ibm.com/db2/labchats

Data Management

DB2 High Availability

Sept 29, 2010

ibm.com/db2/labchats

> Executive's Message

Sal Vella

Vice President, Development,
Distributed Data Servers and Data Warehousing

IBM

> Featured Speaker

Dale McInnis

Availability Architect,
DB2 for Linux, UNIX, and Windows

IBM

Agenda

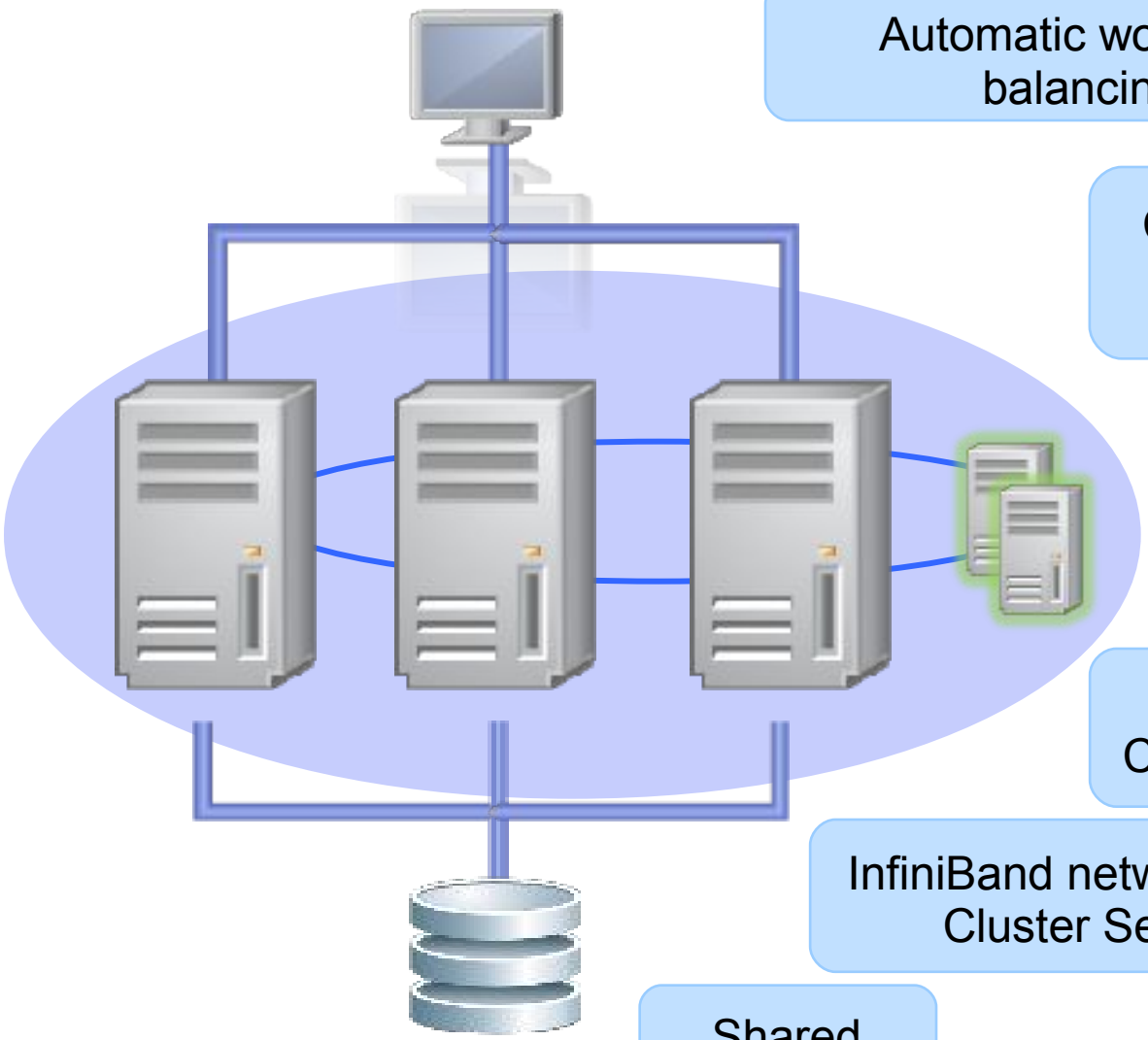
- **High Availability (HA) Options Overview**
  - Purescale, Integrated TSA, HADR
- **Disaster Recovery (DR) Options Overview**
  - HADR, Q Repl, Storage Replication, Dual ETL, Log Shipping
- **TSA Integration**
  - Shared disk failover (ESE or DPF), HADR Takeover automation
- **HADR - Overview and RoS**
- **Trends - Active/Active for DR**

DB2 pureScale

- Unlimited Capacity
  - Buy only what you need, add capacity as your needs grow
- Application Transparency
  - Avoid the risk and cost of application changes
- Continuous Availability
  - Deliver uninterrupted access to your data with consistent performance

Learning from the undisputed Gold Standard... System z

DB2 pureScale Architecture



Automatic workload balancing

Cluster of DB2 nodes running on Power servers

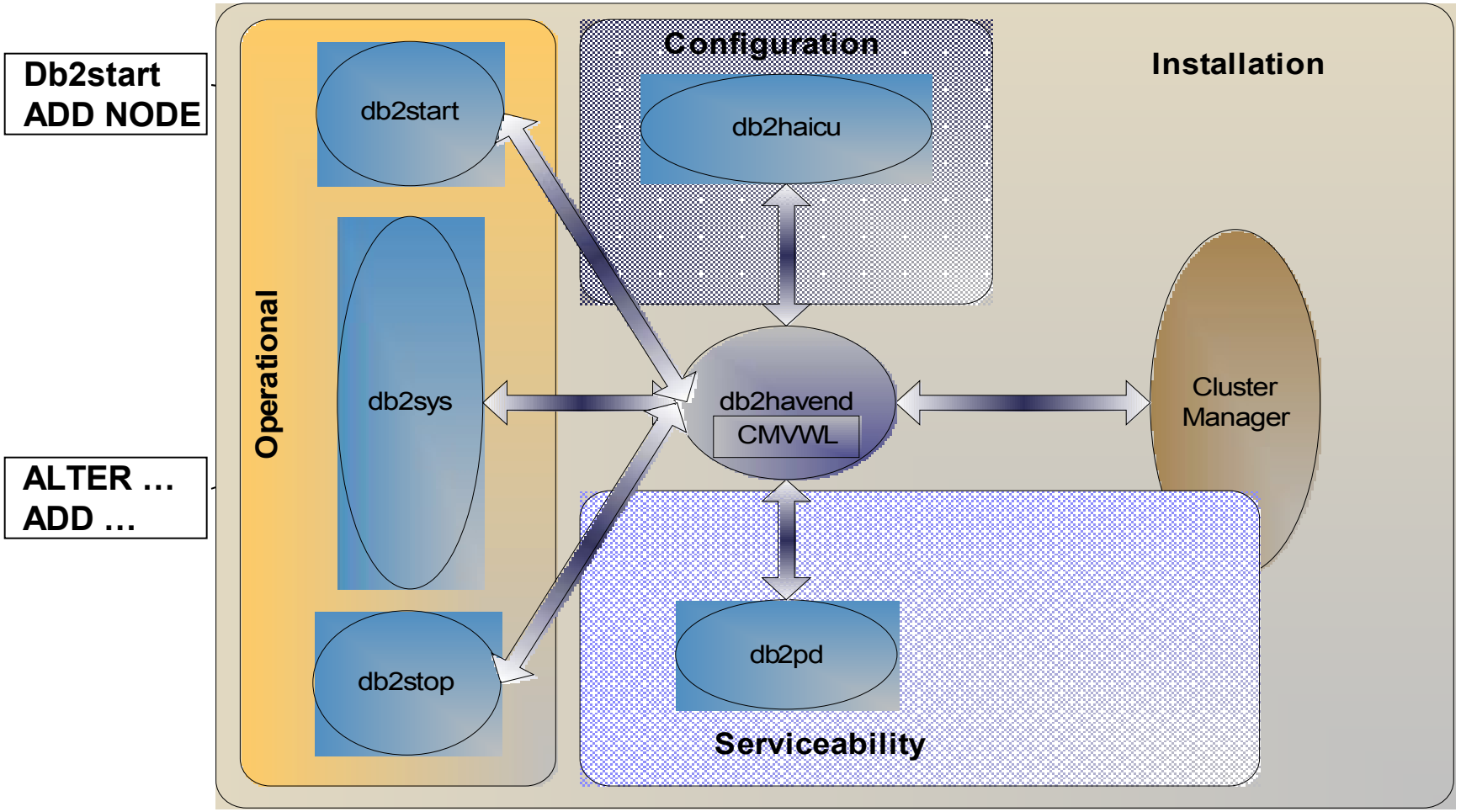
Leverages the global lock and memory manager technology from z/OS

Integrated Cluster Manager

InfiniBand network & DB2 Cluster Services

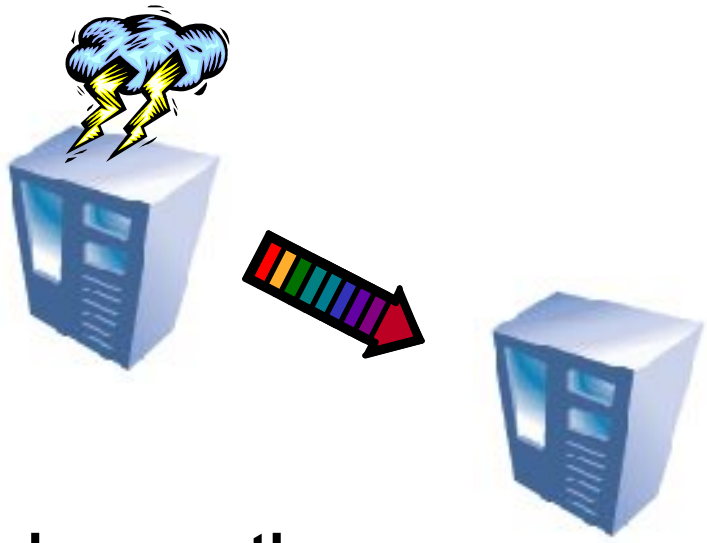
Shared Data

Integrated TSA Architecture



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

High Availability Options

Method	Pros	Cons
Use of a Cluster Manager See white papers at: http://www.ibm.com/software/data/pubs/papers More on this later	•Exploit existing CM infrastructure •Solution available from multiple vendors •Striving toward industry standard API support	•May not be able to exploit redundant hardware •Standby system is "cold" •Applications will suffer from a brown-out period until the memory has been populated •Integration is through scripts
Use of a Hot Standby, e.g. HADR More on this later	•Support for no transaction loss •Minimal impact to production system •No "brown-out" period during fail-over •Support for fail-back and re-integration	•Standby not available for use while in Rollforward mode •Standby needs to be physically and logically identical •Some administrative actions not reflected on standby (eg. NOT LOGGED operations)
Use of Replication More on this later	•Can read (and write) standby •Standby need not be physically and logically identical •Can choose to replicate only critical tables	•Transaction loss a possibility •Extra cycles on production database for transaction capture •Some administrative actions not reflected on standby (eg. NOT LOGGED operations)

Agenda

- **High Availablity Options Overview**
  - Purescale, Integrated TSA, HADR
- **Disaster Recovery Options Overview**
  - HADR, Q Repl, Storage Replication, Dual ETL, Log Shipping
- **TSA Integration**
  - Shared disk failover (ESE or DPF), HADR Takeover automation
- **HADR - Overview and RoS**
- **Trends - Active/Active for DR**

Business Drivers for a Disaster Recover
Functionality by Service Level

Business Availability Definition	Service Level	Level of Sophistication	Recovery Time	Mechanism
Tier 3 Data	Disaster Recovery	Low	Hours or days	Backups
Tier 2 Data	Active / Passive	Medium	Minutes to hours	Dual Loads (+ log shipping), or Storage WAN mirroring
Tier 1 Data	Dual Active	High	Seconds	Dual loads plus Q-Based replication

