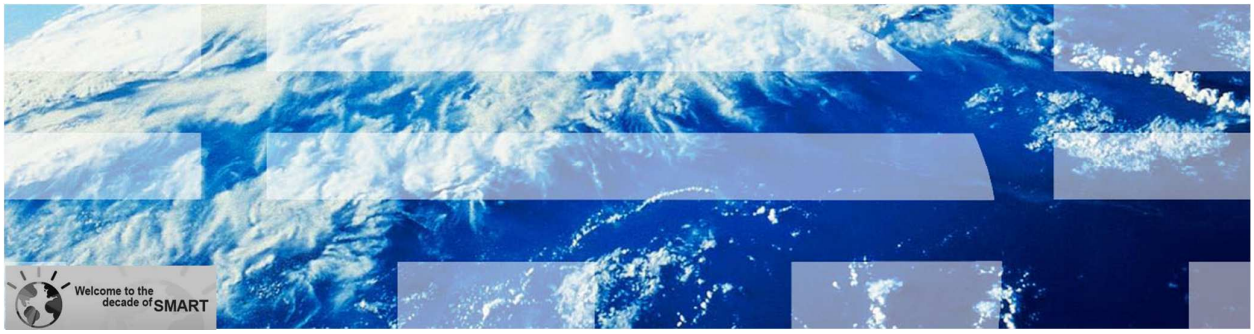
DB2 Night Show DB2 10.1 HADR

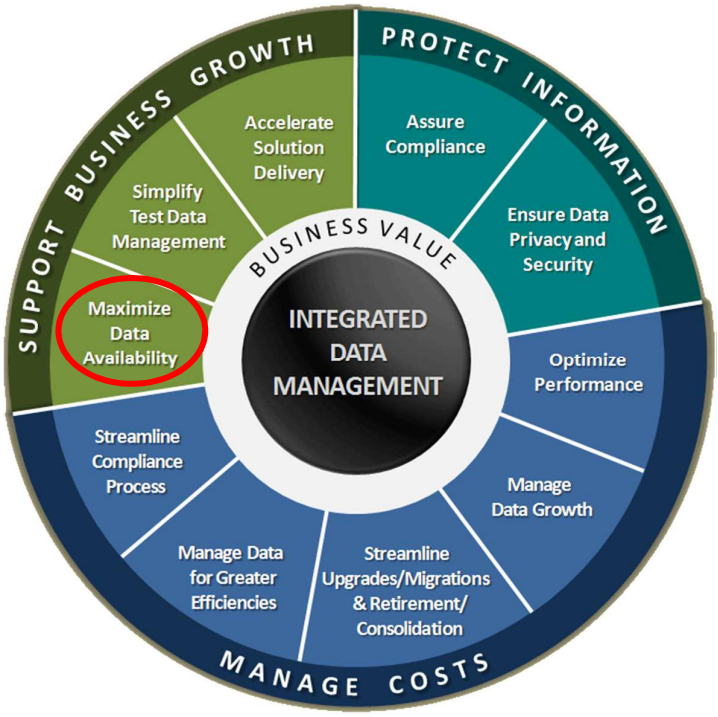


June 08th, 2012
Vikram S Khatri

DB2 10.1 HADR Introduction

Integrated Data Management Solutions
These tools are part of an on-ramp to value with modular business solutions

- ★ **Support Business Growth**
  - ➢ Increase revenue
  - ➢ Retain customers
  - ➢ Enter new markets
- ★ **Protect Information**
  - ➢ Adhere to regulations
  - ➢ Mitigate Risk
- ★ **Manage Costs**
  - ➢ Reduce cost and overhead
  - ➢ Streamline infrastructure
  - ➢ Centralize shared services
  - ➢ Foster cross role collaboration



DB2 10.1 HADR
Packaging options

DB2 Editions	HADR Feature
Express-C Edition	**X**
Express Edition	✓
Workgroup Server Edition	✓
Database Enterprise Developer Edition	✓
Enterprise Server Edition	✓
Advanced Enterprise Server Edition	✓

High Availability
Cold, Warm and Hot

Cold	Warm	Hot
Database software is installed on another server	Database software is installed on another server	Database software is installed on another server
Database instance on the another host is not started during normal operations	Database instance on another host is started and it is receiving updates from the Primary but no end user activity	Database instance on another host is started and it is receiving updates from the Primary. End user access is available to the database.
Example: Active / Passive with a shared database using TSA	Example: HADR, Q-Replication without Read on Standby	Example: HADR with Read on Standby, Multiple Targets with Read on Standby, DB2 pureScale, Peer to Peer replication using SQL or Q

DB2 Licensing
Various Options

License Type	Description
Server License	It is available for DB2 Express edition. Buy license for each physical or virtual server
Fixed Term License (FTL)	It is available for DB2 Express and FTL license gives you access for the software for the period of one year
Socket License	It is available only for Workgroup edition. Example: A 4 way dual core Power 7 server would require 4 Workgroup socket license
Processor Value Unit (PVU) License	Using PVU model
Authorized User Single Install (AUSI) License	For any DB2 product server, buy total number of authorized users license of each server. For each edition, it requires a minimum number of authorized user license.

As a guideline, a hot / hot configuration should be licensed same way you would license each server as if they weren't clustered at all.

HADR
Licensing for HA Standby Server

DB2 Editions	Standby Server Licensing	
Express-C Edition	X	
Express Edition	Cold	No charge for the Standby server
	Warm	100 PVU / 5 Users for Users / 1 LUV (*)
	Hot	Must be licensed same as Primary
Workgroup Server Edition	Cold	No charge for the Standby server
	Warm	100 PVU / 1 Socket / 5 users (*)
	Hot	Must be licensed same as Primary
Database Enterprise Developer Edition	Cold	No charge for the Standby server
	Warm	DEDE License for another user
	Hot	DEDE License for another user
Enterprise Server Edition / Advanced Enterprise Server Edition	Cold	No charge for the Standby server
	Warm	100 PVU / 25 Users (*)
	Hot	Must be licensed same as Primary

* PUV Licensing / Per User Licensing / LUV pricing

IBM High Availability Solutions
Different Models

IBM High Availability Solutions
Use as per Business Requirements

Bronze
- **Standalone or Active / Passive with Shared Database**
- Use Tivoli System Automation (TSA) on standalone DB2 or build Active / Passive with the shared database

Silver
- **Active / Active with separate database**
- Use HADR feature to sync databases and TSA for failover

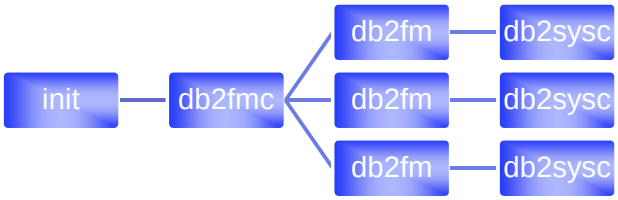
Gold
- **Active / Active with Read on Standby and Multiple Targets**
- Use HADR feature to sync multiple standbys

