScale
  - Clustered, shared-disk architecture
  - Provides improved availability, performance, and scalability
  - Complete application transparency
  - Scales to >100 members
  - Leverages z/OS cluster technology



- New DB2 10.5 pureScale enhancements
  - Rich disaster recovery capabilities with HADR
  - Rolling fix pack updates
  - Online table reorganization
  - Online add member

DB2 pureScale : Technology Overview

Leverage IBM's System z Sysplex Experience and Know-How



Clients connect anywhere,… … see single database

- Clients connect into any member
- Automatic load balancing and client reroute may change underlying physical member to which client is connected

DB2 engine runs on several host computers

- Co-operate with each other to provide coherent access to the database from any member

Integrated cluster services

- Failure detection, recovery automation, cluster file system
- In partnership with STG (GPFS,RSCT) and Tivoli (SA MP)

Low latency, high speed interconnect

- Special optimizations provide significant advantages on RDMA-capable interconnects (eg. Infiniband)

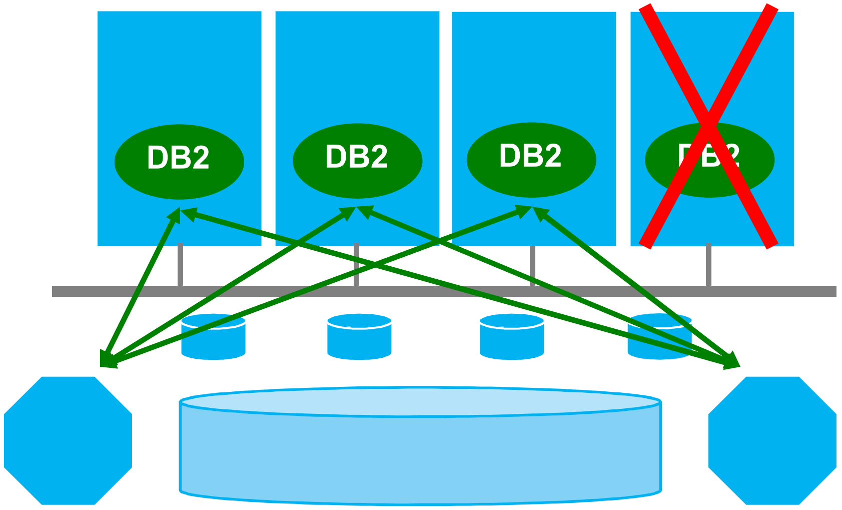
PowerHA pureScale technology from STG

- Efficient global locking and buffer management
- Synchronous duplexing to secondary ensures availability

Data sharing architecture

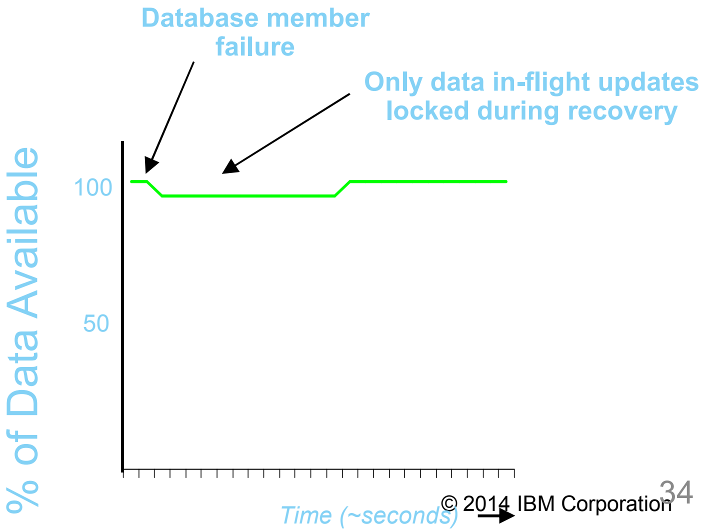
- Shared access to database
- Members write to their own logs
- Logs accessible from another host (used during recovery)

Online Recovery



- A key DB2 pureScale design point is to maximize availability **during** failure recovery processing

- When a database member fails, only data *in-flight* on the failed member remains locked during the automated recovery
  - In-flight = data being updated on the member at the time it failed



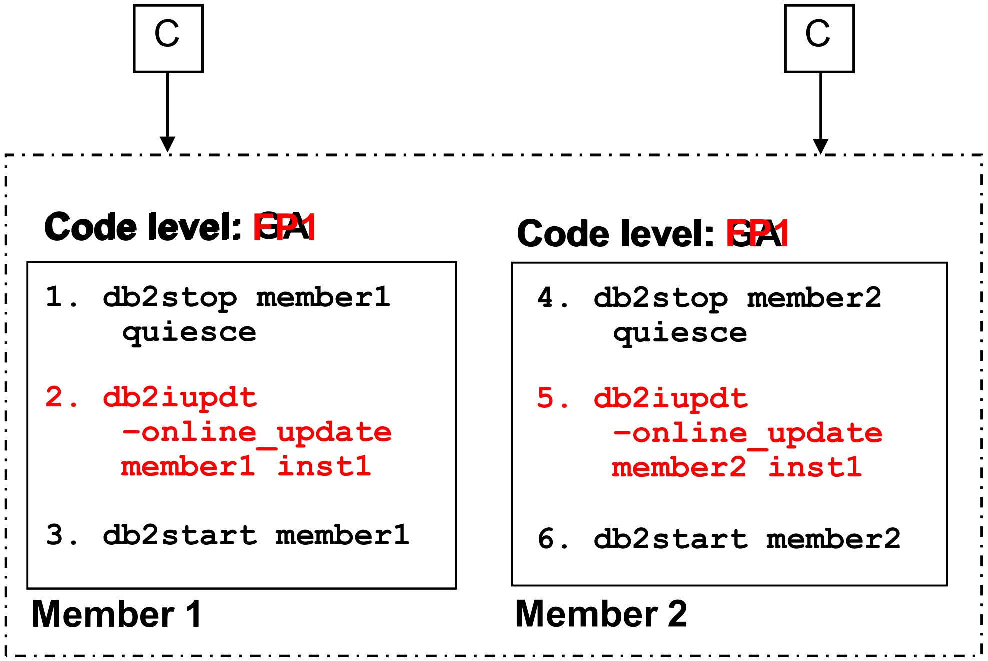
Rolling Fix Pack Updates

- DB2 pureScale fix packs can be applied in an online rolling fashion
  - Transparently install DB2 pureScale fix packs with no outage

- New options for `db2iupdt` to do to online update, do a pre-commit check, and to subsequently commit the changes

- Includes updates of CFs and members

Rolling Fix Pack Updates – Example



```
1. db2stop member1
    quiesce

2. db2iupdt
    -online_update
    member1 inst1

3. db2start member1
```

```
4. db2stop member2
    quiesce

5. db2iupdt
    -online_update
    member2 inst1

6. db2start member2
```

```
7. db2iupdt -commit_new_level inst1
```

Two member cluster (each at GA level) with clients (C) connecting into each member

1. Member 1 is quiesced – clients all move to Member 2
2. DB2 binaries updated on Member 1
3. Member 1 started again and a portion of the clients get rerouted to Member 1 to balance the workload
4. Member 2 is quiesced – clients all move to Member 1
5. DB2 binaries updated on Member 2
6. Member 2 started again and a portion of the clients get rerouted to member 2 to balance the workload

At this point, code is at FP1 level, but can't use any new FP1 features; can test stability and roll down to GA level if necessary

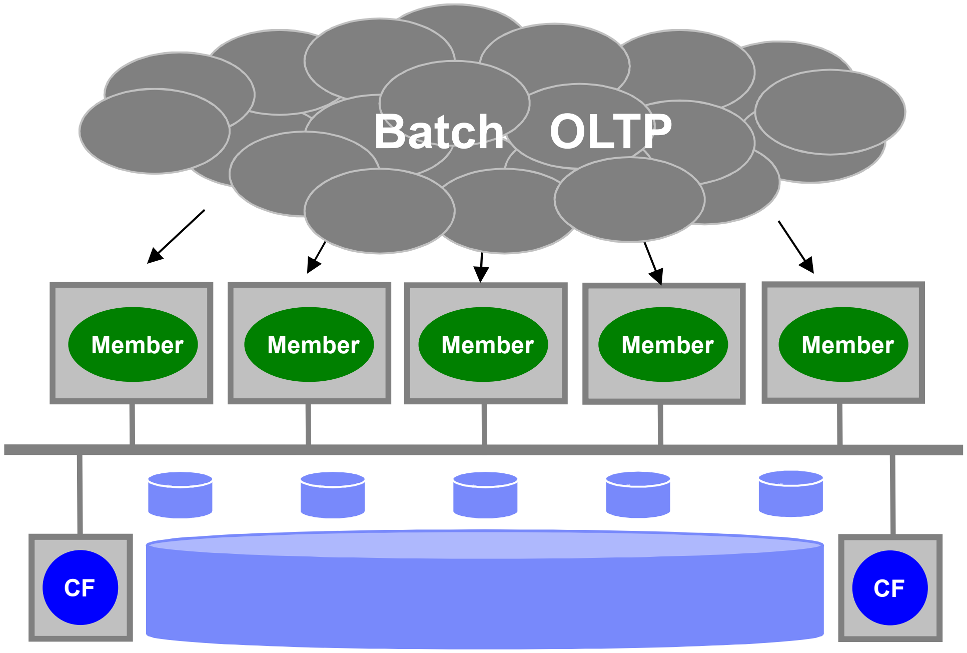
7. Updates are committed

The instance is now completely running at FP1 and new features can be used; cannot roll down to GA any longer.