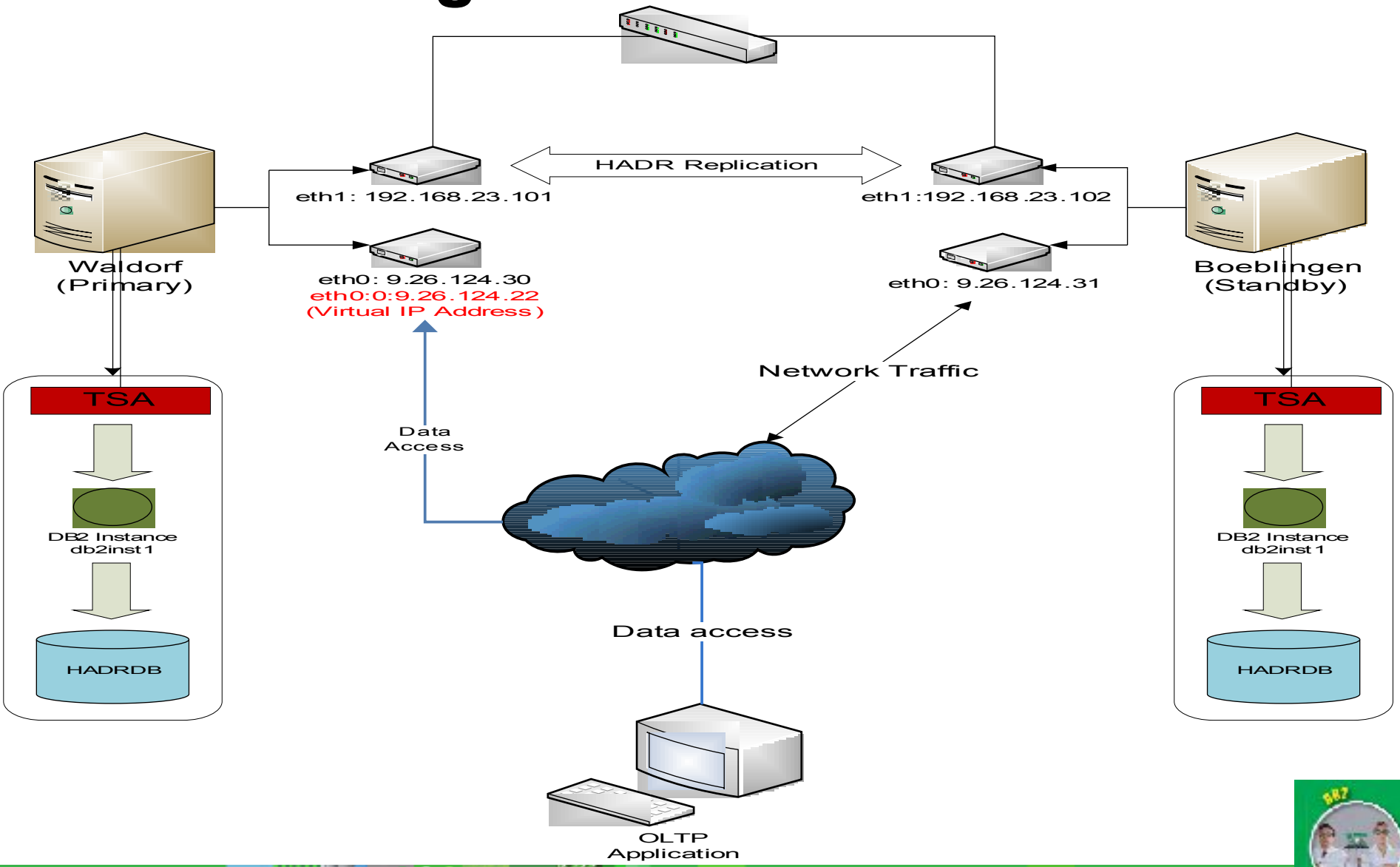
Disaster Recovery Options

Method	Pros	Cons
Transport of database backups	•Low cost •Recovery to time of last backup	•Lose all transactions since last backup
Transport of database backups and archived logs (physical/network)	•Recovery to time of last archived log •Low cost	•Lose all transactions in the active logs •Longer recovery time (database restore followed by roll-forward through all the logs)
Standby Database via Log Shipping	•Support for no transaction loss •Minimal impact to production system	•Standby not available for use while in Rollforward mode •Standby needs to be physically and logically identical •Some administrative actions not reflected on standby (eg. NOT LOGGED operations)

Disaster Recovery Options Con't

Method	Pros	Cons
Standby Database via Q Replication	•Can read (and write) standby (Active/Active) •Standby need not be physically and logically identical •Can choose to replicate only critical tables	•Transaction loss a possibility •Extra cycles on production database for transaction capture •Some administrative actions not reflected on standby (eg. NOT LOGGED operations) •DDL is not supprted by Q Repl
Synchronous mirroring of all data and log disks e.g. Metro Mirror	•No transaction loss •All changes to the database (including administrative) are replicated •Shortest restart time	•No access to the mirrored disks until the relationship with the source is broken •Performance impact of synchronous mirroring to a geographically remote site •High price (software/hardware/network)
Hot Standby, such as HADR with Reads on the standby	•Support for no transaction loss •Minimal impact to production system •No "brown-out" period during fail-over •Support for fail-back and re-integration •Can read the standby (Active/Active)	Standby needs to be physically and logically identical •Some administrative actions not reflected on standby (eg. NOT LOGGED operations)

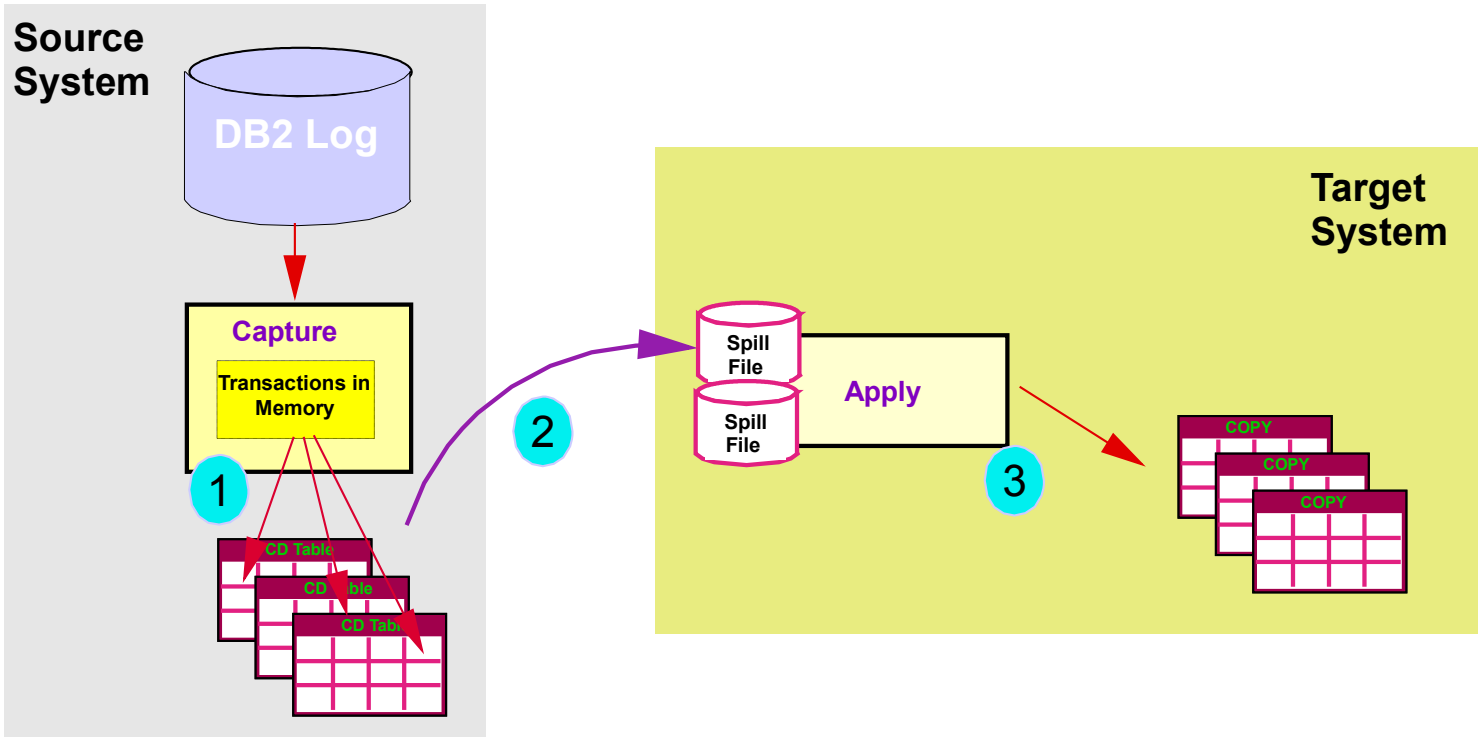
HADR Configuration



Possible Uses of Replication Technologies

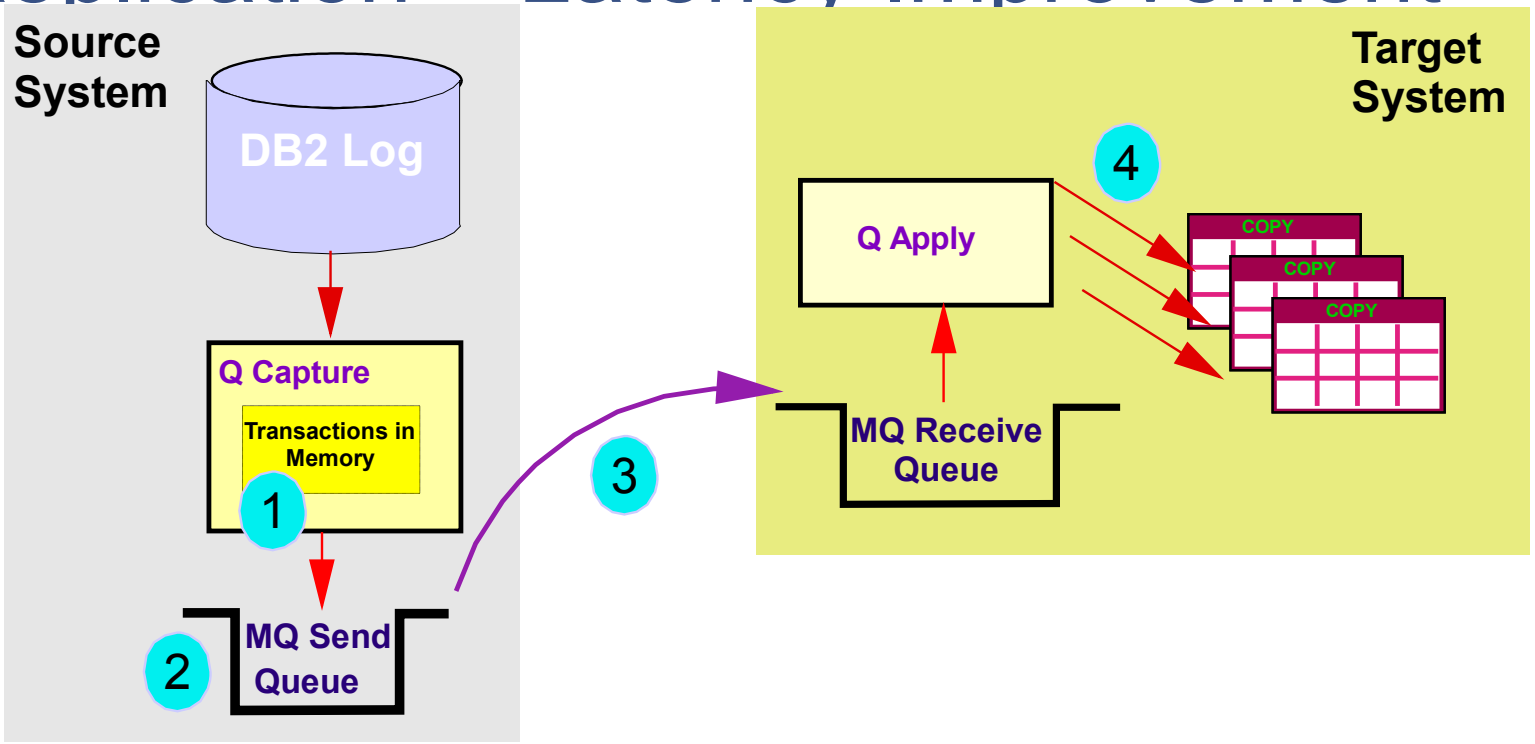
- **Hot Standby**
  - Backup server up and running at all times
- **Multidirectional Replication**
  - Support for geographically-distributed applications
- **Query off-loading**
  - offload production workload