Platinum
- **IBM DB2 10.1 pureScale (*)**
- Multiple machines using shared database using GPFS and TSA

High Availability for Standalone DB2

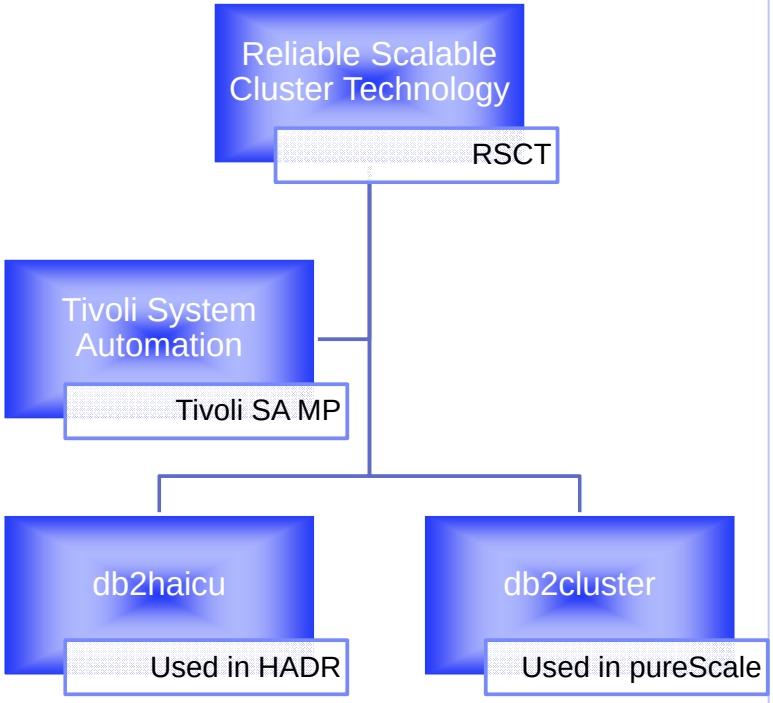
High Availability for Standalone DB2
Fault Monitor

- Available only on Linux / Unix Platforms
- DB2 provides Fault Monitor
  - It keeps DB2 up and running by monitoring DB2 instances
  - It restarts any instance that exists prematurely
- The Fault Monitor Coordinator (FMC) is the process of the Fault Monitor Facility that is started at the boot time
- The `init` daemon starts the FMC and will restart it if it terminates abnormally
- The FMC starts one fault monitor for each DB2 instance
- If a fault monitor fails, FMC starts it automatically
- If a DB2 process fails, the fault monitor will start it



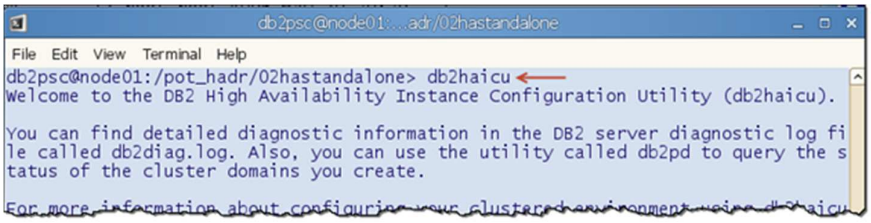
High Availability for Standalone DB2
Tivoli System Automation (TSA)

- Limited License for TSA is included with DB2
- RSCT (like an operating system) provides low level APIs and programs to configure high availability
- TSA is an application of RSCT
- When using TSA, turn off DB2 fault monitor
  `# db2fmcu -d`
- `db2haicu` and `db2cluster` are actually wrappers for RSCT / TSA



Make DB2 Highly Available
db2haicu

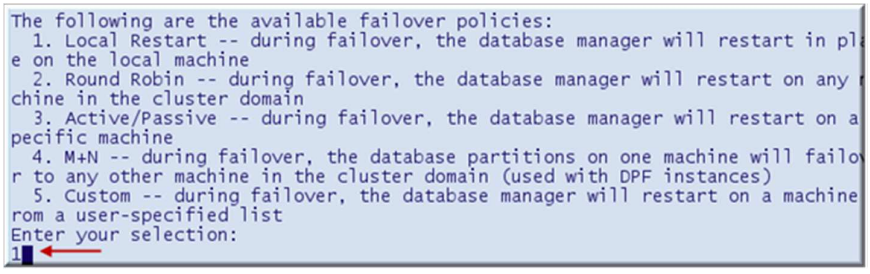
- db2haicu is a text based utility that can be used to configure and administer a peer RSCT domain
- The RSCT command preprpnode needs to be run on each node to generate shared keys.
  $ sudo preprpnode node01
- When run for the first time, it creates a peer domain.



- On later invocation, db2haicu is used to resume the task of configuration of the peer RSCT domain or to maintain it.

Make DB2 Highly Available
db2haicu (continued…)

- On a single instance of DB2, use only one node in the peer domain
- Use Local Restart policy



- Exclude /home mount point or DB2 mount points from TSA purview for standalone DB2.
- No Virtual IP Address since we are using standalone DB2

Make DB2 Highly Available
Explore HA Infrastructure

★ **$ lsrpdomain**

```
db2psc@node01:...adr/02hastandalone
File  Edit  View  Terminal  Help
db2psc@node01:/pot_hadr/02hastandalone> lsrpdomain
Name  OpState RSCTActiveVersion MixedVersions TSPort GSPort
db2ha Online  3.1.2.2           No            12347  12348
db2psc@node01:/pot_hadr/02hastandalone>
```

★ **$ lsrpnode**

```
db2psc@node01:~> lsrpnode
Name   OpState RSCTVersion
node01 Online  3.1.2.2
db2psc@node01:~>
```

★ **$ lssam**

```
db2psc@node01:/pot_hadr/02hastandalone> lssam
Online IBM.ResourceGroup:db2_db2psc_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2psc_0-rs
                 '- Online IBM.Application:db2_db2psc_0-rs:node01
Online IBM.Equivalency:db2_db2psc_0-rg_group-equ
        '- Online IBM.PeerNode:node01:node01
Online IBM.Equivalency:db2_private_network_0
        '- Online IBM.NetworkInterface:eth0:node01
db2psc@node01:/pot_hadr/02hastandalone>
```


Make Applications Highly Available
Automatic Client Reroute

★ TSA restarts failed DB2 process or recovers from critical failures but applications do not recover from these errors.

```
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
        at java.lang.Thread.run(Unknown Source)
com.ibm.db2.jcc.am.SqlException: [jcc][10120][10943][3.63.123] Invalid operation
: statement is closed. ERRORCODE=-4470, SQLSTATE=null
        at com.ibm.db2.jcc.am.fd.a(fd.java:663)
        at com.ibm.db2.jcc.am.fd.a(fd.java:60)
        at com.ibm.db2.jcc.am.fd.a(fd.java:103)
        at com.ibm.db2.jcc.am.ho.wb(ho.java:4177)
        at com.ibm.db2.jcc.am.io.b(io.java:3901)
```

★ Configure Automatic Client Reroute (ACR)

```
### Properties for the Automatic Client Reroute
#enableSysplexWLB=true
enableSeamlessFailover=true
enableConnectionConcentrator=true
clientRerouteAlternateServerName=node01
clientRerouteAlternatePortNumber=50000
enableClientAffinitiesList=1
maxRetriesForClientReroute=10
retryIntervalForClientReroute=30
```

ACR Configuration

High Availability - Maintenance
Apply Fix Pack or System Maintenance

- TSA provides High Availability infrastructure.
- During maintenance periods, turn off HA infrastructure.
- **`$ db2haicu -disable`**
- System Resources are locked.