Multi-Tenancy: Member Subsets

- Previously, an application/tenant could only be configured to run
  1. On one member (client affinity) or
  2. Across all members in cluster (workload balancing)

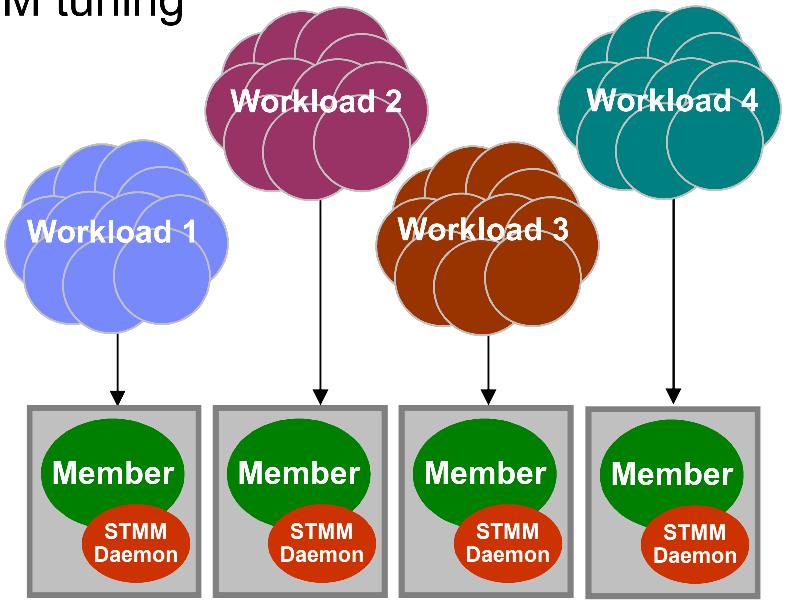
- Can now point applications to subsets of members which enables
  - Isolation of batch from transactional workloads
  - Multiple databases in a single instance to be isolated from each other



Multi-Tenancy: Self-Tuning Memory Management (STMM)

- Prior DB2 pureScale STMM design
  - Single tuning member makes local tuning decisions based on workload running on that member
    - Other member becomes tuning member in case of member failure
  - Broadcasts tuning decisions to other members
  - Works well in single homogeneous workload scenarios

- DB2 pureScale now allows per-member STMM tuning
  - Workload consolidation
  - Multi-tenancy
  - Batch workloads
  - Affinitized workloads



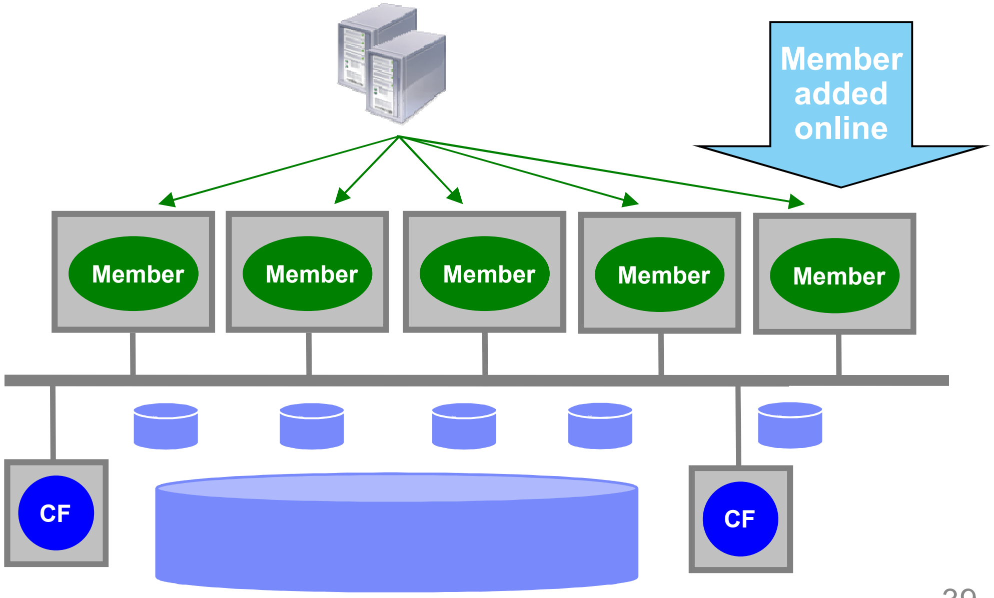
Online Add Member

- New members can be added to an instance while it is online
  - – No impact to workloads running on existing members
  - – Previously, required an outage of the entire instance to add a new member

- No change in add member command

```
db2iupdt -add -m <newHost> -mnet <networkName> <instance>
```

- Offline backup no longer needed after adding new members



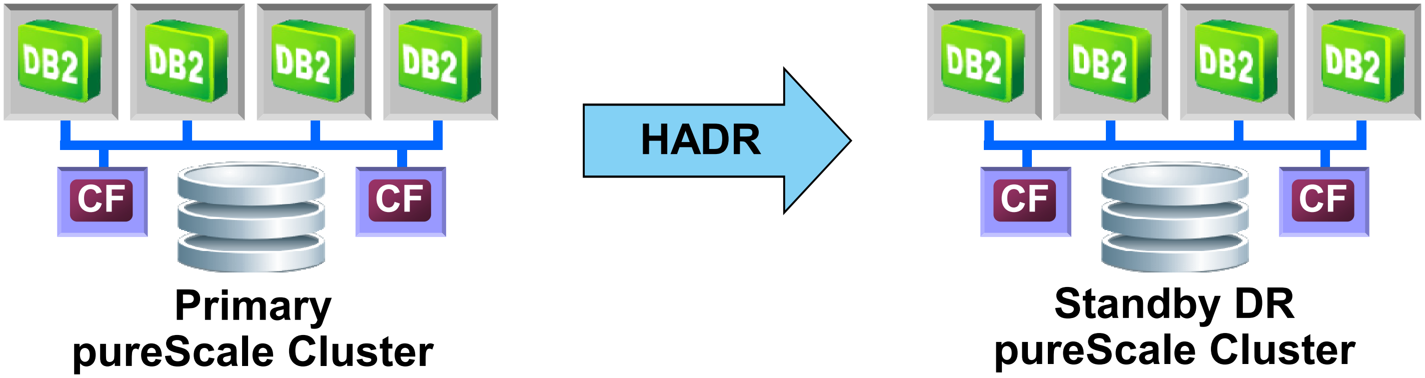
HADR in DB2 pureScale

10.5

- Integrated disaster recovery solution
  - Very simple to setup, configure, and manage

- Support includes
  - Asynchronous, super asynchronous modes
  - Time delayed apply
  - Log spooling
  - Both non-forced (role switch) and forced (failover) takeovers

- Member topology must match between primary and standby clusters
  - Different physical configuration allowed (less resources, sharing of LPAR, etc.)



10.5

pureScale HADR : Attributes

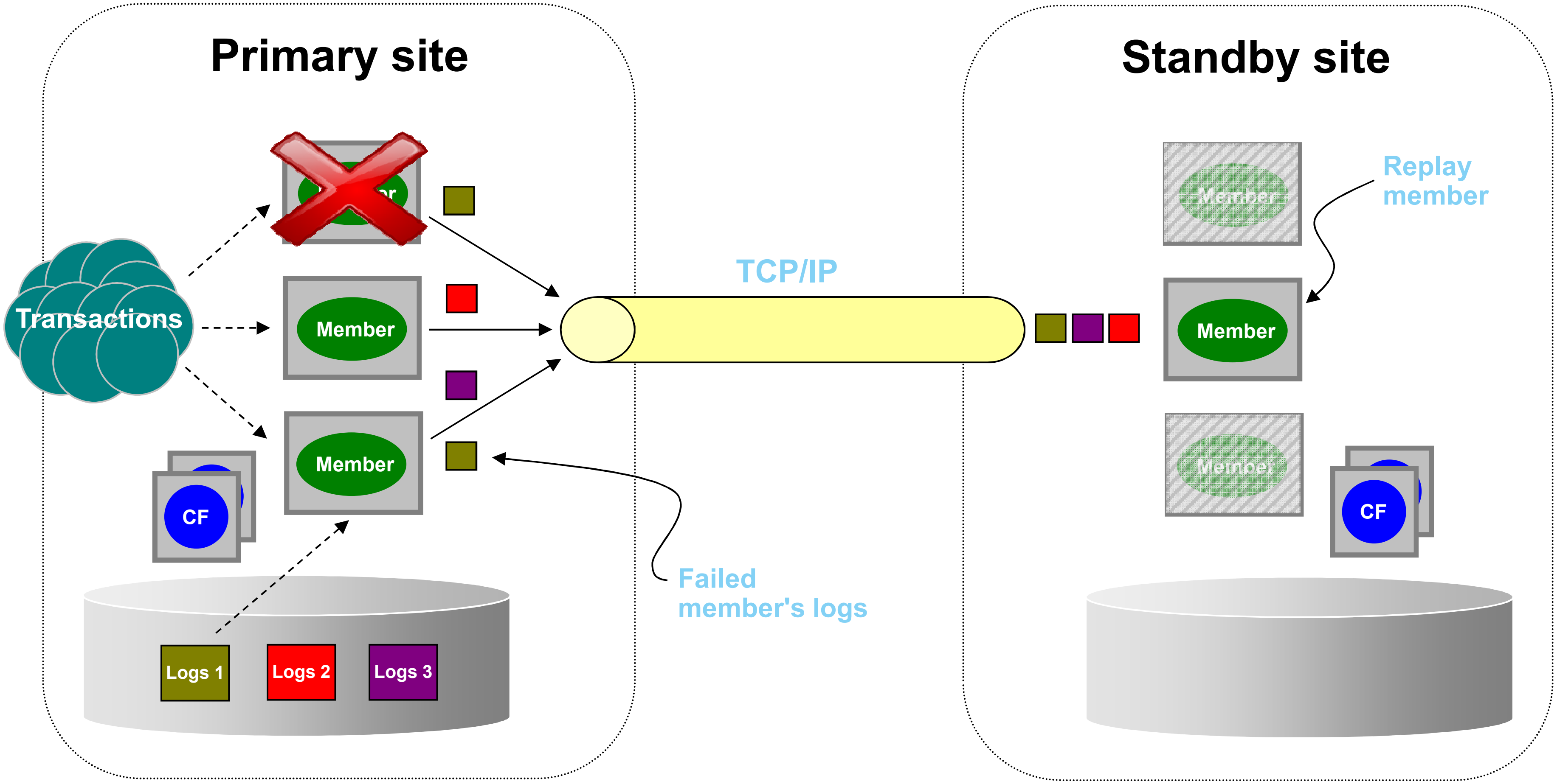
- Single system view
  - START / STOP / ACTIVATE / DEACTIVATE / TAKEOVER commands only need to be issued once, not once per member

- One member on standby is designated the ‘replay’ member
  - All primary members send log to parallel threads on a replay member on standby
  - The replay member is highly available
    - If the current replay member fails, DB2 will automatically run replay on another member