SQL Replication



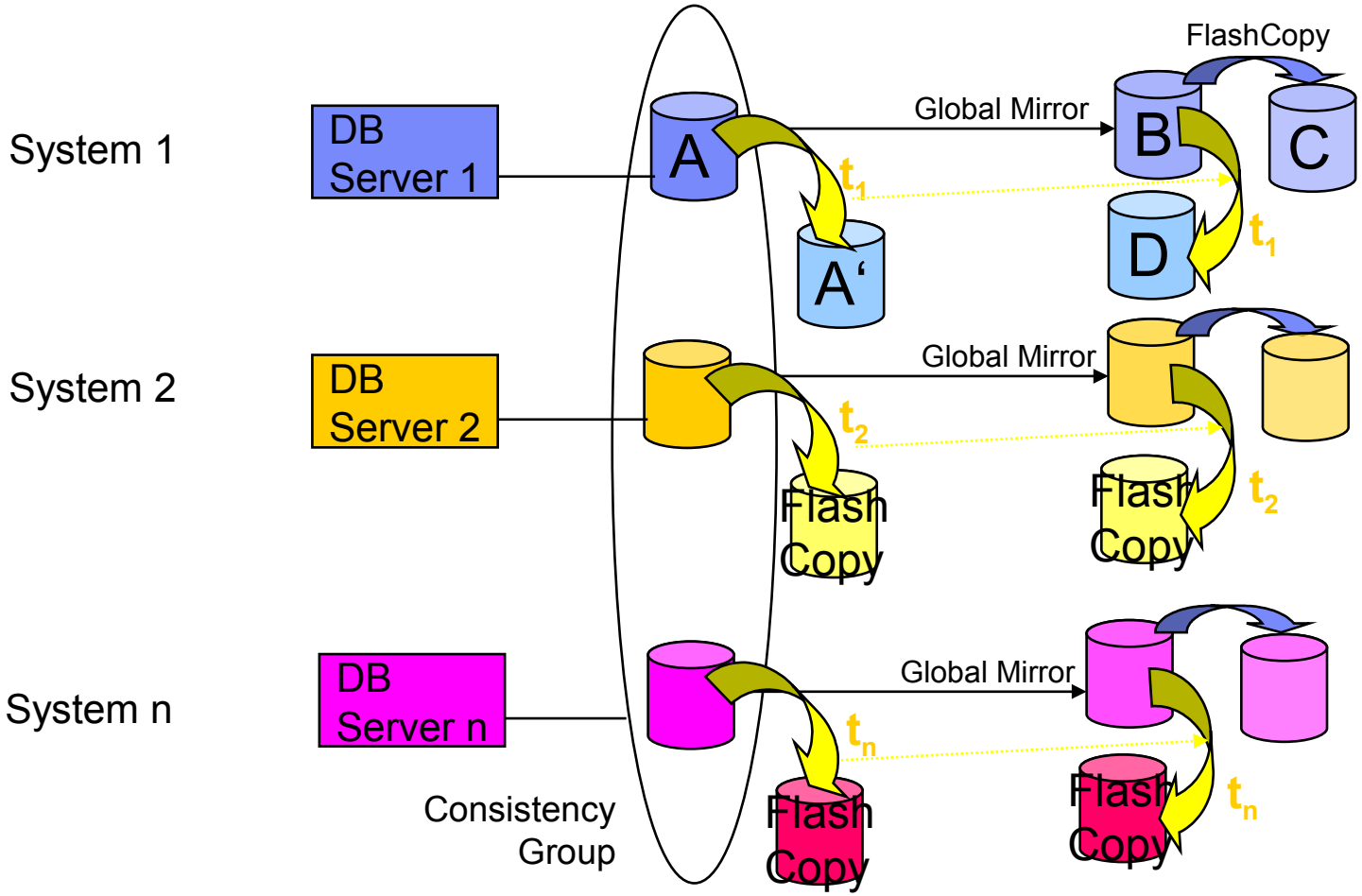
1. Capture breaks transactions apart and inserts into separate CD tables
2. Apply fetches changes via DRDA into memory or disk
3. Stored data is converted into SQL operations on target tables - data can be applied in transactional or table order

Q Replication – Latency Improvement

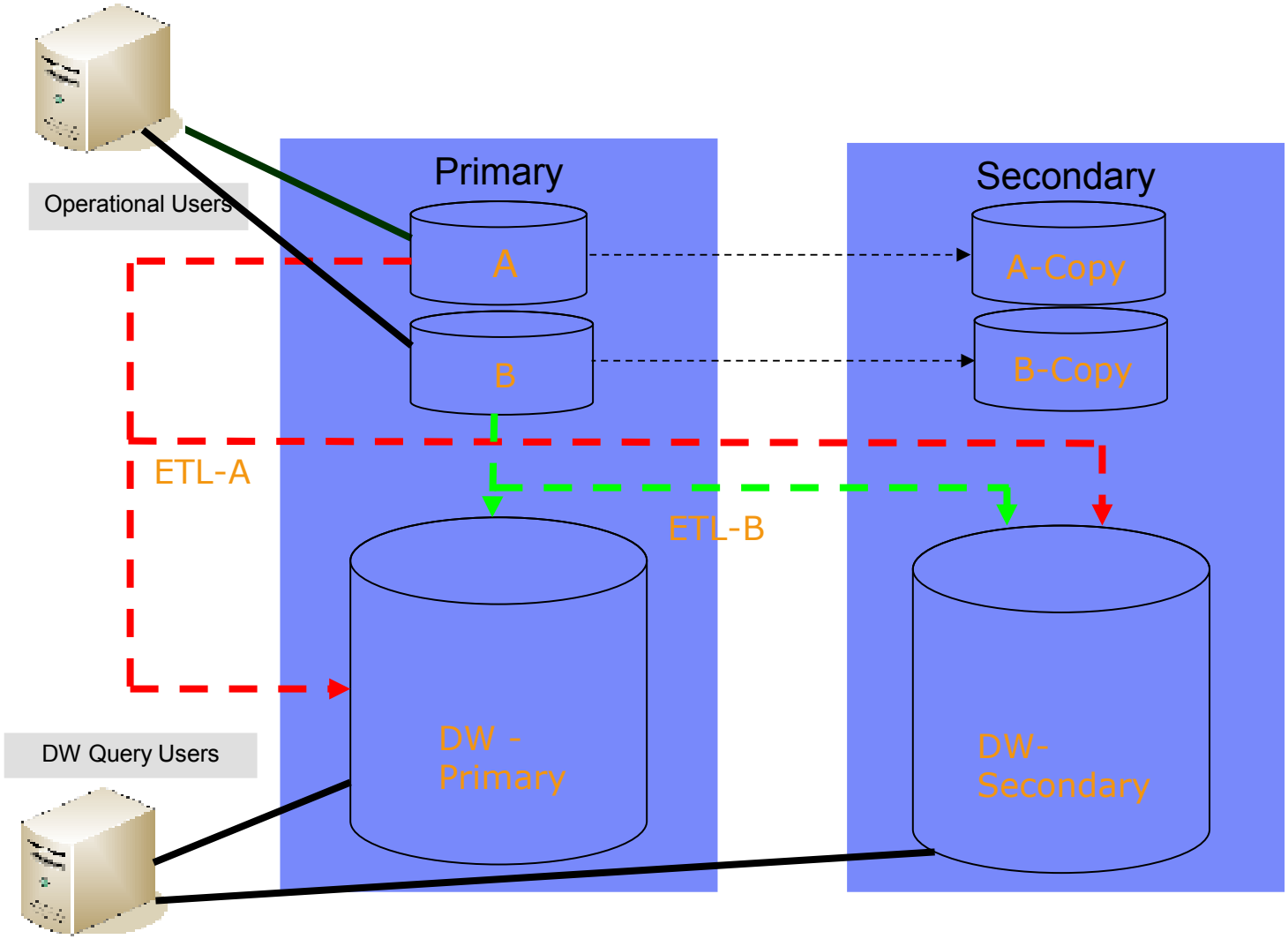


1. Q Capture writes whole transactions - improvement based on size of transaction
2. Q Capture writes data to MQ versus DB2 tables - reducing log contention
3. MQ transports the data - not part of a serialized Apply behavior
4. Q Apply is highly parallelized

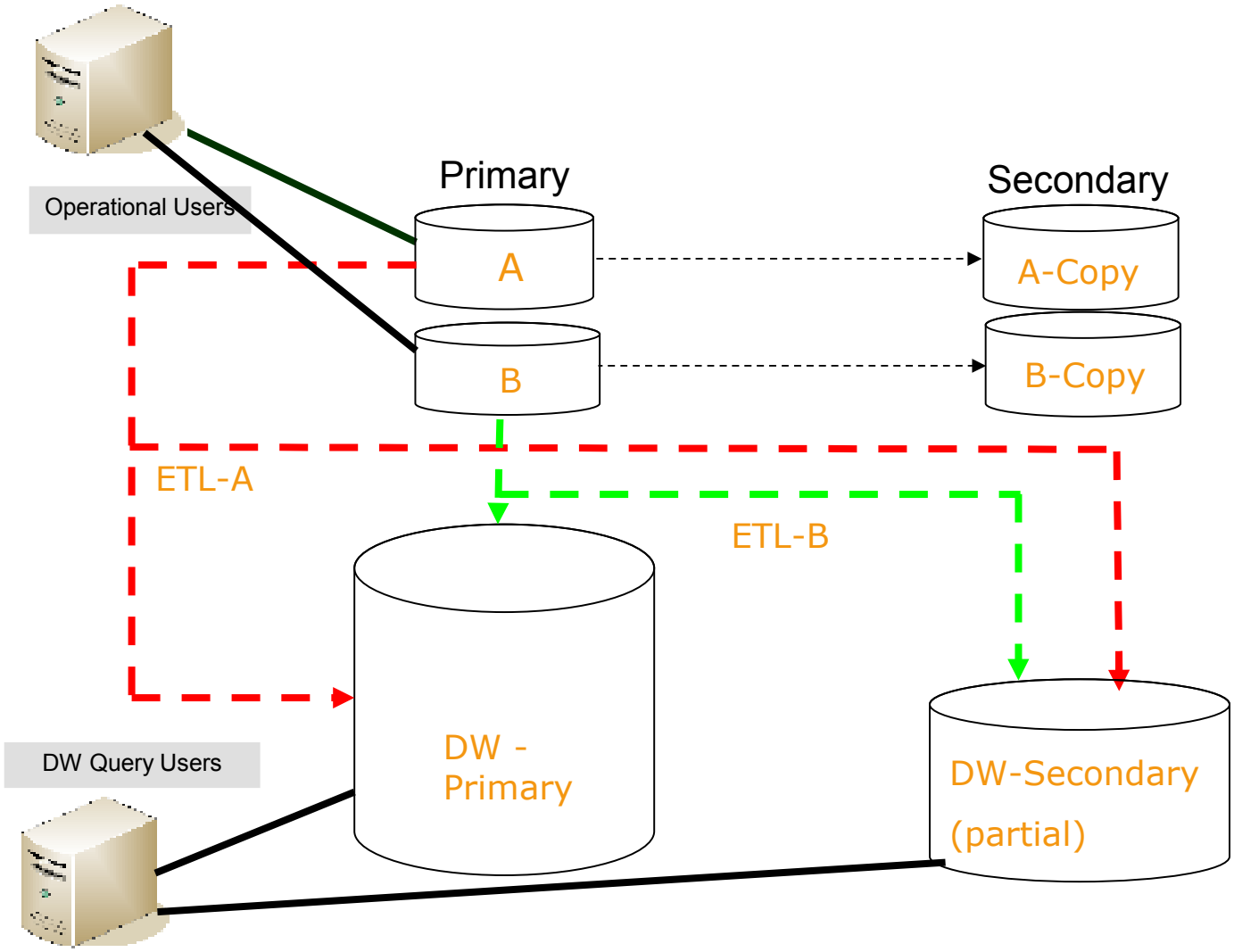
Storage Based Replication



Primary/Secondary Dual Sites: Dual-Site ETL (Full)