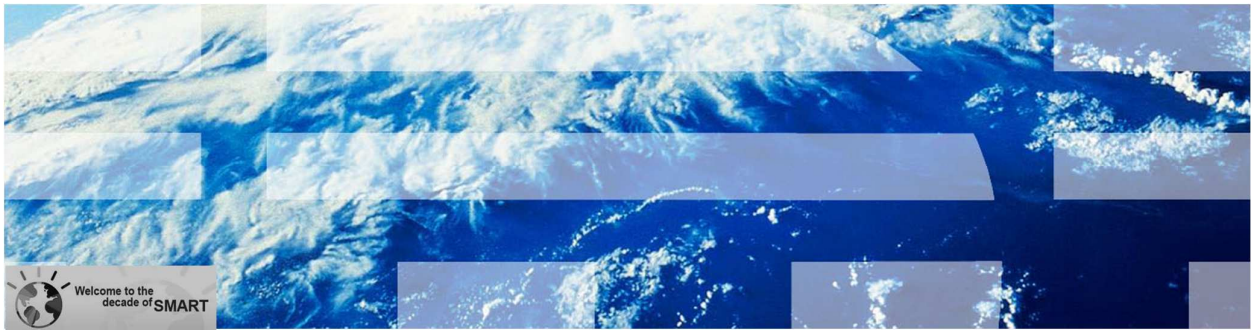
```
db2psc@node01:~/pot_hadr/02hastandalone> lssam
Online IBM.ResourceGroup:db2_db2psc_0-rg Request=Lock Nominal=Online
        '- Online IBM.Application:db2_db2psc_0-rs Control=SuspendedPropagated
                '- Online IBM.Application:db2_db2psc_0-rs:node01
Online IBM.Equivalency:db2_db2psc_0-rg_group-equ
        '- Online IBM.PeerNode:node01:node01
Online IBM.Equivalency:db2_private_network_0
        '- Online IBM.NetworkInterface:eth0:node01
db2psc@node01:~/pot_hadr/02hastandalone>
```

- After maintenance, run `db2haicu` again to reintegrate TSA with DB2
- **`$ db2haicu`**

High Availability - Maintenance
Remove HA

- **`$ db2haicu -delete`**
- Peer domain is removed and thus TSA is removed from managing DB2

High Availability using Active / Passive



High Availability – Active / Passive
Prerequisites and License Install

- TSAMP must be installed – It is part of the DB2 install
- `# ./server/db2/linuxamd64/tsamp/prereqSAM`

```
node01:~/download/server/db2/linuxamd64/tsamp # ./prereqSAM
prereqSAM: All prerequisites for the ITSAMP installation are met on operating system
SUSE Linux Enterprise Server 11 (x86_64)
VERSION = 11
PATCHLEVEL = 1
```

- Install TSAMP License (sam32.lic)

```
# samlicm -i sam32.lic
# samlicm -s
```

```
node01:/pot_hadr/01setup/lic # samlicm -i sam32.lic
node01:/pot_hadr/01setup/lic # samlicm -s
Product: IBM Tivoli System Automation for Multiplatforms 3.2
Creation date: Wed Aug 19 00:00:01 2009
Expiration date: Thu Dec 31 00:00:01 2037
```

- DB2 Fix Pack includes a Try and Buy TSA license and will cause a misleading error on a DB2 system with a valid license
- Remove TSA license before applying Fix Pack and reapply TSA license

High Availability – Active / Passive
Shared Disk for the Database

- DB2 Instance home and storage paths must be shared between two hosts
- Entries in `/etc/fstab` must have `noauto` option.

```
node01:/root/bin        /root/bin         nfs      defaults              0 0
node01:/root/download   /root/download    nfs      defaults              0 0
node01:/pot_hadr        /pot_hadr         nfs      defaults              0 0
/dev/sdd                /db2home          ext3     acl,user_xattr,noauto 0 0
/dev/sde                /db2data1         ext3     acl,user_xattr,noauto 0 0
/dev/sdf                /db2data2         ext3     acl,user_xattr,noauto 0 0
/dev/sdg                /db2log           ext3     acl,user_xattr,noauto 0 0
/dev/sdh                /db2arch          ext3     acl,user_xattr,noauto 0 0
```


High Availability – Active / Passive
DB2 Instance Creation

- Mount shared instance home on first host
  `# mount /db2home`
- Create DB2 instance
  `# db2icrt -p db2c_db2ap -u db2ap db2ap`
- Un-mount shared instance home
  `# unmount /db2home`
- Mount shared instance home on second host
  `# mount /db2home`
- Delete ~/sqllib
- Create DB2 instance
  `# db2icrt -p db2c_db2ap -u db2ap db2ap`
- Mount database Storage Paths and Create DB2 database

High Availability – Active / Passive
Run `db2haicu` to configure High Availability

- Run `preprpnode` on both hosts
  `# preprpnode node02 node03`
- **`$ db2haicu`**
  - creates a new RSCT peer domain
  - creates a network quorum device
  - adds network interfaces to the RSCT resource groups
  - changes DBM parameter to add TSA as a cluster manager
  - uses Active/Passive failover policy
  - configures a virtual IP address between two hosts
  - configures database highly available