- Assisted Remote Catchup (ARCU)
  - If one primary member is not available, standby can obtain its logs via another primary member that is available

- Standby requirement
  - Must also be running with pureScale with the same number of members (they can be logical members)

HADR in DB2 pureScale: Example



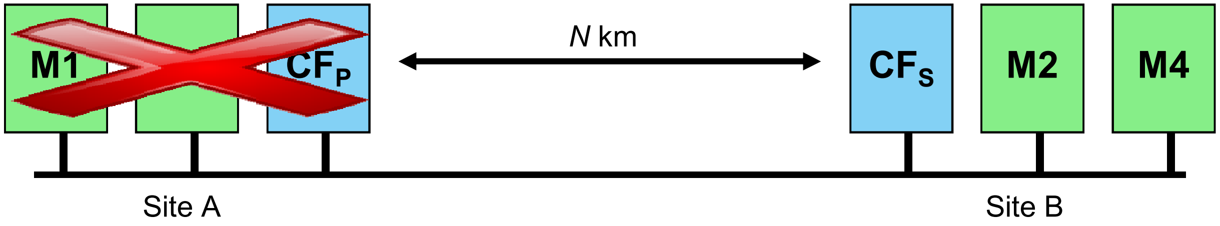
DB2 Continuous Availability Features

- There are four major features which provide relief for outages, namely:
  - HADR
  - PureScale
  - GDPC
  - Logical Replication

Geographically Dispersed pureScale Clusters (GDPC)

- A “stretch” or geographically-dispersed pureScale cluster (GDPC) spans two sites A and B at distances of tens of kilometers
  - Provides active/active access to one or more shared databases across the cluster
  - Enables a level of DR support suitable for many types of localized disasters (e.g. fires, data center power outage)

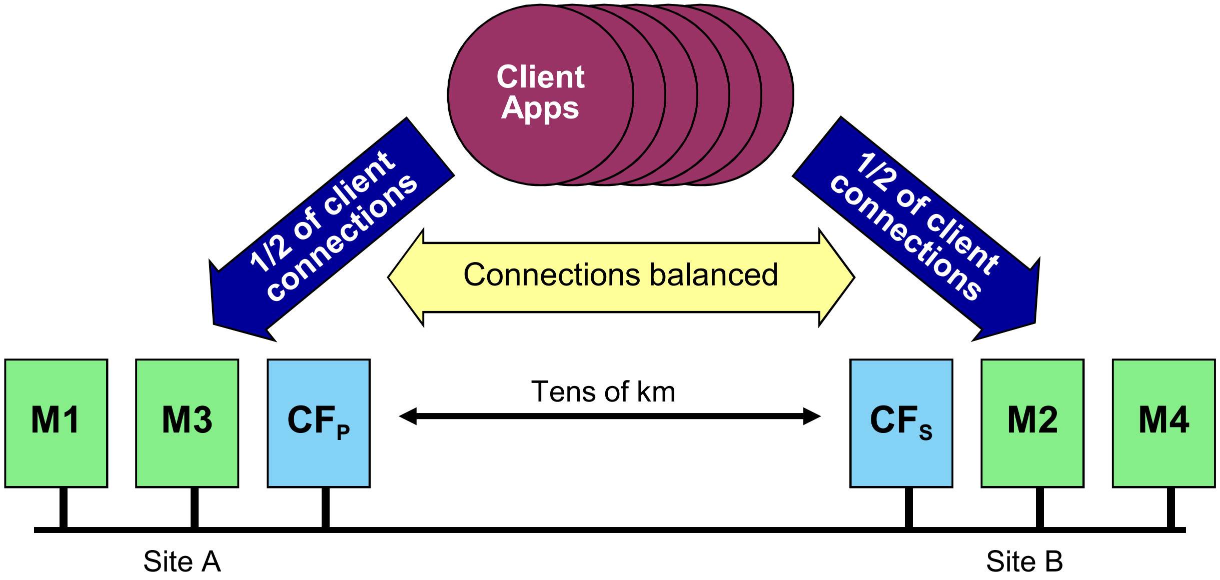
- Platforms supported
  - AIX with InfiniBand
  - RedHat Linux with 10 Gigabit Ethernet



GDPC (cont.)

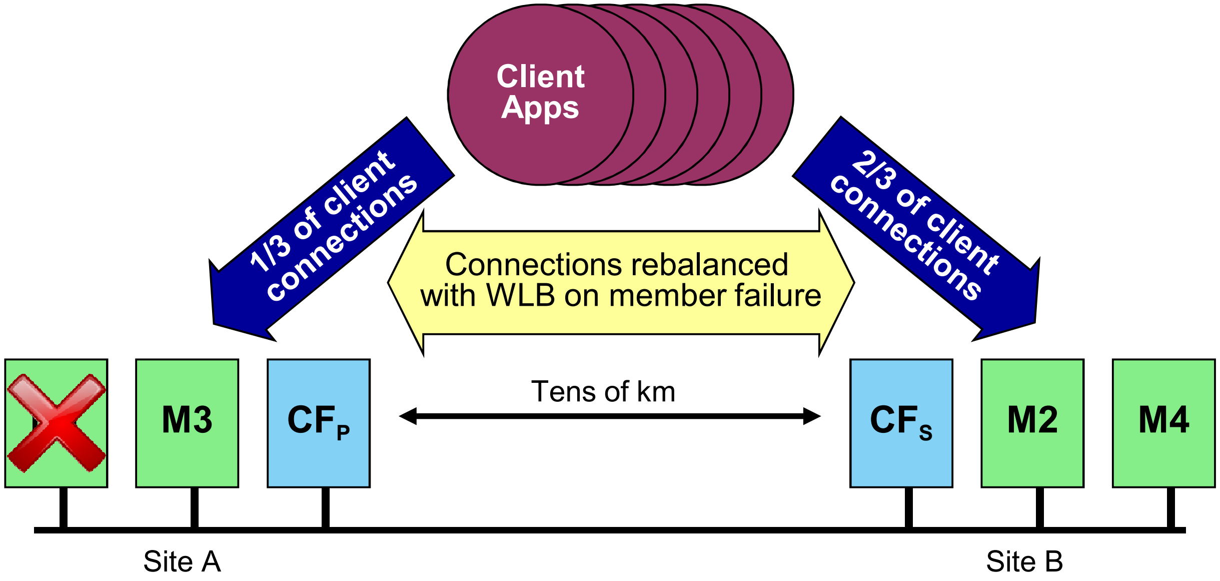
- Both sites A and B are active and available for transactions during normal operation

- On failures, client connections are automatically redirected to surviving members
  - Applies to both individual members within sites and total site failure



GDPC Member Failure

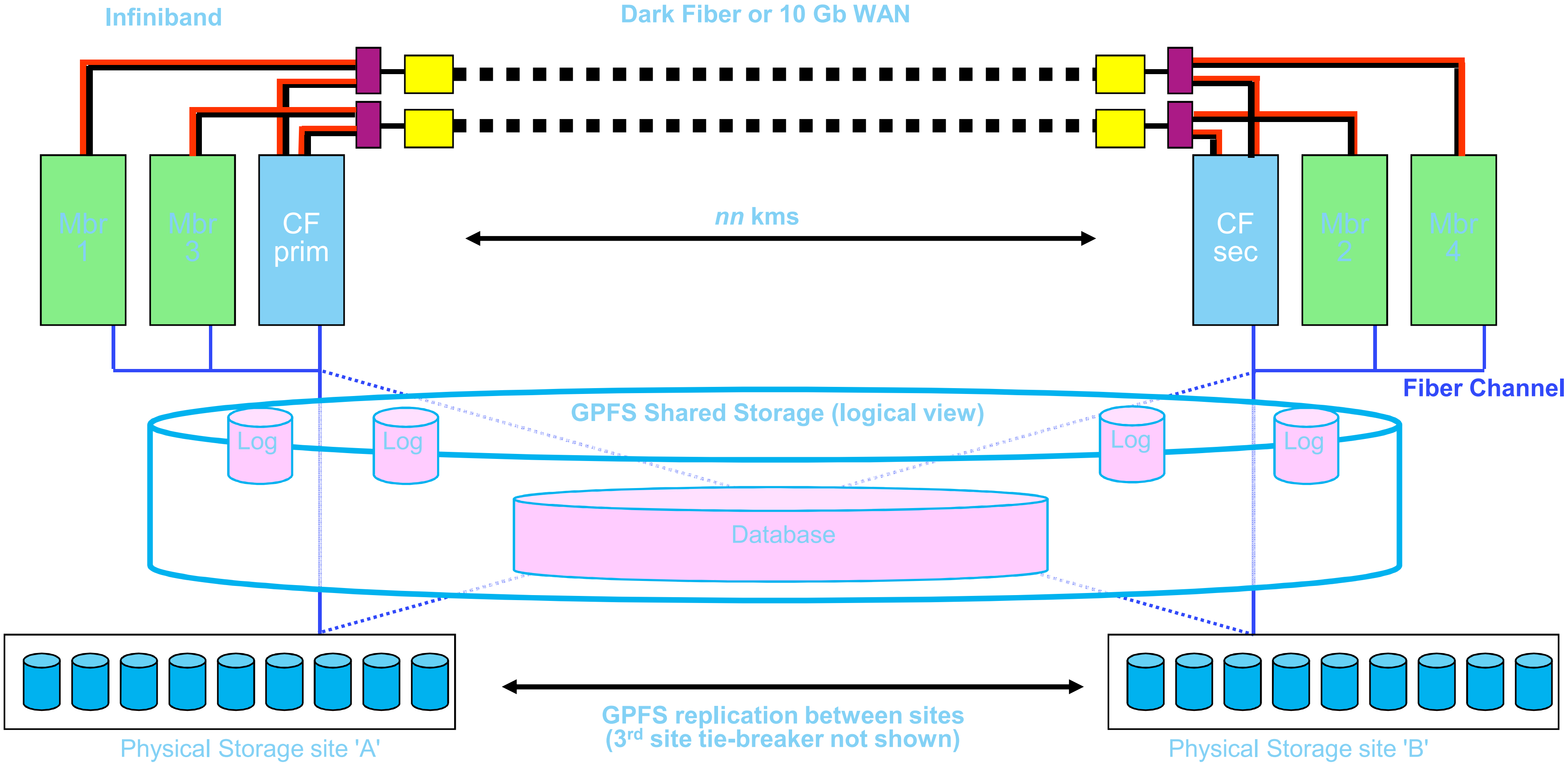
- Handled just like a single site pureScale cluster
  - –Client connections rebalanced with WLB