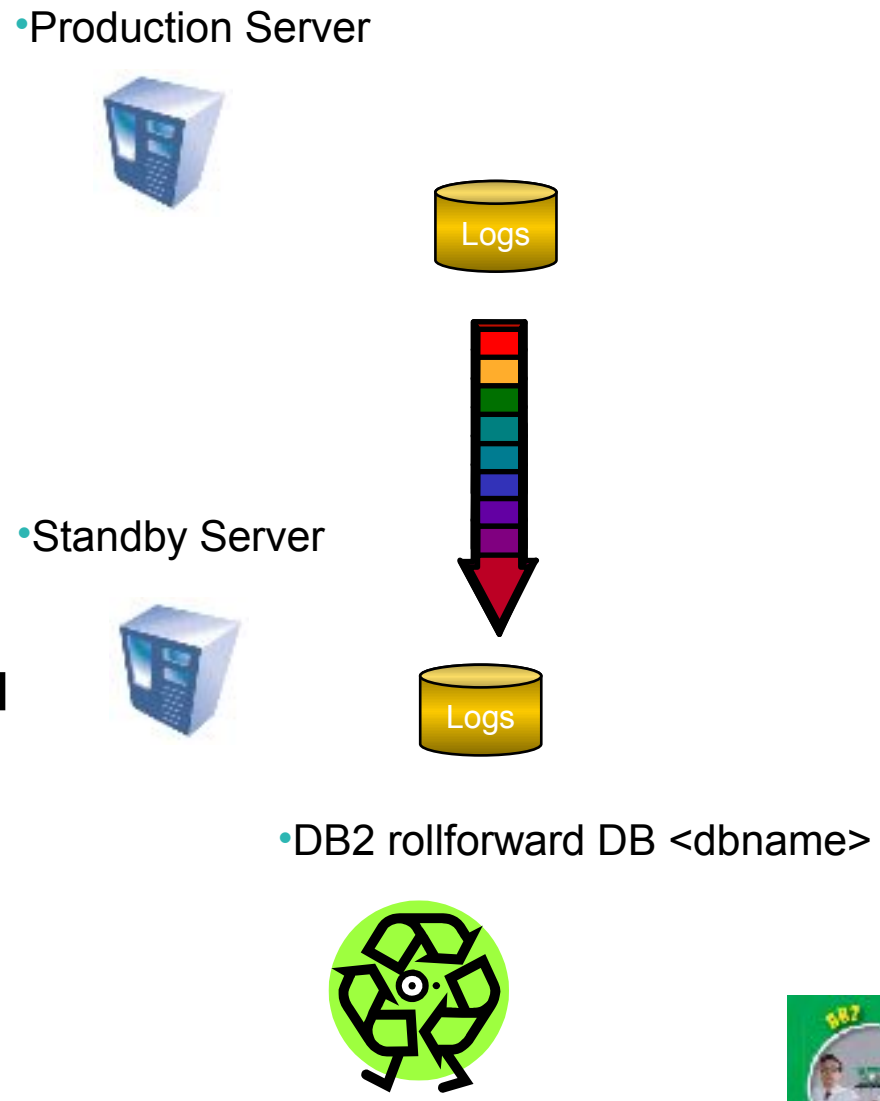


Primary/Secondary Dual Sites: Dual-Site ETL (Partial)



DR Using Log Shipping Overview

- **Send log files to a standby database**
- **Standby database replays log records**
- **Standby not available for any access until failover**
- **Have to re-initialize standby after certain operation**
  - E.g. Load, Index Rebuild, Not-Logged Initially Transaction, …
- **Described in white paper located at**
- **http://www-106.ibm.com/developerworks/db2/library/techarticle/0304mcinnis/0304mcinnis.html**

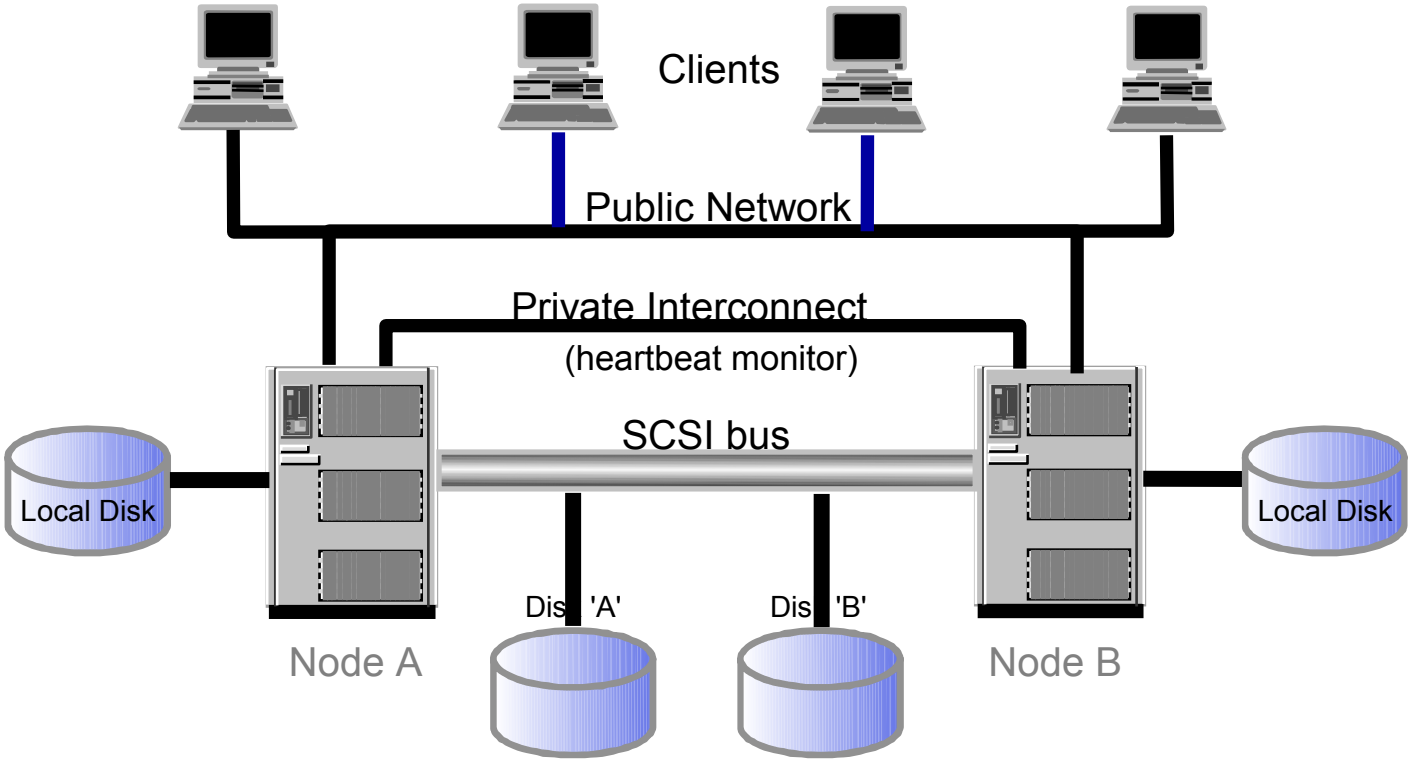


Agenda

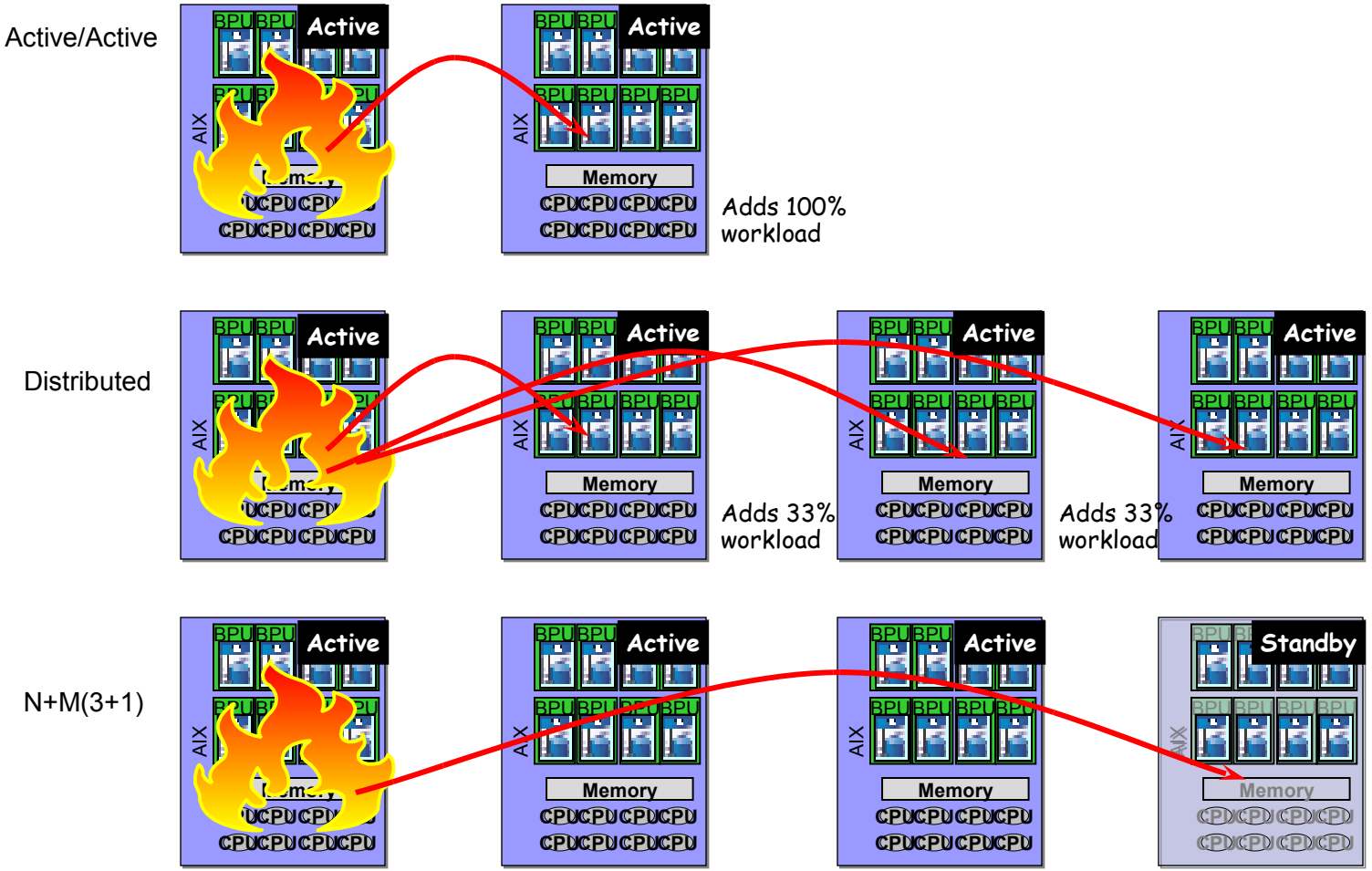
- **High Availablity Options Overview**
  - Purescale, Integrated TSA, HADR
- **Disaster Recovery Options Overview**
  - HADR, Q Repl, Storage Replication, Dual ETL, Log Shipping
- **TSA Integration**
  - Shared disk failover (ESE or DPF), HADR Takeover automation
- **HADR - Overview and RoS**
- **Trends - Active/Active for DR**

Clusters and DB2

- **Length of outage depends on ...**
  - Cluster fail detection and resource takeover time
    - IP address takeover; disk takeover; file system recovery (if needed)
  - Database recovery (redo/undo)
  - Client reconnect
- **Hints/tips to minimize ...**
  - Resource takeover
    - Use DMS DEVICE containers
      - no file system recovery
  - Database recovery
    - Tune *SOFTMAX* and *LOGFILESZ*



DB2 HA Feature: DPF



Clustering Setup Pre-9.5

Overworked admin doing initial setup

Install cluster SW

Install DB2

Create DB2 instance & database

Add nodes / partitions to DB2 instance

Create cluster domain

Add each DB2 node's host to cluster domain

Design and create 'resource models'

Take multiple courses on cluster management

Overworked admin adding a new file system for DB2 (eg. tablespace container or storage path)

Add container or storage path to DB2 (ALTER)

Remember to modify cluster resource model to account for new file system

Refer to cluster management course material to figure out how to do this

BRING OUT THE CONSULTANTS!