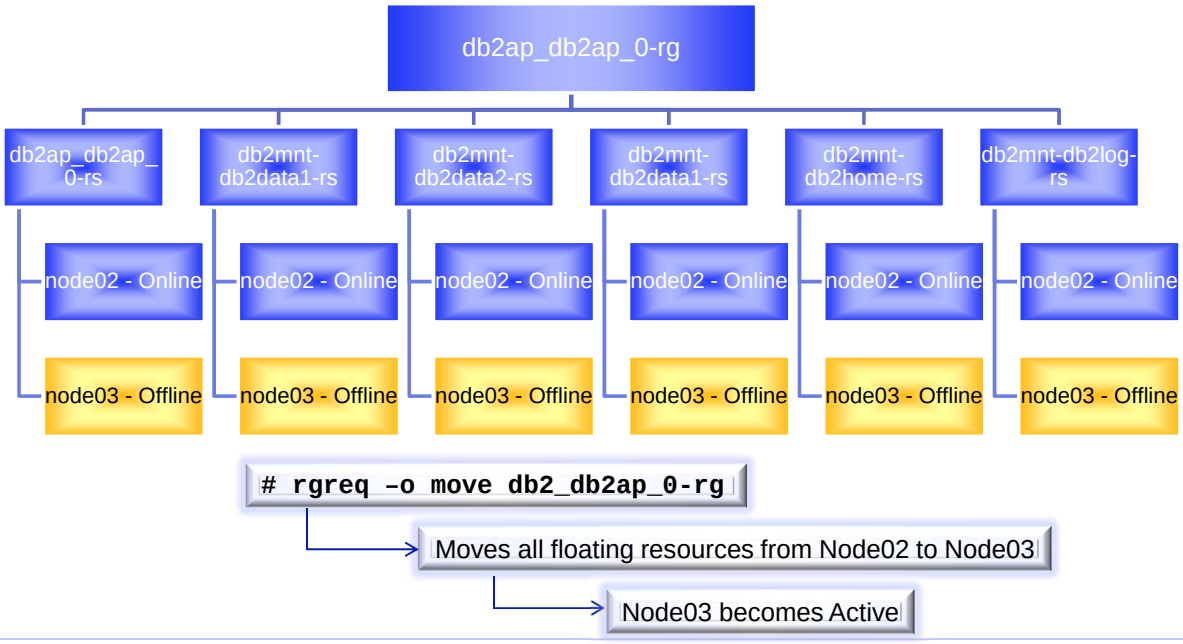
High Availability – Active / Passive
RSCT Resources Hierarchy

- **`$ lssam`**

```
db2ap@node03:~> lssam
Online IBM.ResourceGroup:db2_db2ap_0-rg Nominal=Online
        |- Online IBM.Application:db2_db2ap_0-rs
                |- Offline IBM.Application:db2_db2ap_0-rs:node02
                '- Online IBM.Application:db2_db2ap_0-rs:node03
        |- Online IBM.Application:db2mnt-db2data1-rs
                |- Offline IBM.Application:db2mnt-db2data1-rs:node02
                '- Online IBM.Application:db2mnt-db2data1-rs:node03
        |- Online IBM.Application:db2mnt-db2data2-rs
                |- Offline IBM.Application:db2mnt-db2data2-rs:node02
                '- Online IBM.Application:db2mnt-db2data2-rs:node03
        |- Online IBM.Application:db2mnt-db2home-rs
                |- Offline IBM.Application:db2mnt-db2home-rs:node02
                '- Online IBM.Application:db2mnt-db2home-rs:node03
        |- Online IBM.Application:db2mnt-db2log-rs
                |- Offline IBM.Application:db2mnt-db2log-rs:node02
                '- Online IBM.Application:db2mnt-db2log-rs:node03
        '- Online IBM.ServiceIP:db2ip_192_168_142_100-rs
                |- Offline IBM.ServiceIP:db2ip_192_168_142_100-rs:node02
                '- Online IBM.ServiceIP:db2ip_192_168_142_100-rs:node03
Online IBM.Equivalency:db2_db2ap_0-rg_group-equ
        |- Online IBM.PeerNode:node03:node03
        '- Online IBM.PeerNode:node02:node02
Online IBM.Equivalency:db2_private_network_0
        |- Online IBM.NetworkInterface:eth0:node03
        '- Online IBM.NetworkInterface:eth0:node02
```


High Availability – Active / Passive
Floating Resources

★ Floating Resources



High Availability – Active / Passive
Application Resources

★ Scripts to start, stop and monitor DB2 and mount points are provided by DB2 – Hence no DBA scripting required.

★ `$ lsrsrc –Ab IBM.Application Name StartCommand StopCommand MonitorCommand`

```
resource 8:
        Name           = "db2mnt-db2home-rs"
        StartCommand   = "/usr/sbin/rsct/sapolicies/db2/mountV10_start.ksh /db2home"
        StopCommand    = "/usr/sbin/rsct/sapolicies/db2/mountV10_stop.ksh /db2home"
        MonitorCommand = "/usr/sbin/rsct/sapolicies/db2/mountV10_monitor.ksh /db2home"
resource 9:
        Name           = "db2_db2ap_0-rs"
        StartCommand   = "/usr/sbin/rsct/sapolicies/db2/db2V10_start.ksh db2ap 0"
        StopCommand    = "/usr/sbin/rsct/sapolicies/db2/db2V10_stop.ksh db2ap 0"
        MonitorCommand = "/usr/sbin/rsct/sapolicies/db2/db2V10_monitor.ksh db2ap 0"
```

★ `$ lsrsrc –Ab IBM.Application Name StartCommand StartCommandTimeout`

```
resource 8:
        Name               = "db2mnt-db2home-rs"
        StartCommand       = "/usr/sbin/rsct/sapolicies/db2/mountV10_start.ksh /db2home"
        StartCommandTimeout = 900
resource 9:
        Name               = "db2_db2ap_0-rs"
        StartCommand       = "/usr/sbin/rsct/sapolicies/db2/db2V10_start.ksh db2ap 0"
        StartCommandTimeout = 900
```

```
# chrsrc –s "Name=='db2_db2ap_0-rs'" IBM.Application StartCommandTimeout=60
```


High Availability – Active / Passive
Node Detection Failure

★ **$ lscomg**
$ lscomg –i CG1

```
db2psc@node03:~> lscomg
Name Sensitivity Period Priority Broadcast SourceRouting NIMPathName NIMParameters Grace        MediaType
CG1  4           1      1        Yes       Yes                                    -1 (Default) 1 (IP)
db2psc@node03:~> lscomg -i CG1
Name NodeName              IPAddress       Subnet      SubnetMask
eth0 node02.hadr.ibm.local 192.168.142.102 192.168.0.0 255.255.0.0
eth0 node03.hadr.ibm.local 192.168.142.103 192.168.0.0 255.255.0.0
```

★ Time to detect Node Network Failure is `Sensitivity` x (`Period` x 2) i.e. 8 seconds

★ Grace period is time in seconds when RSCT sends ICMP echo to check if system is reachable or not.
- ★ If that node responds within that grace period, RSCT will not detect this as a node failure.
- ★ Grace period is to protect from memory or CPU starvation issues.

```
# chcomg –s 10 –p 3 –g 30 CG1
```

```
Name Sensitivity Period Priority Broadcast SourceRouting NIMPathName NIMParameters Grace
CG1  10          3      1        Yes       Yes                                    30
db2psc@node03:~>
```

★ Node Detection Failure is now i.e. 10 x (3 x 2) = 60 seconds

High Availability – Active / Passive
Virtual IP Address – Application Transparency