Client Apps
1/3 of client connections
2/3 of client connections
Connections rebalanced with WLB on member failure
M3
CF_P
Tens of km
CF_S
M2
M4
Site A
Site B

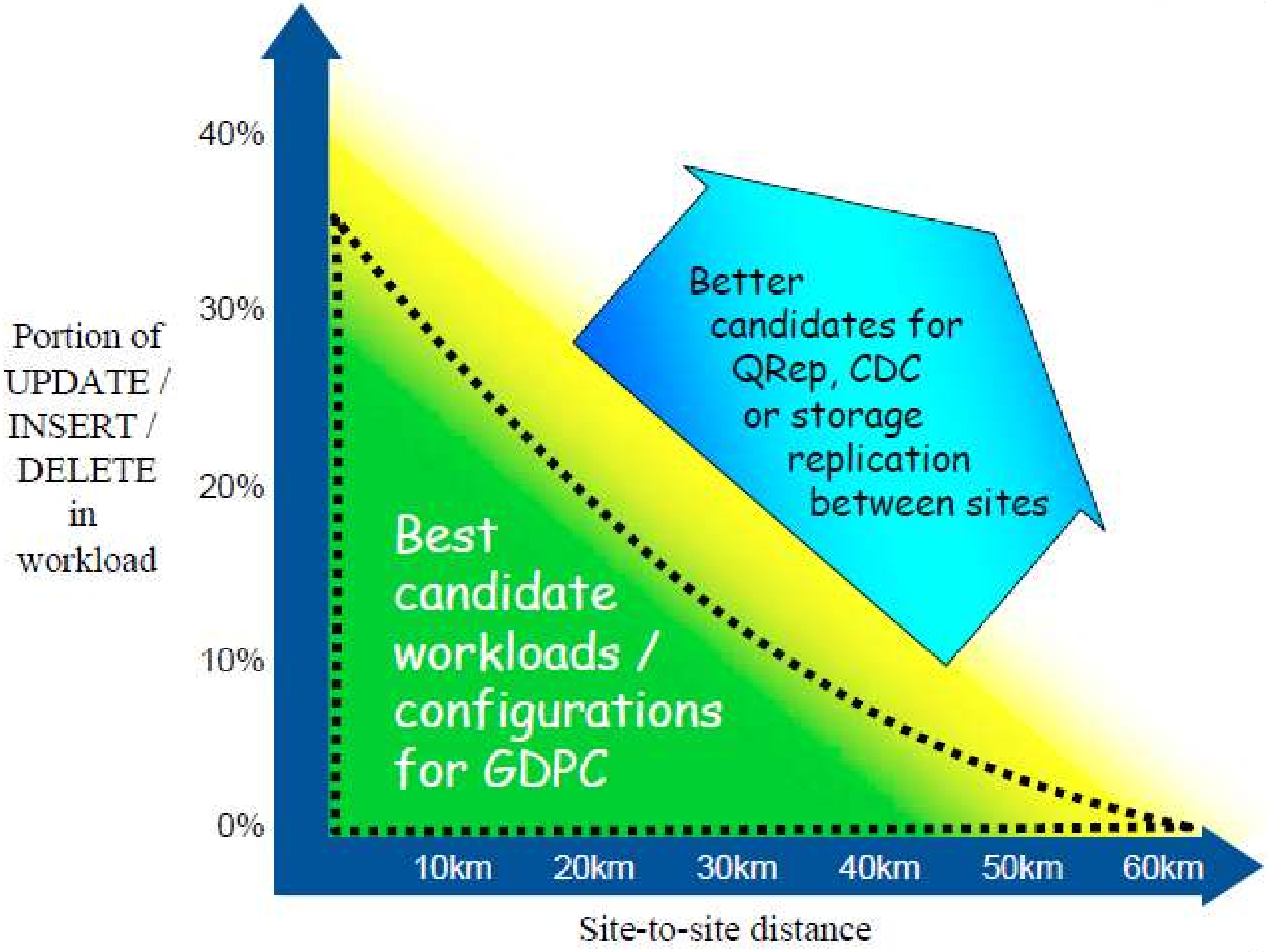
Example GDPC Configuration

🟪 IB Switch

🟨 Infiniband range extender


Infiniband
Dark Fiber or 10 Gb WAN
Mbr 1
Mbr 3
CF prim
nn kms
CF sec
Mbr 2
Mbr 4
Fiber Channel
GPFS Shared Storage (logical view)
Log
Log
Log
Log
Database
GPFS replication between sites
(3rd site tie-breaker not shown)
Physical Storage site 'A'
Physical Storage site 'B'

Suitability of GDPC

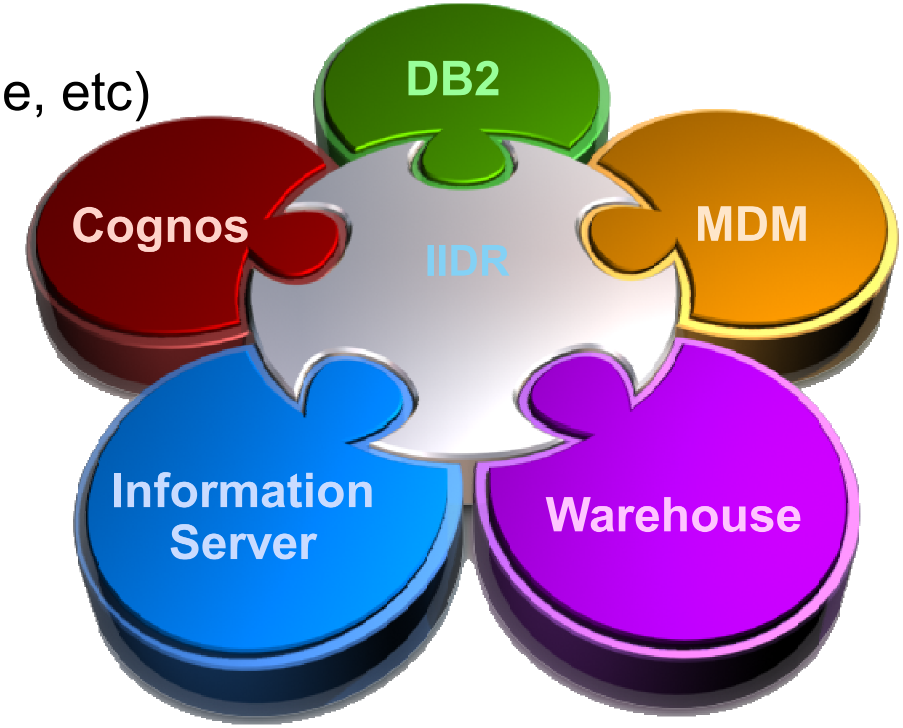


DB2 Continuous Availability Features

- There are four major features which provide relief for outages, namely:
  - HADR
  - PureScale
  - GDPC
  - Logical Replication

IBM InfoSphere Data Replication (IIDR)

- ***Used for any change data capture need***
  - High Availability and Disaster Recovery
  - Offload query and reporting workloads
  - Warehousing
  - Master data systems (MDM, etc)
  - Information Server (for DataStage, etc)

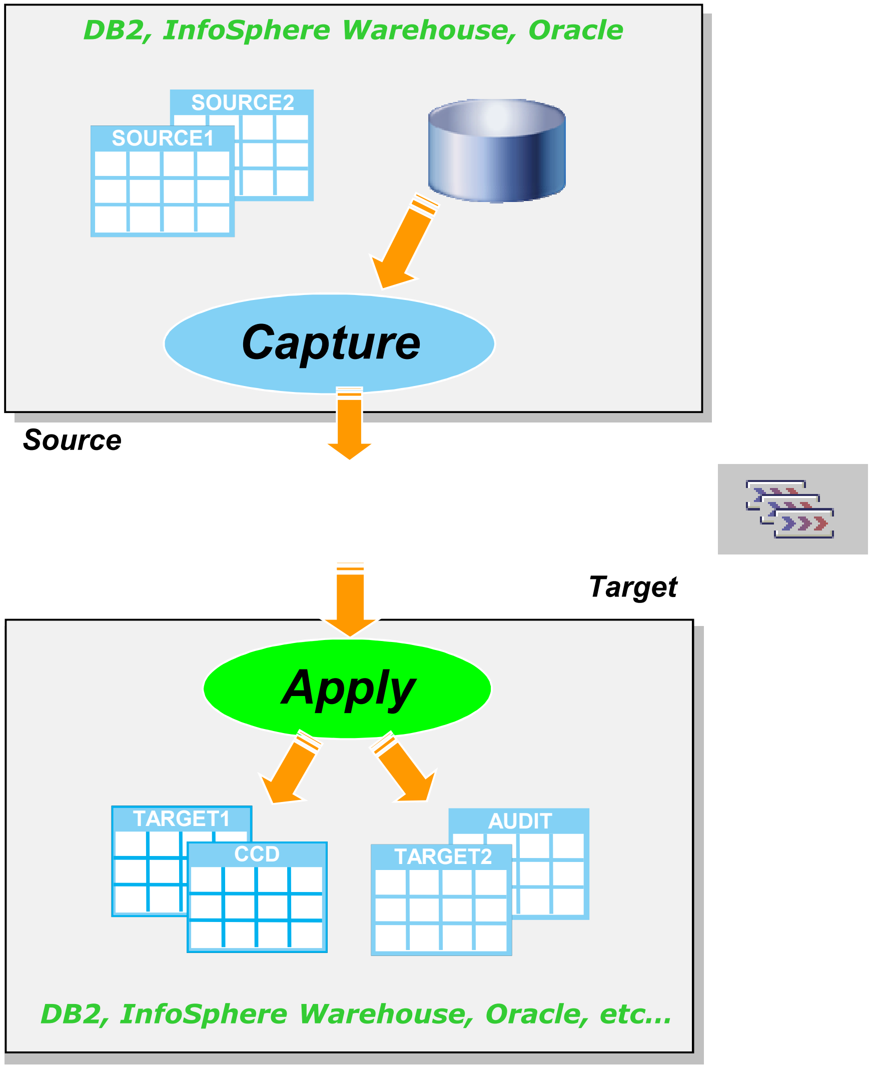


- ***Three technologies available***
  - Q Replication
  - SQL Replication
  - Change Data Capture (CDC)