Clustering Setup with 9.5

Relaxed admin doing initial setup

- Install Viper 2
- Create DB2 instance & database
- Run DB2 HA config. tool – db2haicu
- Add nodes / partitions to DB2 instance

- Add container or storage path to DB2 (ALTER)

Relaxed admin adding a new file system for DB2 (tablespace container or storage path)

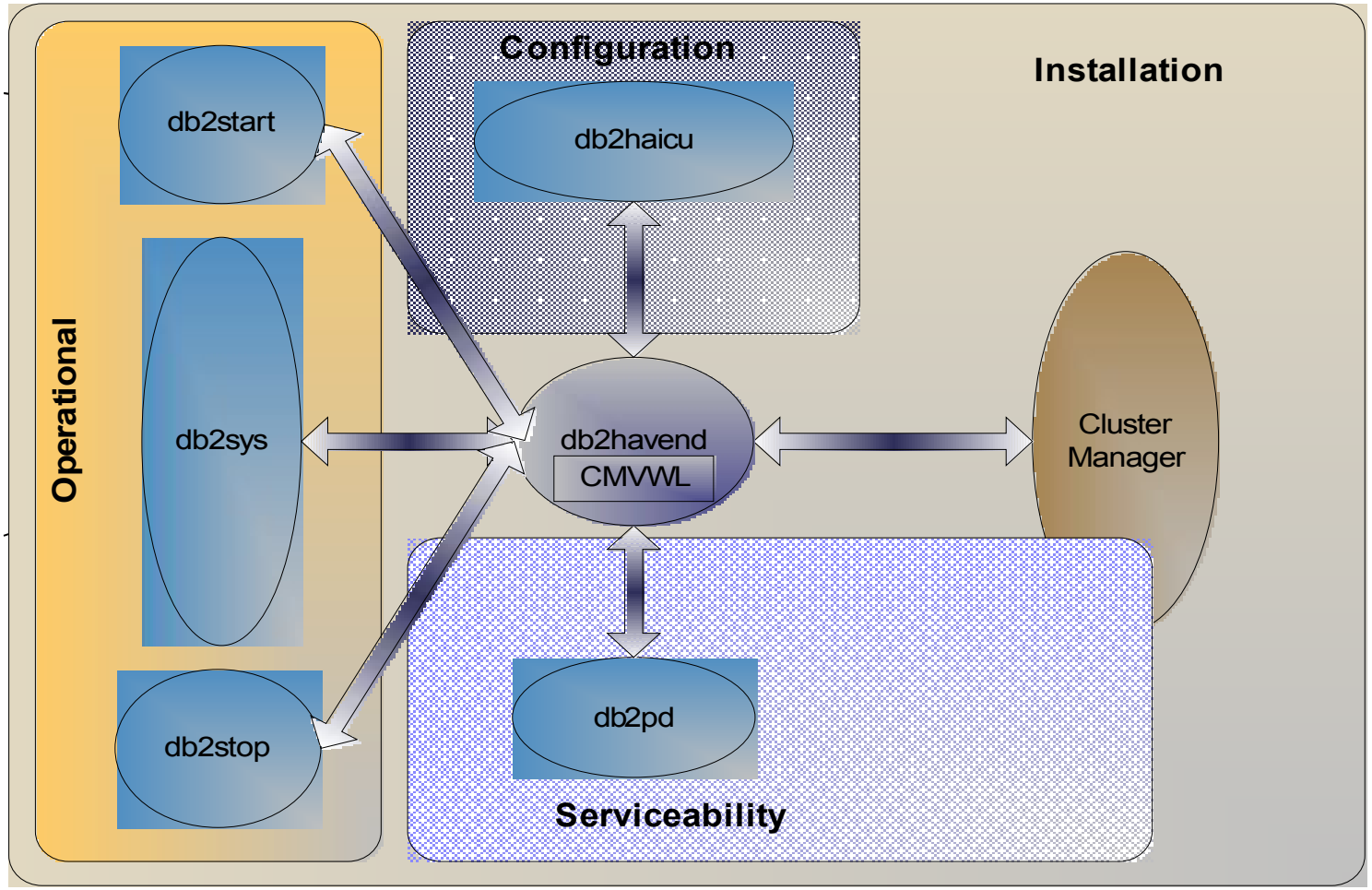
DB2 9.5 Integrated HA and DR

- **Cluster manager services provided with DB2**
  - DB2 …
    - Provides interface to setup cluster manager
      - discovers resources & allows confirmation
      - allows failover policy to be specified
    - DB2 automatically maintains cluster configuration, add node, add tablespace, …
    - DB2 automates failover (via cluster manager)
  - Supports HADR and non-HADR configurations
  - In 9.5, DB2 utilizes Tivoli SA, and supports AIX and Linux
    - Exploits architected new vendor independent layer cluster manager support layer
    - We are working with other cluster manager dev teams to extend support
- **<u>NO SCRIPTING REQUIRED!</u>**
  - One set of internal scripts that are used by all cluster managers