★ `eth0:0` is the floating Virtual IP Address managed by RSCT

```
db2ap@node03:~> ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0C:29:58:33:A5
          inet addr:192.168.142.103  Bcast:192.168.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe58:33a5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:421757 errors:0 dropped:0 overruns:0 frame:0
          TX packets:775957 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:239688105 (228.5 Mb)  TX bytes:635310867 (605.8 Mb)

eth0:0    Link encap:Ethernet  HWaddr 00:0C:29:58:33:A5
          inet addr:192.168.142.100  Bcast:192.168.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

★ **$ lsrsrc –Ab ServiceIP**

```
db2ap@node03:~> lsrsrc -Ab IBM.ServiceIP Name IPAddress
Resource Persistent Attributes for IBM.ServiceIP
resource 1:
        Name      = "db2ip_192_168_142_100-rs"
        IPAddress = "192.168.142.100"
resource 2:
        Name      = "db2ip_192_168_142_100-rs"
        IPAddress = "192.168.142.100"
db2ap@node03:~>
```


High Availability – Active / Passive
Automatic Client Reroute

★ Even with Floating IP Address, Automatic Client Reroute settings at the driver level is must for the driver to wait and recover from failover

```
### Properties for the Automatic Client Reroute
#enableSysplexWLB=true
enableSeamlessFailover=true
enableConnectionConcentrator=true
clientRerouteAlternateServerName=192.168.142.100
clientRerouteAlternatePortNumber=50001
enableClientAffinitiesList=1
maxRetriesForClientReroute=10
retryIntervalForClientReroute=30
```

★ Set ACR Properties
- `enableClientAffilitiesList`
- `clientRerouteAlternateServerName`
- `clientRerouteAlternatePortNumber`

★ The driver will wait `10x30=300` seconds for a failover to complete. Application will wait for 5 minutes for the `Passive` to become `Active`

★ No application change required

High Availability – Active / Passive
Failover

★ The `Active` / `Passive` failover can take from 3-5 minutes
- Mount the floating file systems
- Enable Virtual IP Address
- Start DB2 and crash recovery, if any

★ If `db2sysc` process fails on the `Active`, a local restart of DB2 will be initiated by the RSCT without causing an actual failover

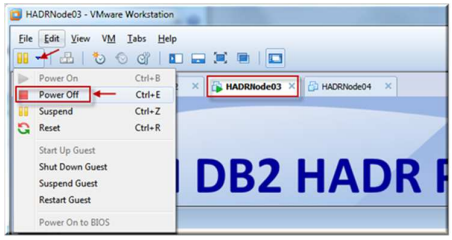
★ The quorum device must be configured for the failover to occur

★ **`$ lsrsrc –Ab IBM.TieBreaker`**
`$ lsrsrc –c IBM.PeerNode`

```
db2ap@node03:/pot_hadr/03haactivepassive> lsrsrc -c IBM.PeerNode OpQuorumTieBrea
ker
Resource Class Persistent Attributes for IBM.PeerNode
resource 1:
        OpQuorumTieBreaker = "db2_Quorum_Network_192_168_142_2:21_57_19"
db2ap@node03:/pot_hadr/03haactivepassive>
```


High Availability – Active / Passive
Failover – Power off the Machine

- ★ Power off the Active Machine



- ★ RSCT detects the node failure and initiates the transfer of the floating resources
- ★ **`$ lssam | grep node02`**

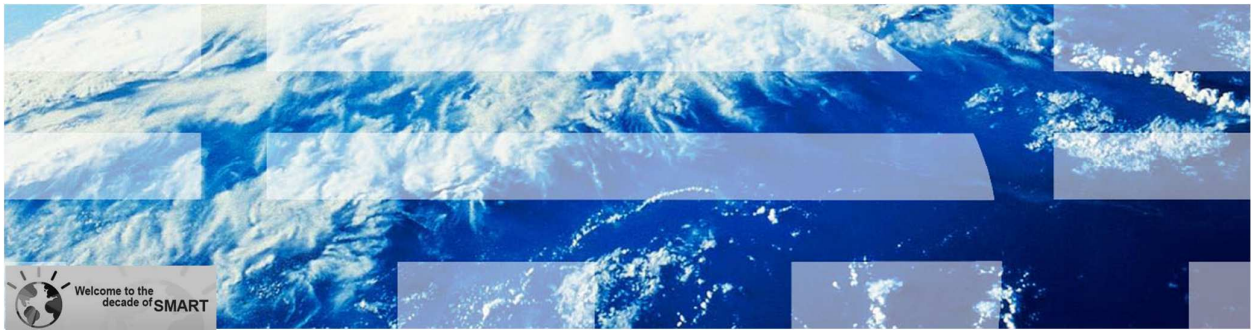
```
db2psc@node02:~/pot_hadr/03haactivepassive> lssam | grep node02
                |- Online IBM.Application:db2_db2ap_0-rs:node02
                |- Online IBM.Application:db2mnt-db2data1-rs:node02
                |- Online IBM.Application:db2mnt-db2data2-rs:node02
                |- Online IBM.Application:db2mnt-db2data3-rs:node02
                |- Online IBM.Application:db2mnt-db2home-rs:node02
                |- Online IBM.Application:db2mnt-db2log-rs:node02
                |- Online IBM.ServiceIP:db2ip_192_168_142_100-rs:node02
        '- Online IBM.PeerNode:node02:node02
        '- Online IBM.NetworkInterface:eth0:node02
```


High Availability – Active / Passive
Power on failed machine

- ★ Power on the failed machine
- ★ RSCT starts the process of reintegration of the node into the cluster.
- ★ **`$ lssam`**