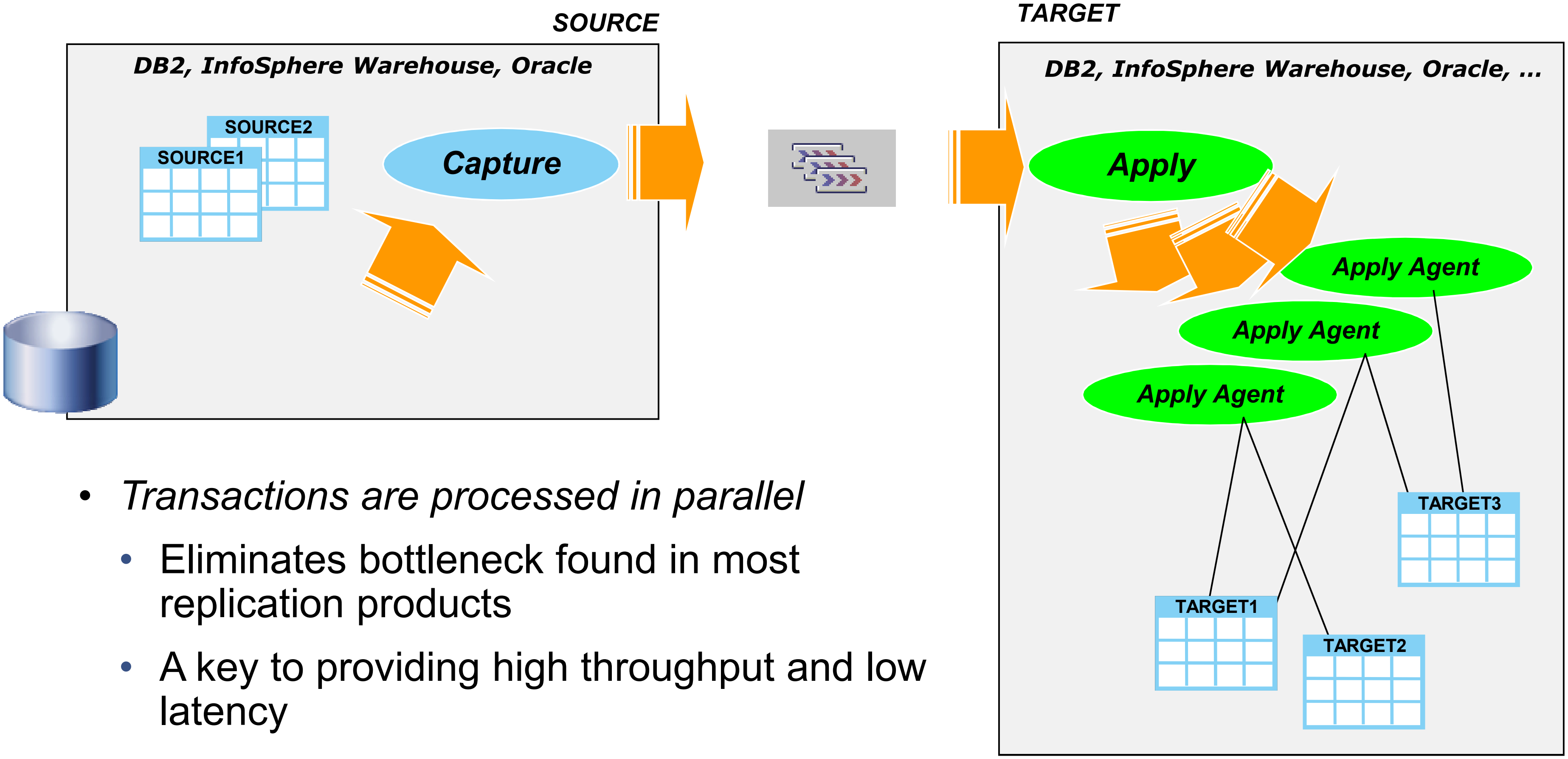
- ***Q Replication***
  - High performance
  - High availability

Q Replication

- *Change data read from database log*
- *Changes shipped directly to target*
  - No staging
- *MQ provides transport persistence*
  - A key to recoverability
- *Data then applied to target tables*
  - Highly parallel … more on next slide
- *DBA friendly*
  - All metadata in tables
  - Statistics and messages in tables


DB2, InfoSphere Warehouse, Oracle
SOURCE2
SOURCE1
Capture
Source
Target
Apply
TARGET1
CCD
AUDIT
TARGET2
DB2, InfoSphere Warehouse, Oracle, etc…

Best of Breed Performance – Parallel Apply



- *Transactions are processed in parallel*
  - Eliminates bottleneck found in most replication products
  - A key to providing high throughput and low latency

- *One of many reasons for good performance*
  - Homogeneous or heterogeneous

Can Q Replication keep up?

- Some actual numbers reported by two different customers (on z/OS)
- 111 MILLION rows replicated 1500 miles in 24 hours with average latency during the day of under 1 second and batch cycle average of 1.29 seconds

	Average	Maximum	Total	Total
Time	Latency	Latency	Transactions	Rows
Of Day	In Seconds	In Seconds	Applied	Applied
Market Hours	0.977	23.36	5,623,725	27,132,804
Non-Market Hours	1.296	321.45	4,874,703	83,214,164

- Financial customer is obtaining 600,000,000 rows replicated a day
- 1000’s customers using Q Repl in production today, across all industry sectors
  - Manufacturing, retail, financial, health, …

Why Use Q Replication for Continuous Availability?

- Advantages
  - ➢Allows the fastest switchover with transactionally consistent data
  - ➢Practically unlimited distance
  - ➢Excellent solution for scheduled outage
    - Allows flexibility of OS level, DB level, application level, data format
    - Can be easily tested and monitored
  - ➢Allows for database read or write activity on secondary
    - Secondary site may be used for other applications
    - It is the only solution for geographically dispersed updateable databases
  - ➢Can supplement other HA solutions
  - ➢No impact on application response time (capture is asynchronous)

- Disadvantages
  - ➢Asynchronous
    - Some data is left behind in a failure scenario

Change Data Capture (CDC)

- Real-time changed data capture across database systems
  - –Captures data from production systems without impacting performance
  - –Applies data to target systems in real time

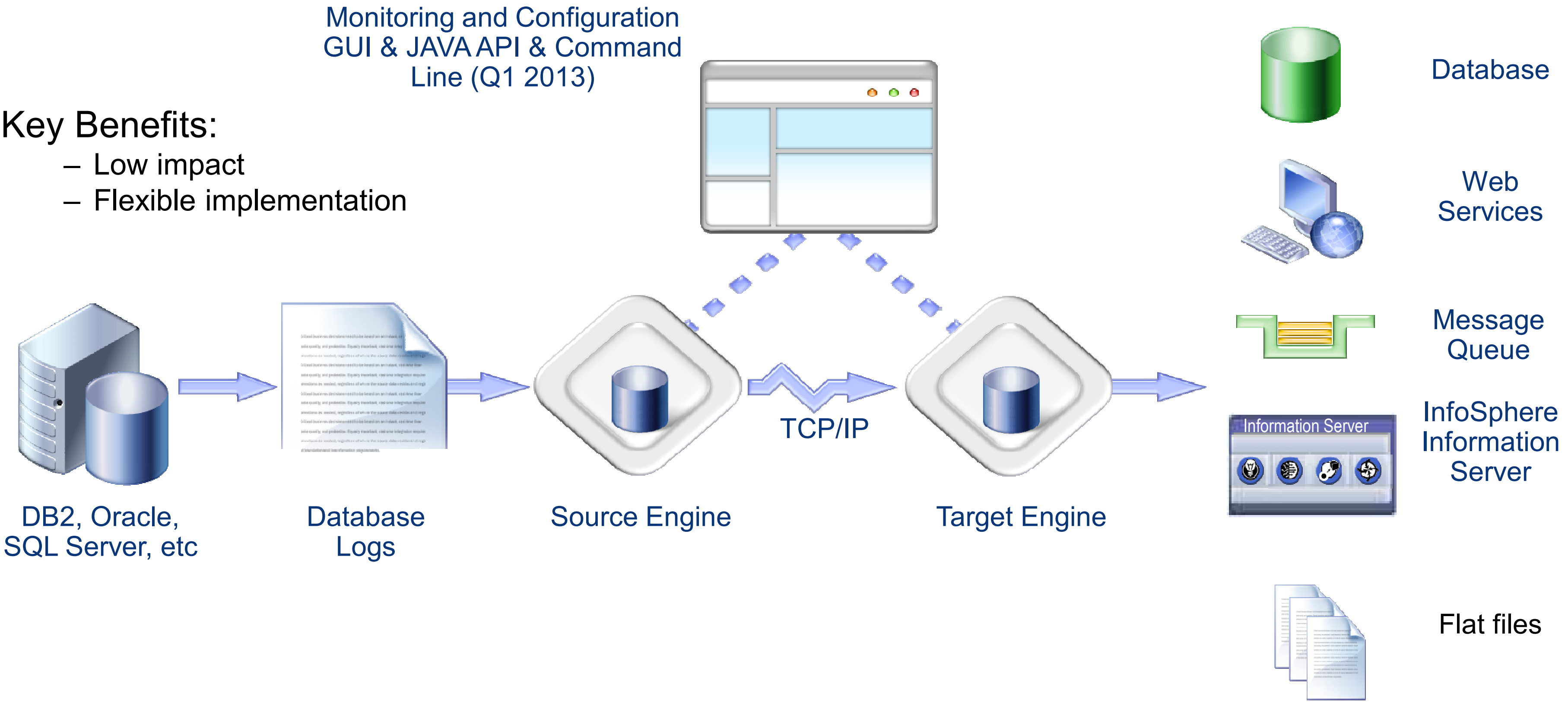
- Transforms database operations into XML documents
  - –Supports simple XML transactions

- Creates audit trails for full data traceability

Log-Based Change Data Capture

Key Benefits:

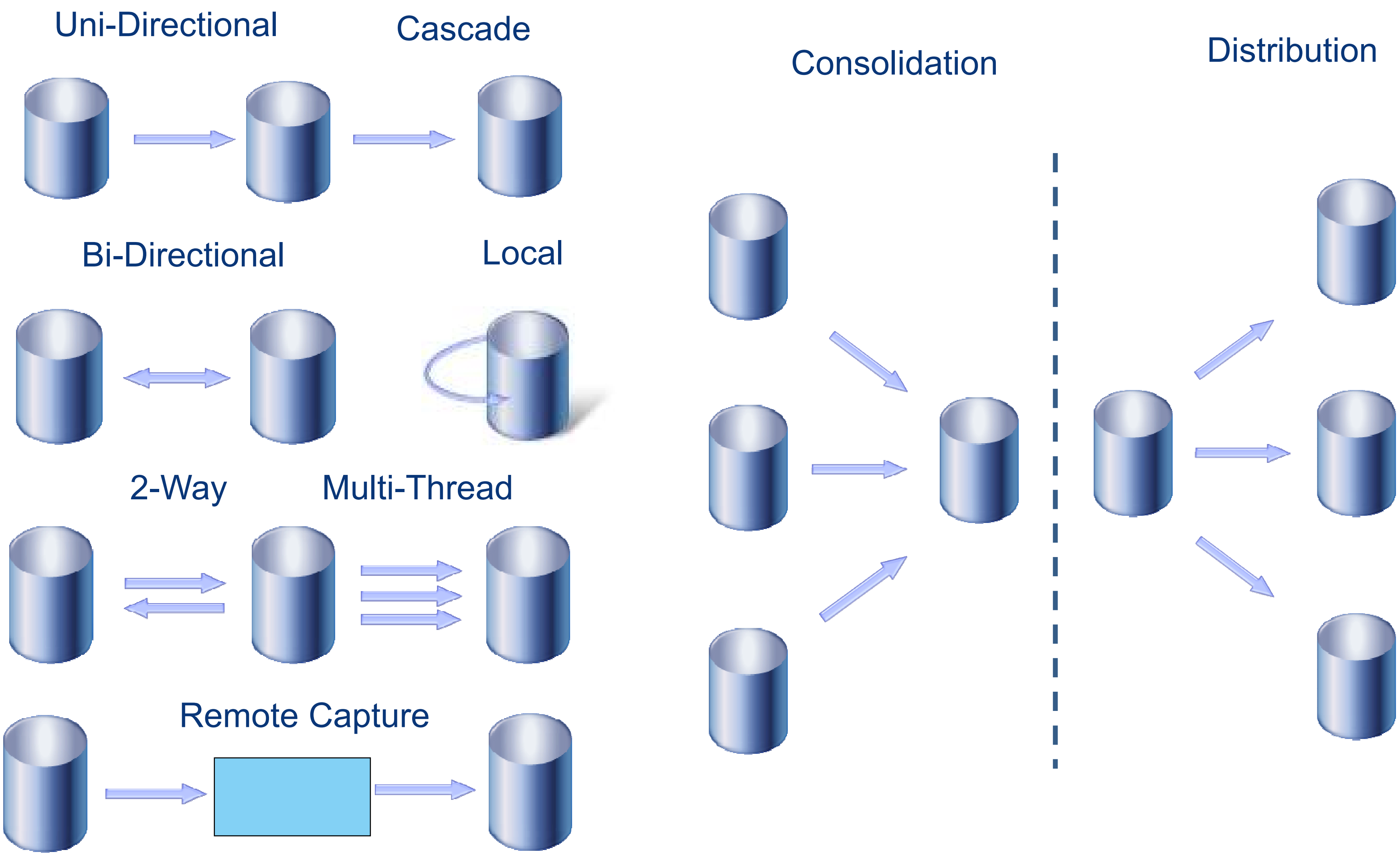
- – Low impact
- – Flexible implementation



Key Benefits:

- Low impact
- Flexible implementation
- Heterogeneous platform support
- Easy to use

Flexible Implementation Topologies



Uni-Directional

Cascade

Consolidation

Distribution

Bi-Directional

Local

2-Way

Multi-Thread

Remote Capture

Agenda

- Definitions
- Why is resilience important
- How does DB2 address these availability challenges
- Tips and Techniques
- What are real customers doing

Efficient use of storage space

- Do you really need all of this data?
  - –Archive any that is no longer required
- Keep temporary data into is own table spaces
- Load into temp tables then copy into base tables
  - –Protects base tables in case of a load failure
- Exploit DB2 features to minimize need for a reorg
  - –Time Clustered Tables (ITC)
  - –Ranges Partitioned Tables(RP)

Separation of duties

- Offload query workload from OLTP servers
  - Several large WCS customers use Q Repl to stand up a reporting server, feed from the OLTP database

Component Failure

- Redundancy
  - –Shared Disk Clustering provides resiliency for everything above the storage subsystem
  - –HADR will provide redundancy for any component
- Automation
  - –TSA/MP is provided free of charge for DB2 use only

Maintenance

- Scheduled during non-peak periods
- Avoid Reorgs Upgrade to V 10.1 / V 10.5
  - ITC tables
  - Smart data / index prefetching
  - Jump Scan
- Offload maintenance

Disaster Recovery

- DOCUMENT!
- PRACTICE, PRACTICE, PRACTICE !
- Practice what you will eventually execute in the case of a disaster

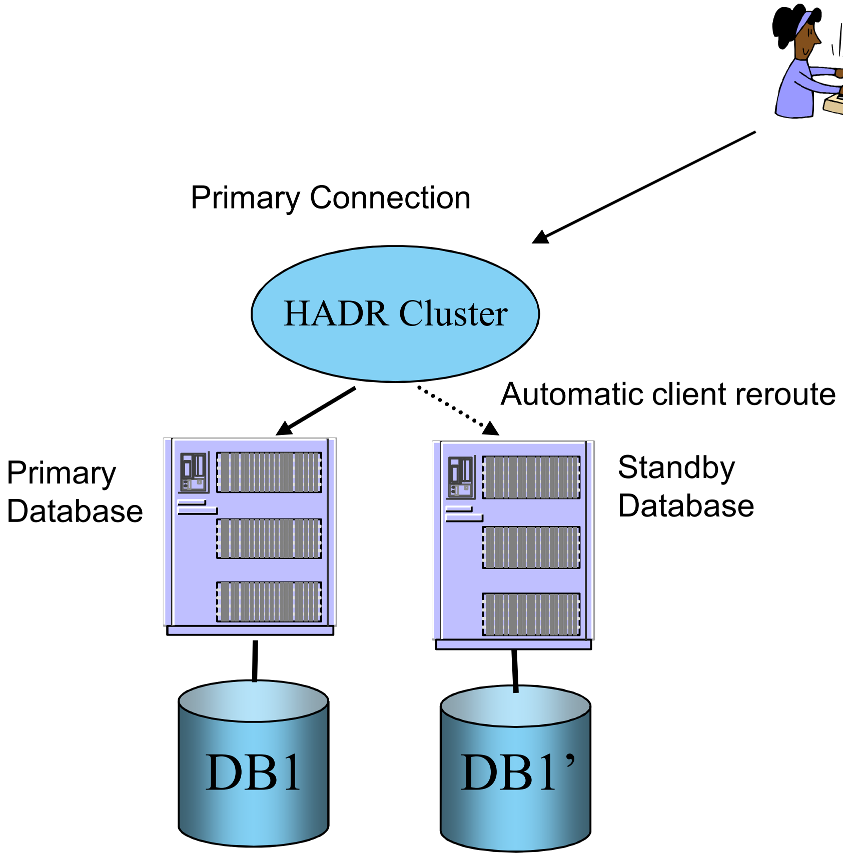
Agenda

- Definitions
- Why is resilience important
- How does DB2 address these availability challenges
- Tips and Techniques
- What are real customers doing

Building a Continuous Availability Architecture Example 1

- Customer running DB2 and WebSphere Commerce Suite for online retail processing
- Currently using HADR for Disaster Recovery
- Need a way to minimize the impact of planned database maintenance, especially reorgs.
- Business Continuity Architecture Roadmap
  - Phase 0: Currently using HADR in each DC
  - Phase 1: Implement Q Repl locally and HADR for DR, using DB2 V 10.1
  - Phase 2: Implement pureScale with Q Repl locally and HADR for DR, using DB2 V 10.5