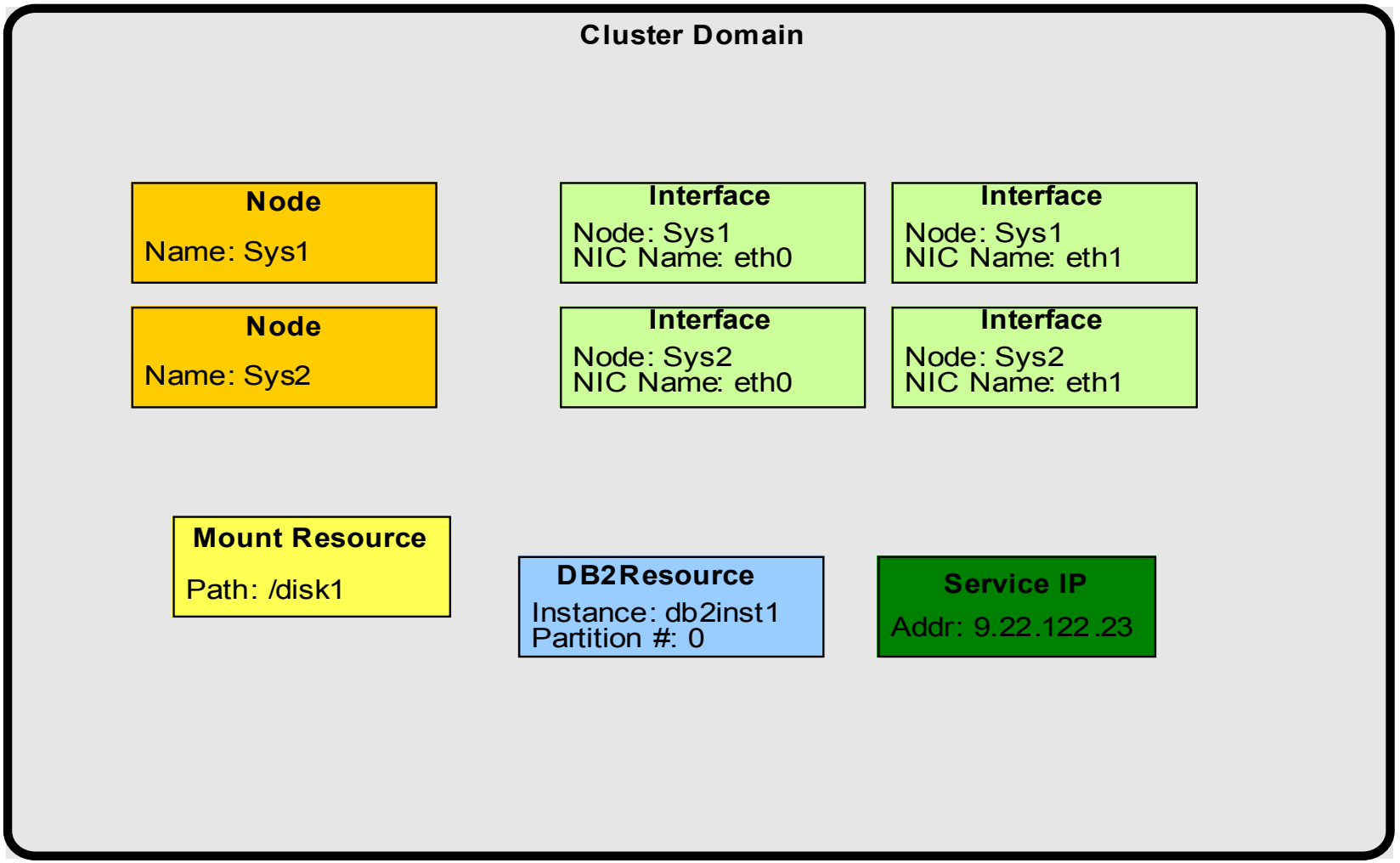
Integrated HA / DR architecture

Db2start
ADD NODE

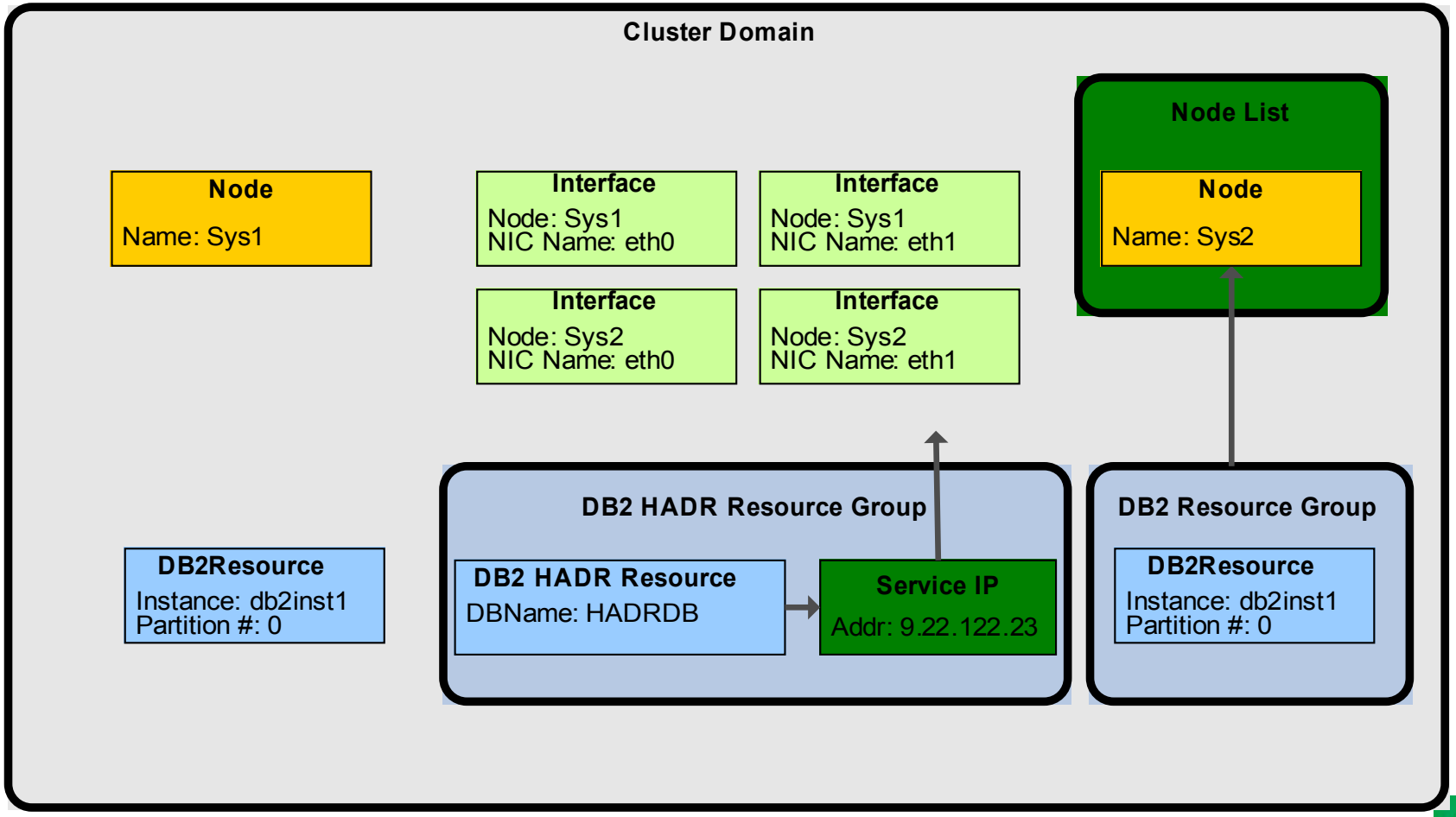
ALTER …
ADD …



DB2 ESE shared storage HA definitions



DB2 HADR configuration for HA

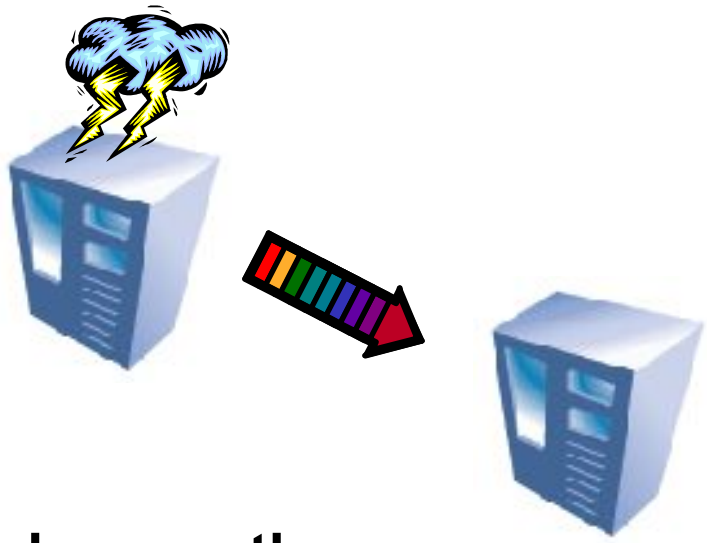


Agenda

- **High Availability Options Overview**
  - Purescale, Integrated TSA, HADR
- **Disaster Recovery Options Overview**
  - HADR, Q Repl, Storage Replication, Dual ETL, Log Shipping
- **TSA Integration**
  - Shared disk failover (ESE or DPF), HADR Takeover automation
- **HADR - Overview and RoS**
- **Trends - Active/Active for DR**

Basic Principles of HADR

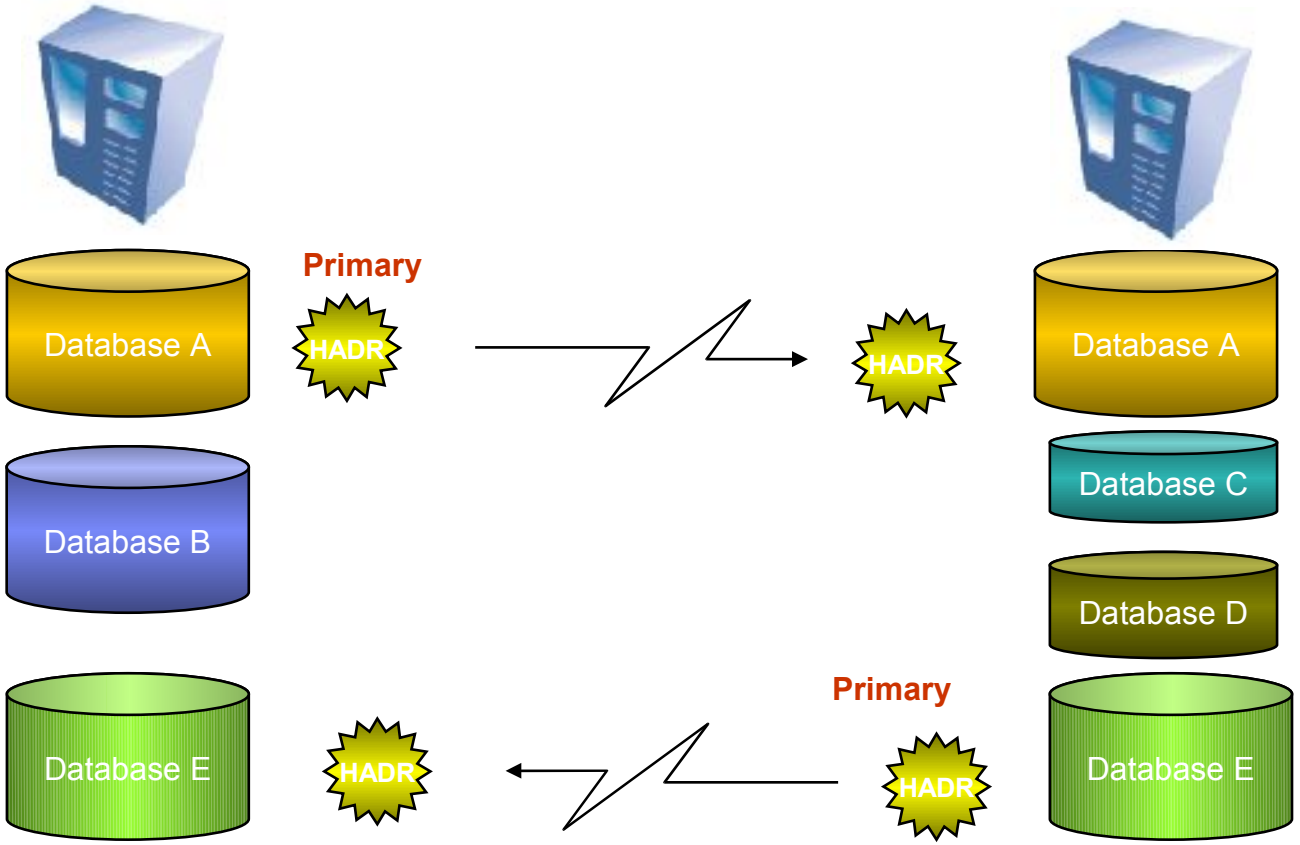
- **Two active machines**
  - Primary
    - Processes transactions
    - Ships log entries to the other machine
  - Standby
    - Cloned from the primary
    - Receives and stores log entries from the primary
    - Re-applies the transactions



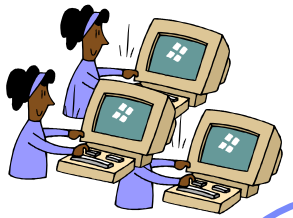
- **If the primary fails, the standby can take over the transactional workload**
  - The standby becomes the new primary
- **If the failed machine becomes available again, it can be resynchronized**
  - The old primary becomes the new standby

Scope of Action

HADR replication takes place at the database level.