```
Online IBM.ResourceGroup:db2_db2ap_0-rg Nominal=Online
        |- Online IBM.Application:db2_db2ap_0-rs
                |- Online IBM.Application:db2_db2ap_0-rs:node02
                '- Offline IBM.Application:db2_db2ap_0-rs:node03
        |- Online IBM.Application:db2mnt-db2data1-rs
                |- Online IBM.Application:db2mnt-db2data1-rs:node02
                '- Offline IBM.Application:db2mnt-db2data1-rs:node03
        |- Online IBM.Application:db2mnt-db2data2-rs
                |- Online IBM.Application:db2mnt-db2data2-rs:node02
                '- Offline IBM.Application:db2mnt-db2data2-rs:node03
        |- Online IBM.Application:db2mnt-db2data3-rs
                |- Online IBM.Application:db2mnt-db2data3-rs:node02
                '- Offline IBM.Application:db2mnt-db2data3-rs:node03
        |- Online IBM.Application:db2mnt-db2home-rs
                |- Online IBM.Application:db2mnt-db2home-rs:node02
                '- Offline IBM.Application:db2mnt-db2home-rs:node03
        |- Online IBM.Application:db2mnt-db2log-rs
                |- Online IBM.Application:db2mnt-db2log-rs:node02
                '- Offline IBM.Application:db2mnt-db2log-rs:node03
        '- Online IBM.ServiceIP:db2ip_192_168_142_100-rs
                |- Online IBM.ServiceIP:db2ip_192_168_142_100-rs:node02
                '- Offline IBM.ServiceIP:db2ip_192_168_142_100-rs:node03
Online IBM.Equivalency:db2_db2ap_0-rg_group-equ
        |- Online IBM.PeerNode:node03:node03
        '- Online IBM.PeerNode:node02:node02
Online IBM.Equivalency:db2_private_network_0
        |- Online IBM.NetworkInterface:eth0:node03
        '- Online IBM.NetworkInterface:eth0:node02
```

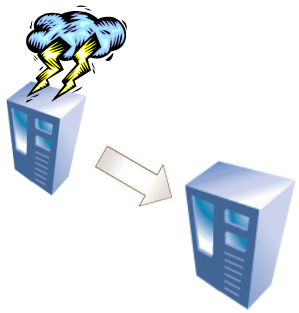

HA (Active / Passive) & Disaster Recovery



17 slides
Vikram S Khatri

High Availability & Disaster Recovery
Basic Principles

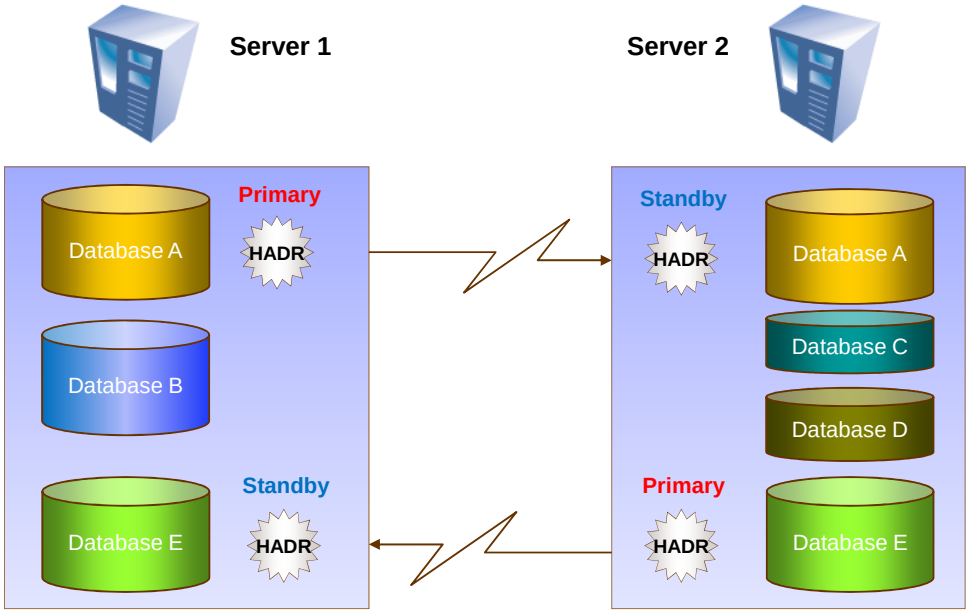
- ★ Two Active Machines
  - ➢ Primary
    - Processes transactions using database replication
    - Ships log buffer entries to the standby machine
  - ➢ Standby
    - Cloned from the primary
    - Receives and stores log buffer entries from the primary
    - Re-applies the transactions
    - Stays in a roll-forward mode



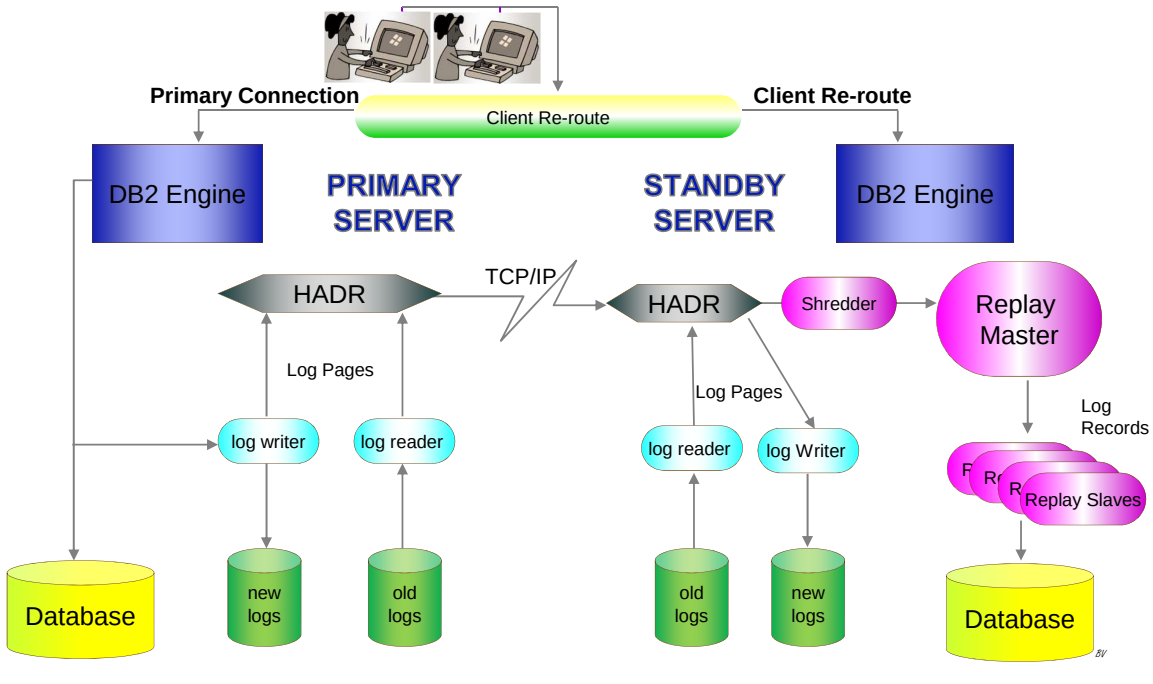
- ★ If the primary fails, the standby can take over the transactional workload
  - ➢ The standby becomes the new primary
- ★ If the failed machine becomes available again, it can be resynchronized
  - ➢ The old primary becomes the new standby

High Availability & Disaster Recovery
Scope of Action

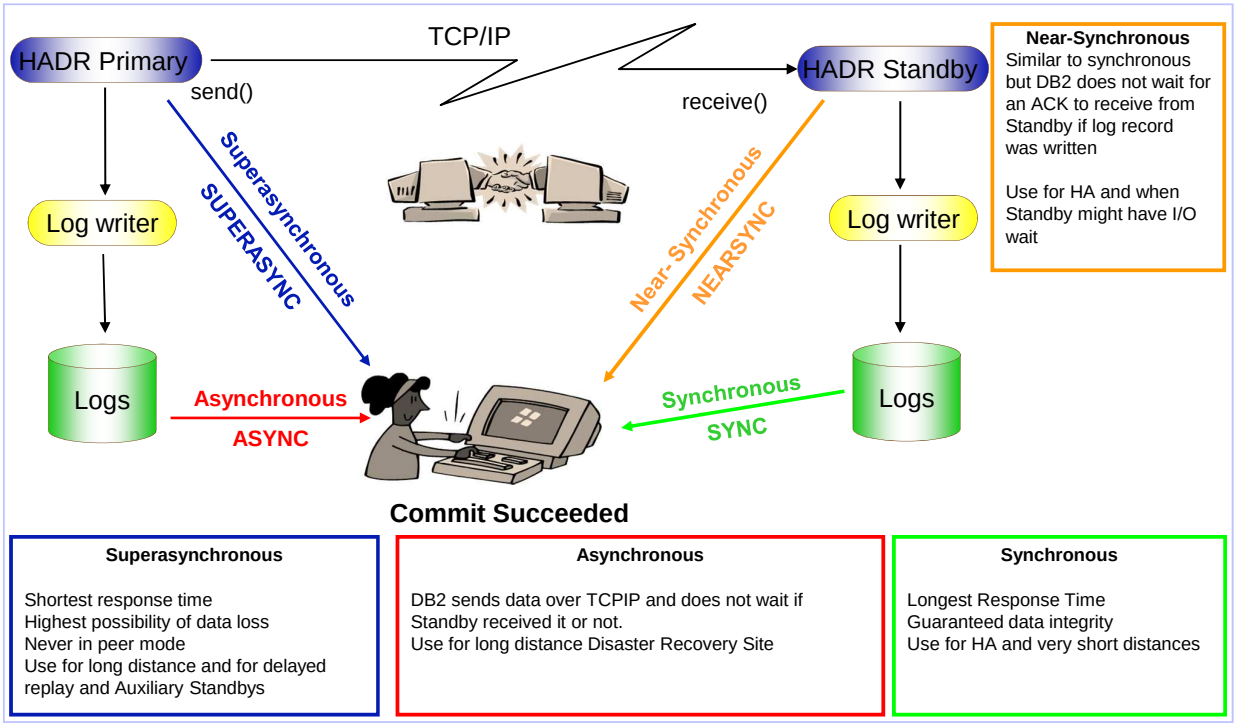
★ HADR replication takes place at the database level



High Availability & Disaster Recovery
Architecture



High Availability & Disaster Recovery
Synchronization Modes



High Availability & Disaster Recovery
Primary Configuration Parameters

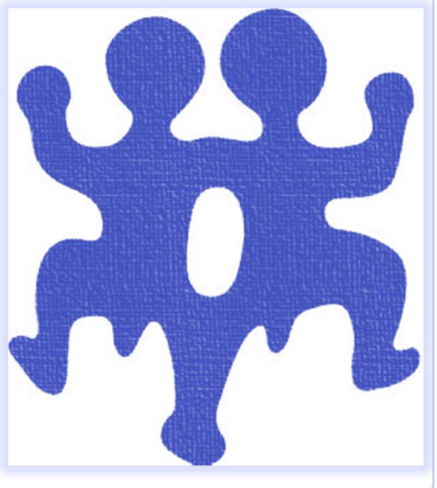
★ **Update DB CFG to Enable HADR on Primary**