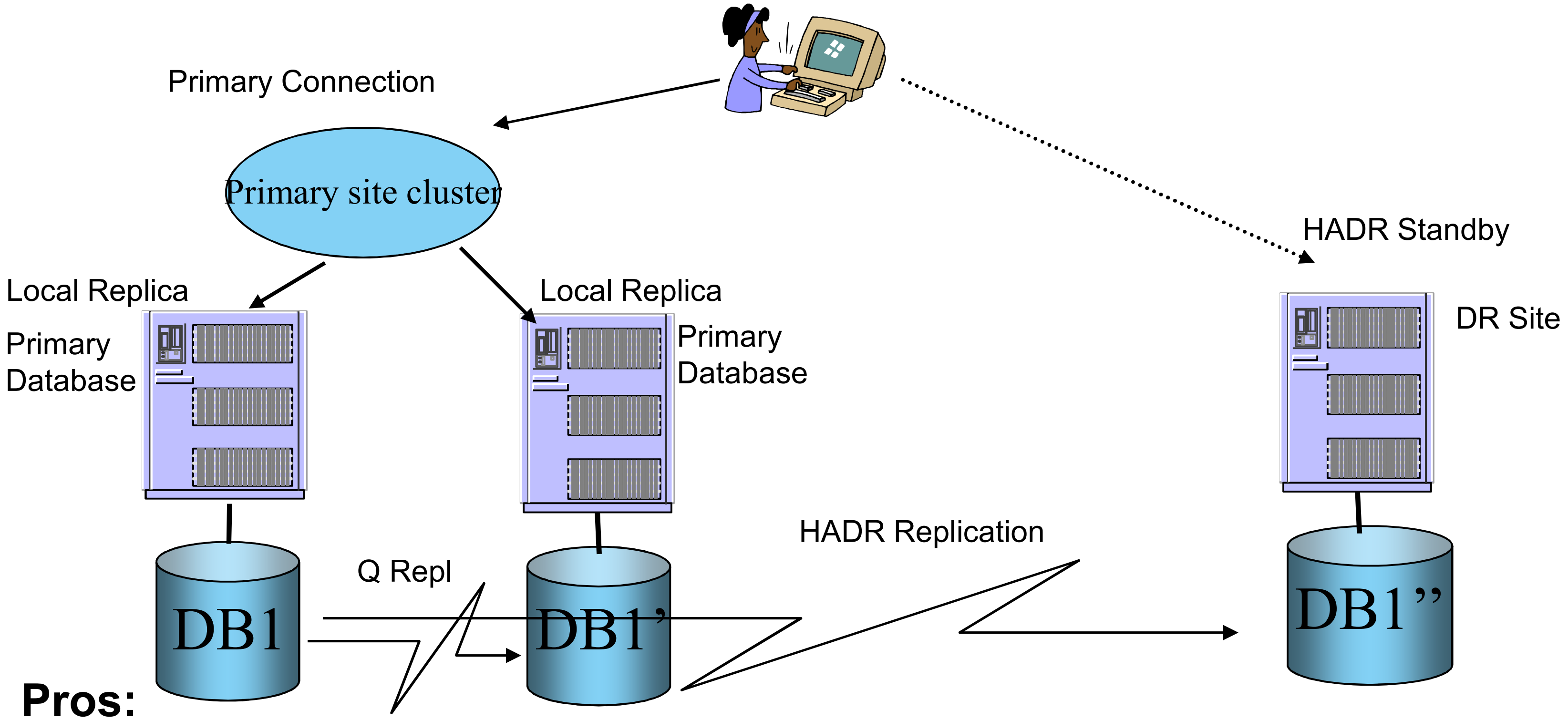
Phase 0: Current configuration



Pros:
- **Inexpensive local failover or DR solution**
- **Protection from software, server, storage or site failure**
- **Simple to setup and monitor**
- **Failover time in the range of 30 sec or less**

Cons:
- **Two full copies of the database (also a plus from a redundancy perspective)**
- **Standby database can not handle active workload**

Phase 1: Implement Q Repl locally and HADR for DR, using DB2 V 10.1



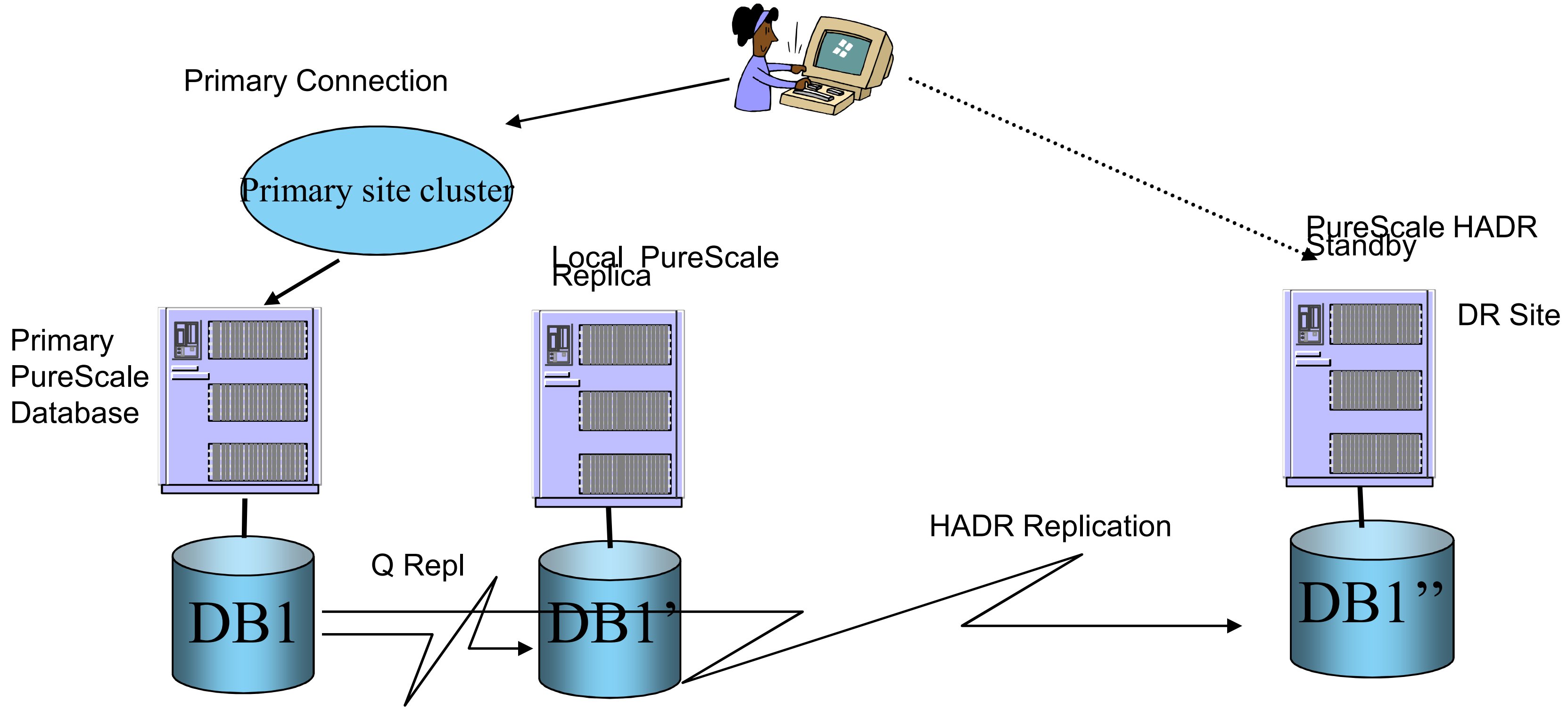
Pros:

- **All DB maintenance offloaded to local replica**
- **Protection from software, server, storage or site failure**
- **Failover time in the range of 30 sec or less local to remote DC**

Cons:

- **Three full copies of the database**
- **Must drain workload off of primary DB to move workload to local replica**

Phase 2: Implement pureScale with Q Repl locally and HADR for DR, using DB2 V 10.5



Pros:

- **All DB maintenance offloaded to local replica**
- **Protection from software, server, storage or site failure**
- **Local CA from pureScale**

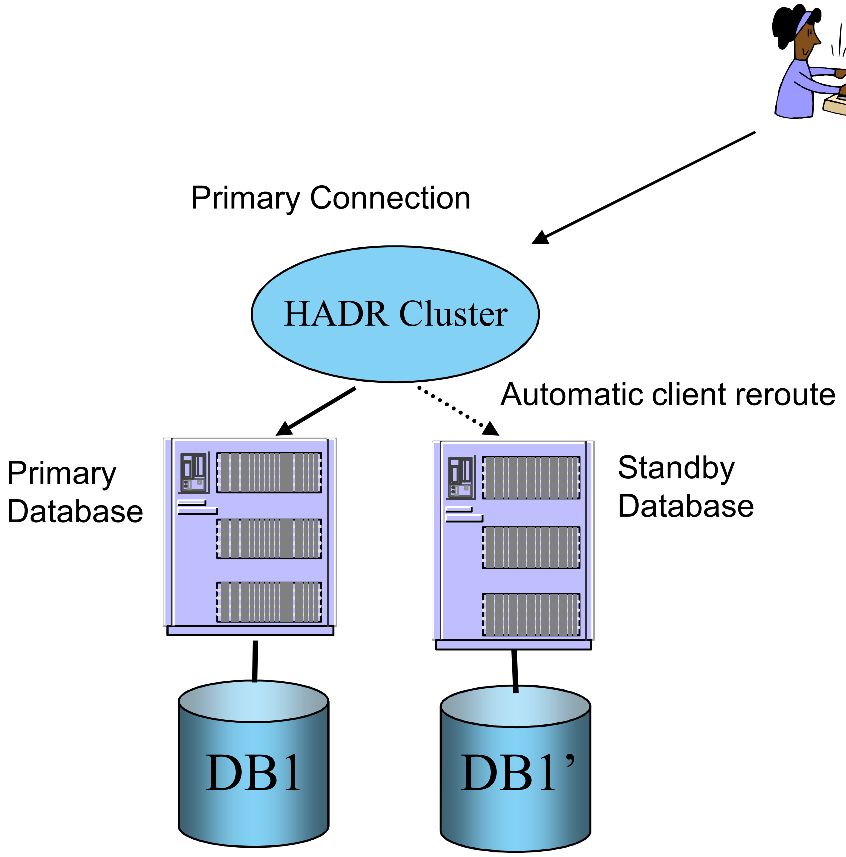
Cons:

- **Three full copies of the database**
- **Must drain workload off of primary DB to move workload to local replica**

Building a Continuous Availability Architecture Example 2

- Customer running DB2 and WebSphere Application Server online banking processing
- Currently have 2 data centers (DCs) within 1 mile
- Need to ensure continuous availability, customers have limited time to process transactions
- Business Continuity Architecture Roadmap
  - Phase 0: Currently using HADR in each DC
  - Phase 1: Implement HADR in both DCs with Q Repl between DCs
  - Phase 2: Implement pureScale GPFS between DCs
    - If co-located DC use GDPC
    - If not co-located use Q Rep