HADR Implementation



Primary Connection

Client Reroute

DB2 Engine

PRIMARY SERVER

STANDBY SERVER

DB2 Engine

TCPIP

HADR

Log Pages

HADR

Log Pages

Shredder

Log Records

Replay Master

Log Pages

Log Writer

Log Reader

Log Pages

Log Reader

Log Writer

Log Records

Replay Slaves

Tables

Indexes

Logs

Old

Logs

Old

Logs

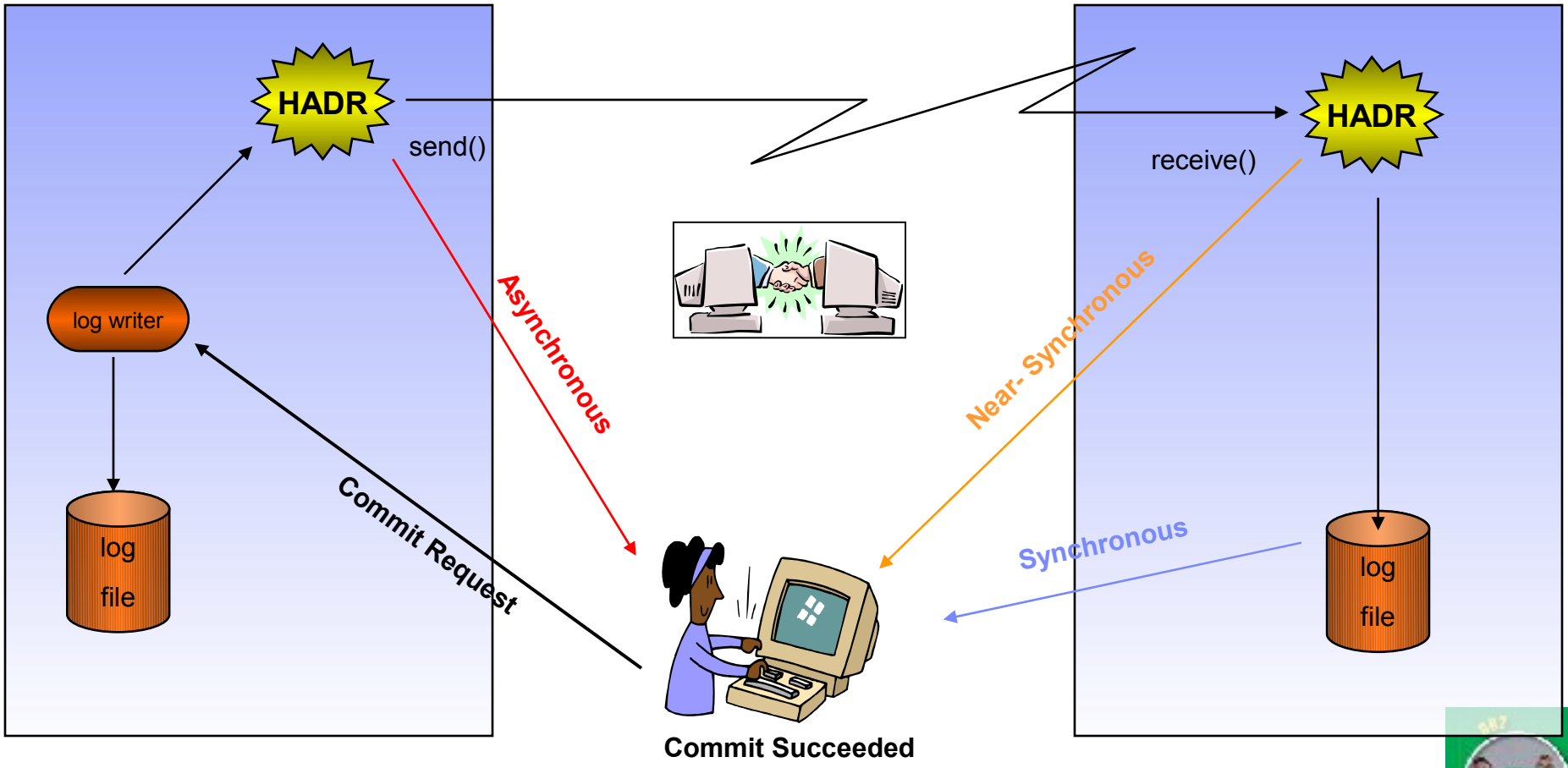
Logs

Tables

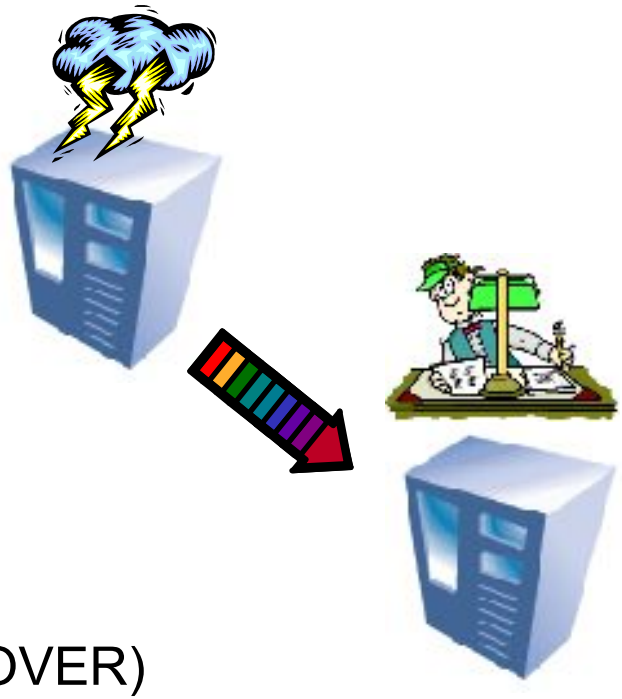
Indexes

Synchronization modes

Synchronous, Near Synchronous, Asynchronous



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 HDR ON DATABASE <dbname>
        <USER <username> [USING <password>]]      [BY FORCE]
```

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 mode
- 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 secondary

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

1.Issue a normal TAKEOVER

- The primary and standby change roles

1.Suspend the new standby

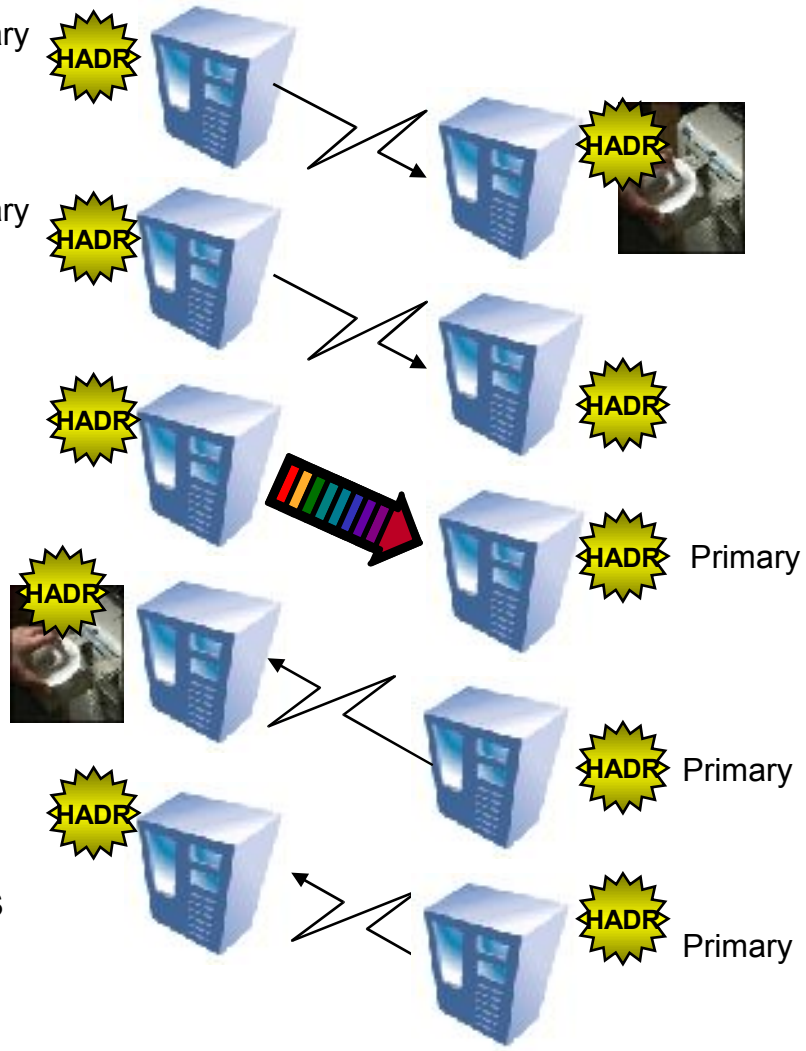
2.Upgrade the new standby

3.Reactivate the new standby

- Let it catch-up with primary

1.Optionally, TAKEOVER again

- The primary and standby play their original roles



Monitoring HADR - snapshot

- **db2 get db snapshot …**