```
UPDATE DB CFG FOR DB HADR USING LOGINDEXBUILD ON;

UPDATE DB CFG FOR DB HADR USING INDEXREC RESTART;

UPDATE DB CFG FOR HADR USING
    HADR_LOCAL_HOST         192.168.142.100
    HADR_LOCAL_SVC          51600
    HADR_REMOTE_HOST        192.168.142.104
    HADR_REMOTE_SVC         51800
    HADR_REMOTE_INST        db2ap
    HADR_SYNCMODE           NEARSYNC
    HADR_TIMEOUT            120
;
```


High Availability & Disaster Recovery
Clone the Primary

★ **Clone the primary**

- The target database (referred to as a standby) is created using DB2 restore facility, flash copy, or by split mirror
- After the restore, the standby is placed in “perpetual roll forward” mode
- Strict symmetry of table space and container configuration is required on the standby.
  - Name, path, size all must match.
  - Relative container paths are allowed, and the full path may differ in this case.
- If HADR fails to replicate a container operation on the standby.
  - Then replication of the affected table space stops and the table space will be left in "roll forward in progress" .



High Availability & Disaster Recovery
Standby Configuration Parameters

★ **Update DB CFG to Enable HADR on the Standby**

```
UPDATE DB CFG FOR HADR USING
    HADR_LOCAL_HOST         192.168.142.104
    HADR_LOCAL_SVC          51800
    HADR_REMOTE_HOST        192.168.142.100
    HADR_REMOTE_SVC         51600
    HADR_REMOTE_INST        db2ap
    HADR_SYNCMODE           NEARSYNC
    HADR_TIMEOUT            120
;
```


High Availability & Disaster Recovery
Start the Standby

★ **Start the Standby**

```
db2 deactivate db hadb
db2 start hadr on database as standby
```


High Availability & Disaster Recovery
Start the Primary

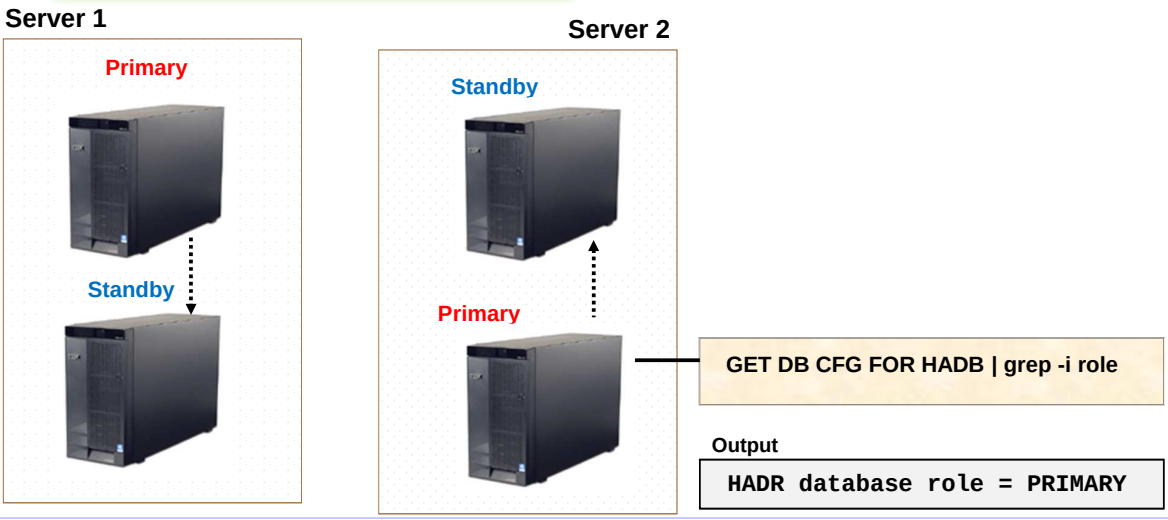
★ **Start the Primary**