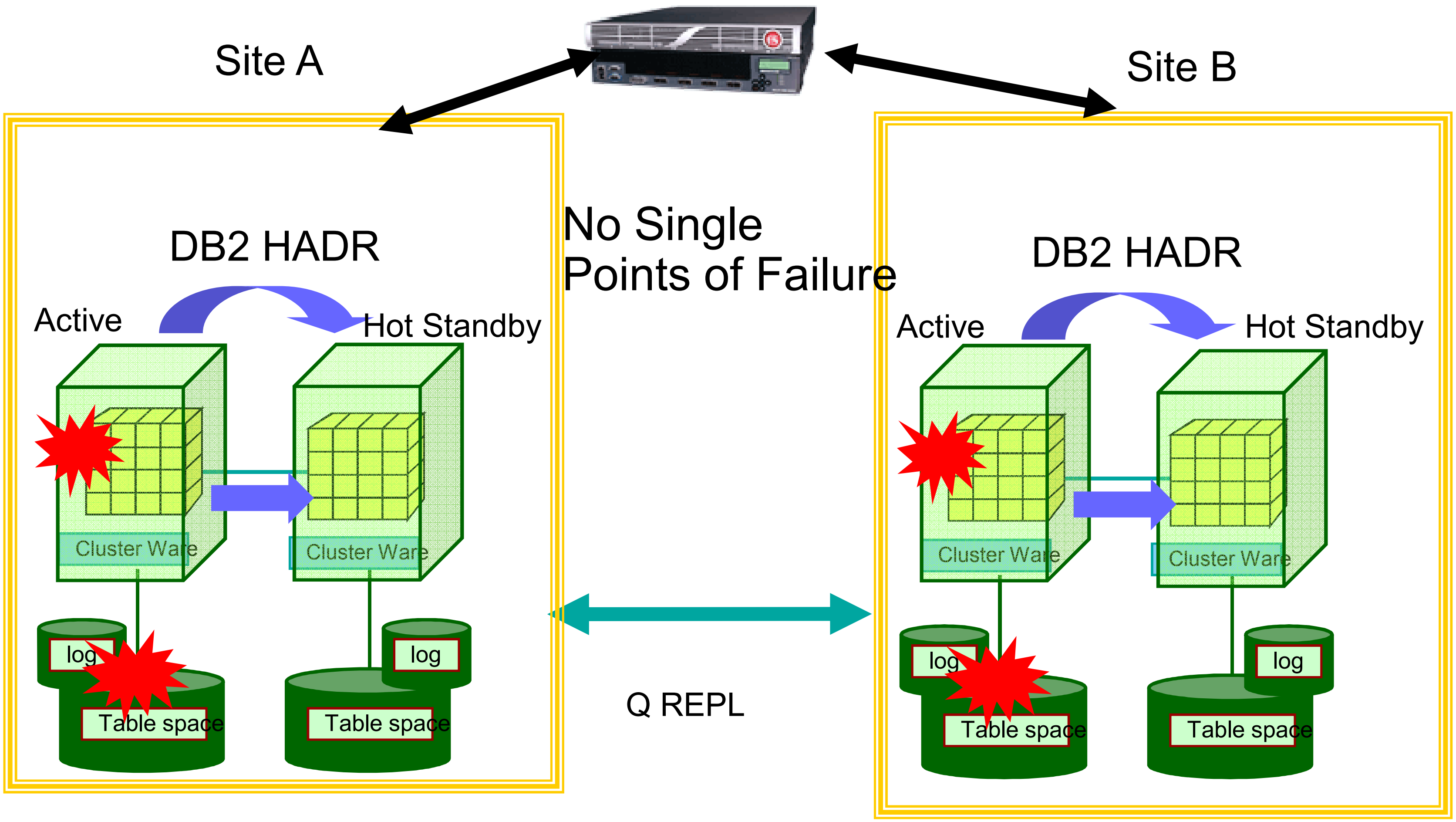
Phase 0: Current configuration



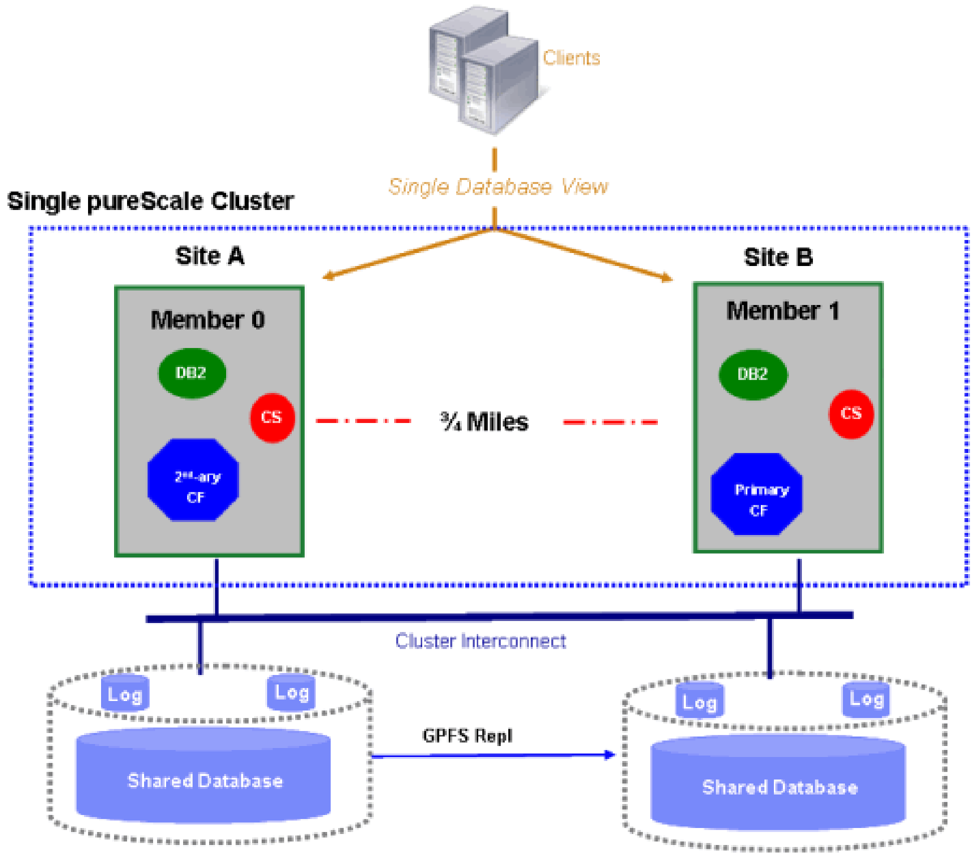
Pros:
- **Inexpensive local failover or DR solution**
- **Protection from software, server, storage or site failure**
- **Simple to setup and monitor**
- **Failover time in the range of 30 sec or less**

Cons:
- **Two full copies of the database (also a plus from a redundancy perspective)**
- **Standby database can not handle active workload**

Phase 1: Implement HADR in both DCs with Q Repl between DCs - Active / Active



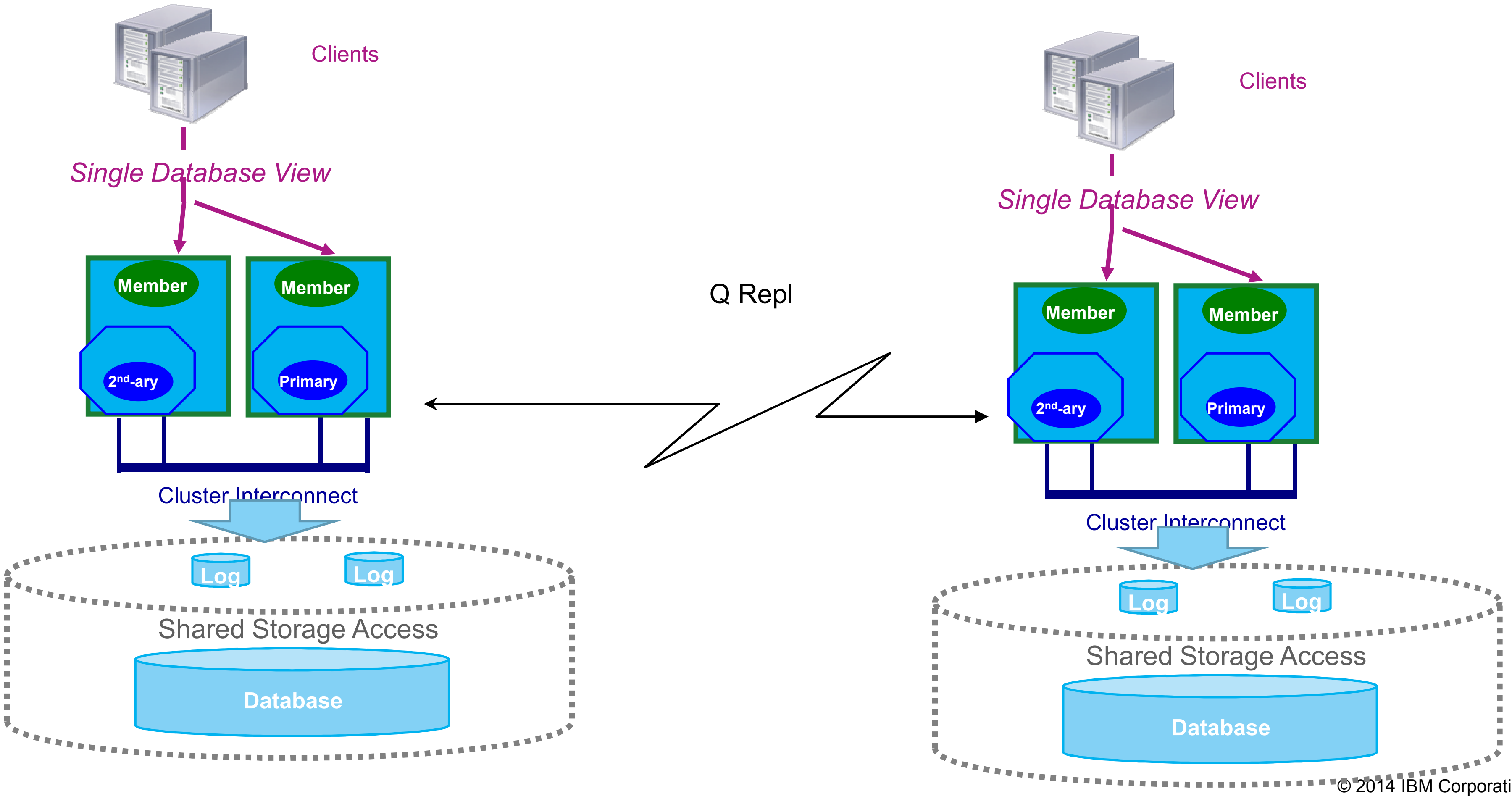
Phase 2: Implement pureScale GDPC between DCs (local DCs)



Pros:
- **Single view of the entire cluster**
- **Both sites are active**
- **Protection from software, server, storage or site failure**
- **Failover time in the range of 30 sec or less**

Cons:
- **Limited to no more than 100 km**
- **Requires high speed interconnect between the sites**

Phase 2+: Implement pureScale with Q Repl between DCs (non-collocated DCs) Active/Active



Thank
You

Dale McInnis

IBM Canada Ltd.

dmcinnis@ca.ibm.com

Title: Continuous Availability with DB2