```
HADR Status
  Role                          = Primary
  State                          = Peer
  Synchronization mode          = Nearsync
  Connection status             = Connected, 01/16/2004 16:23
  Heartbeats missed             = 0
  Local host                    = bluestar.ibm.com
  Local service                 = 17003
  Remote host                   = xman.ibm.com
  Remote service                 = 17002
  Remote instance                = dmcinnis
  timeout(seconds)              = 100
  Primary log position(file, page, LSN) = S0000005.LOG, 1747, 0000000003D83A27
  Standby log position(file, page, LSN) = S0000005.LOG, 1747, 0000000003D83A26
  Log gap running average(bytes) = 2345
```

St Heartbeats missed

Log Gap

cted

Congested

Remote Catch Up

Peer

Disconnected

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.

Are LOBs replicated?

- **LOBs**
  - User can define LOBs as logged or not logged.
  - LOBs larger than 1 GB can only be defined as not logged.
  - Logged LOBs are replicated.
  - Not logged LOBs: data is not replicated, but LOB space is allocated on standby. The LOBs on the standby will have the right size, but the content will be binary zero.

HADR Restrictions

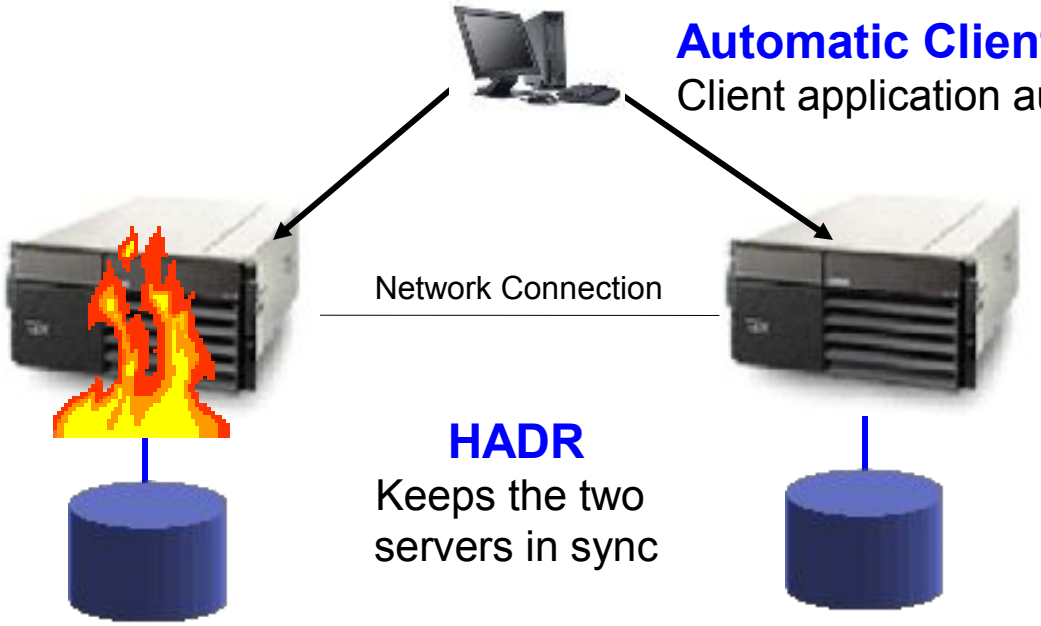
- **Same OS on primary and standby.**
- **Same endian.**
- **Same db2 major version.**
- **Same minor version (fix packs) recommended.**
  - Different minor version is allowed because it is needed for rolling upgrade. But it is not recommended for normal operation.
  - When minor versions are different, the primary can not be newer because a newer primary might generate log records the standby can not understand.

HADR Specific DB Configuration Parameters

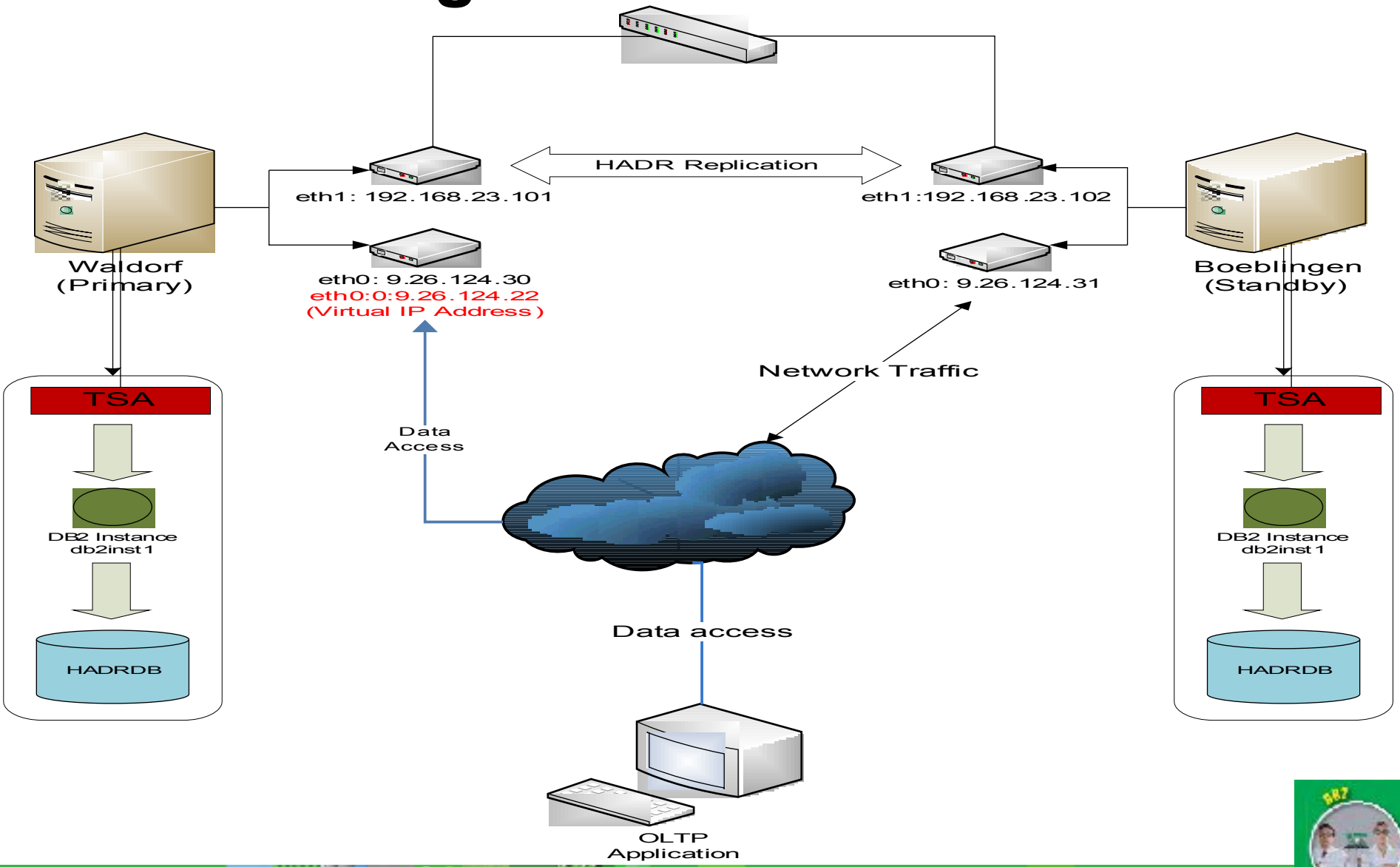
- **BLOCKNONLOGGED**
  - Added in v 9.5 fp4, v 9.7
  - Prevents NLI, non-recoverable LOADs, tables to be defined with non-logged LOB
- **LOGINDEXBUILD**
  - Logs all pages of the index as it is being built
  - Ensures all indexes are available when takeover is issued

DB2 Delivers fast failover at a fraction of the cost

- Redundant copy of the database to protect against site or storage failure
- Support for Rolling Upgrades
- **Failover in under 15 seconds**
  - **Real SAP workload with 600 SAP users – database available in 11 sec.**
- 100% performance after primary failure

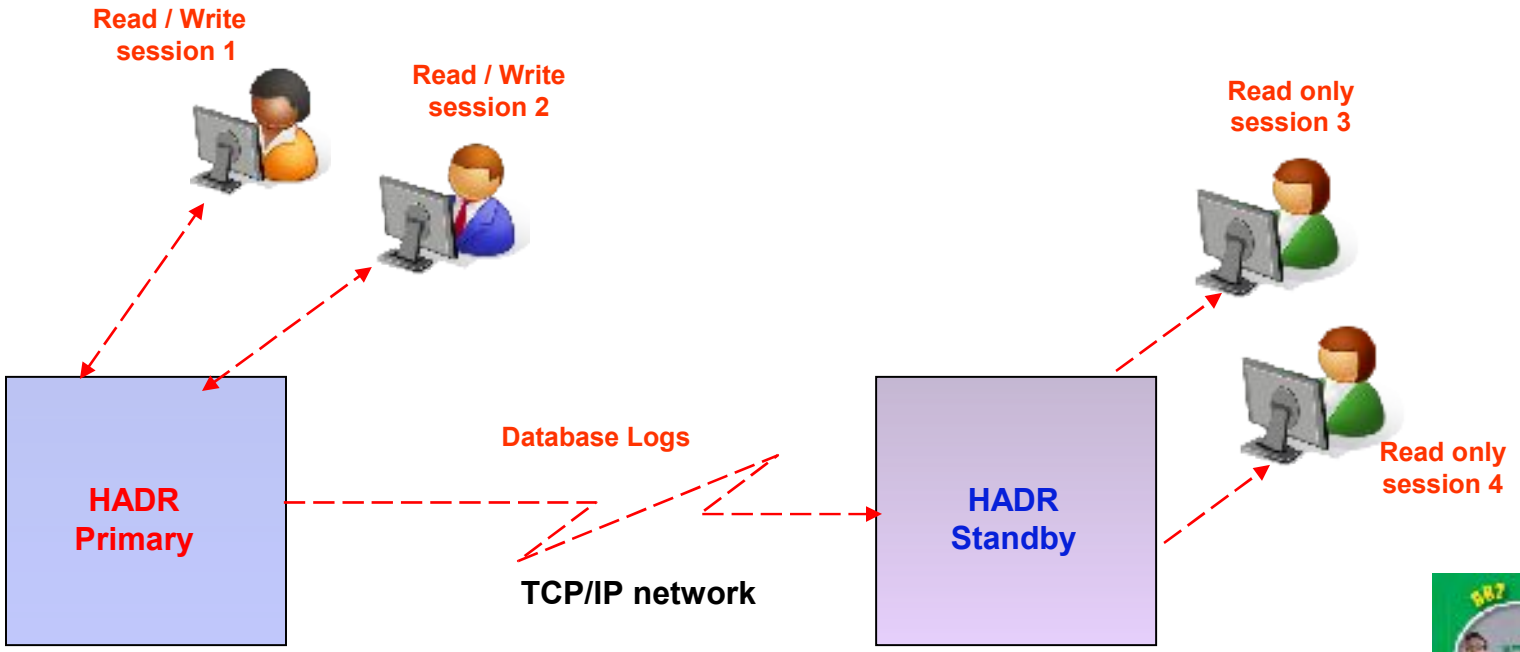


HADR Configuration

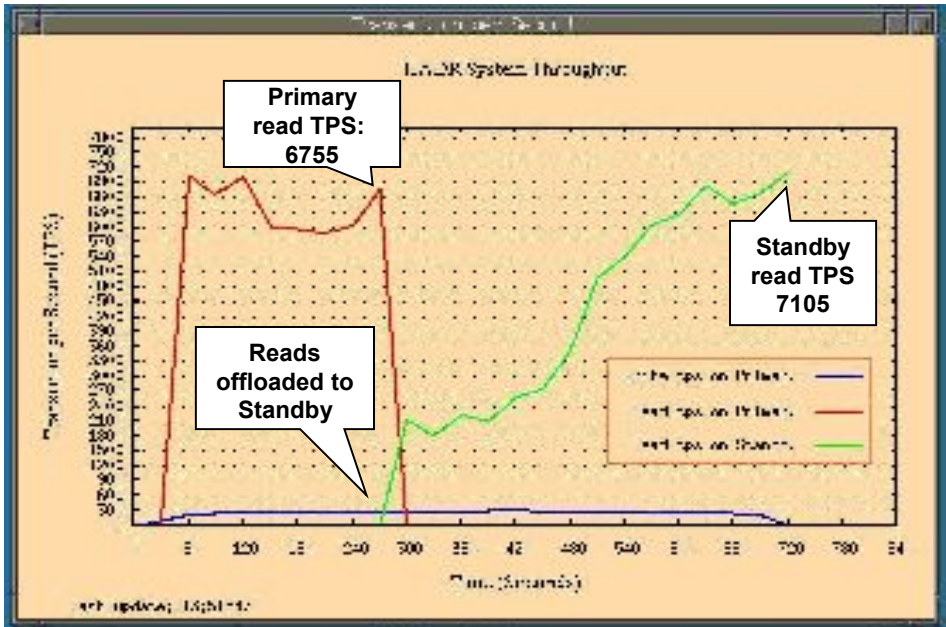


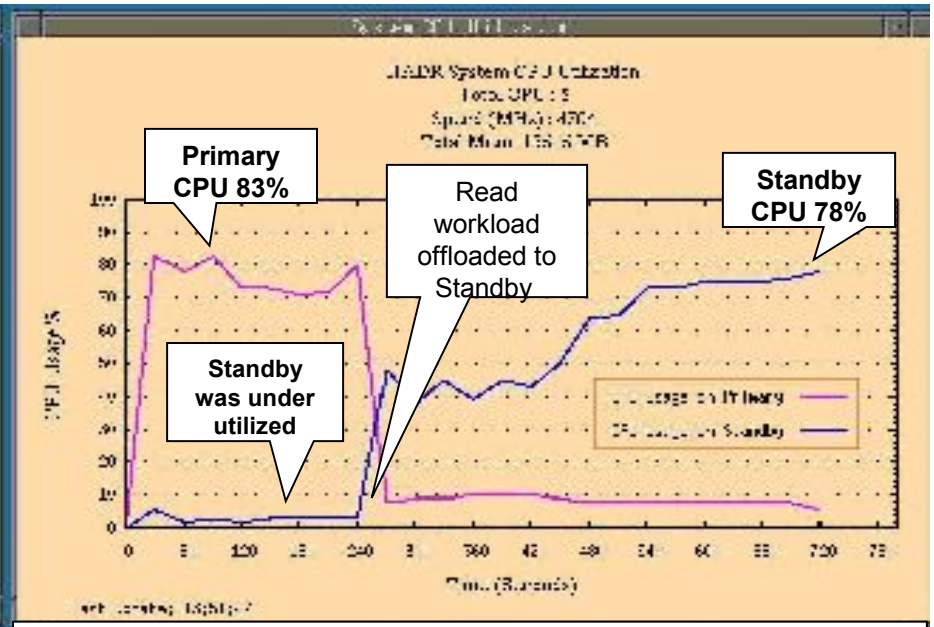
Value Proposition of Reads On Standby

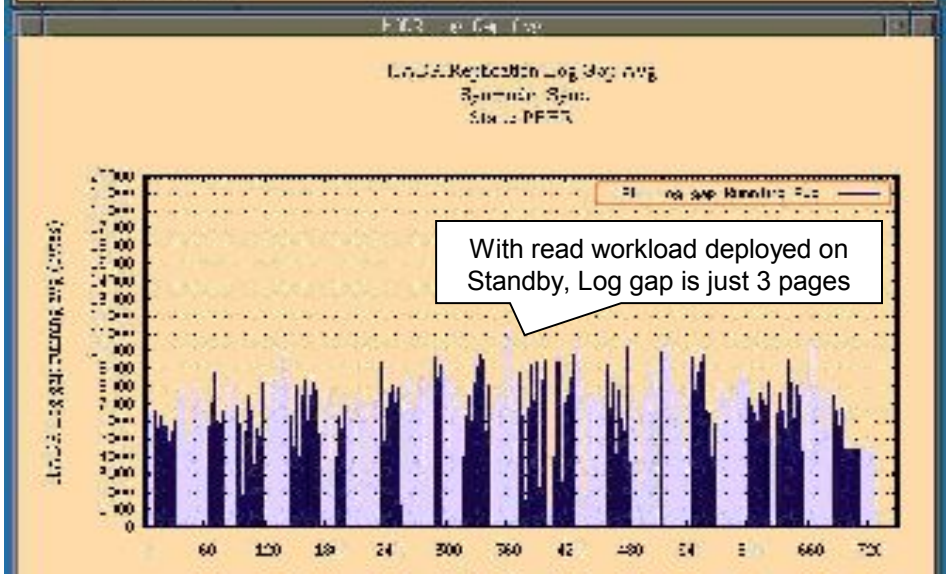
- **Reads on Standby allows read only workloads.**
- **Improve resource utilization on your HA or DR hardware**
- **Offload reporting work from your primary, increase capacity of HADR system**
- **Maximize Return on Investment**



HADR Performance With Reads on Standby







- **Great performance**

  After a little warm up, Standby is able to produce better throughput (read TPS) than Primary

- **No impact to Primary**

  Write transaction rate on Primary is pretty much constant

- **No compromise to availability**

  With heavy reads on Standby, HADR log gap average is just 3 pages

  Takeover time of 1.5 – 2 seconds with heavy reads on Standby, comparable with no reads.

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.

HADR Reads on Standby

Client Governing Rules

- Only Uncommitted Read (UR) aka “dirty read” isolation level allowed
  - Default behavior: Queries with higher isolation will receive an error.
  - Set registry variable DB2_STANDBY_ISO=UR to allow all applications to run at UR isolation with no modifications
- All clients will be terminated on replay of DDL/maintenance operations on the standby.
  - Clients are allowed back only after replay of DDL/maintenance operations are completed.
- Clients are forced off of standby on Takeover
- Client’s write attempts will receive an error.

HADR Reads on Standby

Limitations

- Concurrent replay of DDL/maintenance operations and read.
- LOB, XML, LONG VARCHAR and LONG GRAPHIC reads
  - supported only on primary
- Self Tuning Memory Management
  - supported only on primary
- Creation or declaration of user defined temp tables
  - supported only on primary

Which option to use?

Main Priority	Best Approach
Instant failover and active standby	Q Replication
Simplified setup/mgmt and/or very quick failover and/or no transaction loss guarantee	HADR
Less expensive solution for server failure	Local cluster failover HACMP / TSA / MSCS / etc

Agenda

- **High Availability Options Overview**
  - Purescale, Integrated TSA, HADR
- **Disaster Recovery Options Overview**
  - HADR, Q Repl, Storage Replication, Dual ETL, Log Shipping
- **TSA Integration**
  - Shared disk failover (ESE or DPF), HADR Takeover automation
- **HADR - Overview and RoS**
- **Trends - Active/Active for DR**

Trends

- **Need to provide relief from both planned and outplanned outages**
  - Offline reorgs
  - Backups
  - Loads
  - Schema evolution
  - Version upgrades
- **This will require two active systems with complete redundancy of the data**
  - Data will have to be logically applied to remove version dependence
  - Applications must be able to run on either system

> Questions

55

Thank You!

ibm.com/db2/labchats

Thank you for attending!

56