```
db2 start hadr on database as primary
```


High Availability & Disaster Recovery
Mutual Takeover

★ **Takeover command always needs to be executed on the Standby**

```
db2 takeover hadr on database
```



High Availability & Disaster Recovery
Shutdown / Disable HADR

★ If you want to shutdown the HADR operation, the recommended procedure is (if you want to maintain the role):

Primary:
```
db2 deactivate db hadb
db2stop
```

Standby:
```
db2 deactivate db hadb
db2stop
```

★ Stop HADR on Primary First

```
db2 stop hadr on database hadb
```

★ Stop HADR on Standby

```
db2 deactivate db hadb
db2 stop hadr on database hadb
```

In order to restart a standby as standby again after a STOP HADR is issued, a new restore or DBINIT must be done.

High Availability & Disaster Recovery
Monitoring

★ `$ db2pd -hadr -d hadb`

```
db2ap@node02:/home/db2psc/pot_hadr/04hadr> ./hadr08

Database Member 0 -- Database HADB -- Active -- Up 0 days 00:07:21 -- Date 05/24
/2012 13:31:01

                            HADR_ROLE = PRIMARY
                        REPLAY_TYPE = PHYSICAL
                      HADR_SYNCMODE = NEARSYNC
                         STANDBY_ID = 1
                      LOG_STREAM_ID = 0
                         HADR_STATE = PEER
                  PRIMARY_MEMBER_HOST = 192.168.142.100
                     PRIMARY_INSTANCE = db2ap
                       PRIMARY_MEMBER = 0
                  STANDBY_MEMBER_HOST = 192.168.142.104
                     STANDBY_INSTANCE = db2ap
                        STANDB
                  HADR_CONNECT
            HADR_CONNECT_STA
         HEARTBEAT_INTERVAL(
                HADR_TIMEOUT(
       TIME_SINCE_LAST_RECV(
```

```
db2ap@node04:/home/db2psc/pot_hadr/04hadr> ./hadr08

Database Member 0 -- Database HADB -- Standby -- Up 0 days 00:09:33 -- Date 05/2
4/2012 13:32:49

                            HADR_ROLE = STANDBY
                        REPLAY_TYPE = PHYSICAL
                      HADR_SYNCMODE = NEARSYNC
                         STANDBY_ID = 0
                      LOG_STREAM_ID = 0
                         HADR_STATE = PEER
                  PRIMARY_MEMBER_HOST = 192.168.142.100
                     PRIMARY_INSTANCE = db2ap
                       PRIMARY_MEMBER = 0
                  STANDBY_MEMBER_HOST = 192.168.142.104
                     STANDBY_INSTANCE = db2ap
                       STANDBY_MEMBER = 0
                  HADR_CONNECT_STATUS = CONNECTED
             HADR_CONNECT_STATUS_TIME = 05/24/2012 13:23:44.605064 (1337880224)
           HEARTBEAT_INTERVAL(seconds) = 30
```


High Availability & Disaster Recovery
Automated Failover

★ We used `db2haicu` to automate failover between `Active` / `Passive` but `db2haicu` does not have support for automated failover between `Active` / `Passive` and a DR site.

★ This does not mean that it can not be accomplished. With RSCT commands, this can be accomplished.

★ To do the failover between `Active` / `Passive` and a DR site, the DBA must execute `db2 takeover hadr on database by force` manually on the standby.

High Availability & Disaster Recovery
Automatic Client Reroute

★ Driver Level ACR Properties

```
### Properties for the Automatic Client Reroute
#enableSysplexWLB=true
#enableConnectionConcentrator=true
enableSeamlessFailover=true
clientRerouteAlternateServerName=192.168.142.100,192.168.142.104
clientRerouteAlternatePortNumber=50001,50001
enableClientAffinitiesList=1
maxRetriesForClientReroute=20
retryIntervalForClientReroute=3
```

★ Set ACR Properties

- `enableClientAffilitiesList`
- `clientRerouteAlternateServerName`
- `clientRerouteAlternatePortNumber`

★ The driver will wait maximum `20x3=60` seconds and attempt 20 retries at an interval of 3 seconds.

High Availability Disaster Recovery & Read on Standby

HADR and Read on Standby
Configure HADR Between Primary and Standby - Recap

- ★ Prepare Primary – Create Instance and Database
- ★ Update DB CFG for HADR
- ★ Take Full Backup
- ★ Prepare Standby – Create Instance and Restore Database
- ★ Update DB CFG for HADR
- ★ Start HADR on Standby
- ★ Start HADR on Primary

HADR and Read on Standby
Enable Read on Standby Configuration

- ★ On Standby Database set DB2 registry variables

```
db2set db2_standby_iso=UR
db2set db2_standby_ros=ON
```

- ★ Set same DB2 registry variables on the Primary

```
db2set db2_standby_iso=UR
db2set db2_standby_ros=ON
```

These registry variables are ignored on the Primary and are used when it becomes the Standby

HADR and Read on Standby
Configure TSA on the Standby

- ★ **# preprpnode node01 node02** – On first host
  # preprpnode node01 node02 – On second host
- ★ Run `db2haicu` on the Standby first. It does the following:
  - Creates a HA domain
  - Adds 2 nodes to the domain
  - Uses gateway as a quorum device
  - Adds Ethernet interfaces to the RSCT resources
  - Changes DBM CFG cluster manager configuration on Standby to TSA
  - Configuration completes with the following message.

```
Adding HADR database 'HADB' to the domain ...
HADR database 'HADB' has been determined to be valid for high availability. Howe
ver, the database cannot be added to the cluster from this node because db2haicu
 detected this node is the standby for HADR database 'HADB'. Run db2haicu on the
 primary for HADR database 'HADB' to configure the database for automated failov
er.
All cluster configurations have been completed successfully. db2haicu exiting ..
.
db2psc@node02:/pot_hadr/05hadrros>
```


HADR and Read on Standby
Configure TSA on the Primary

- ★ Run `db2haicu` on the Primary. It does the following:
  - ➢ It looks for an existing HA domain which was created on the Standby and is available on the Primary
  - ➢ Changes DBM CFG cluster manager configuration on Primary to TSA
  - ➢ Automates HADR failover
  - ➢ Configures a Virtual IP Address for the Primary, which is a floating resource available to whosoever the primary is
  - ➢ Cluster configuration completes with the following similar message

```
Adding virtual IP address '192.168.142.100' to the domain ...
Adding virtual IP address '192.168.142.100' to the domain was successful.
All cluster configurations have been completed successfully. db2haicu exiting ..
.
db2psc@node01:/pot_hadr/05hadrros>
```


HADR and Read on Standby
Verify Cluster Resources

★ **`lssam`**

```
db2psc@node01:/pot_hadr/05hadrros> lssam
Online IBM.ResourceGroup:db2_db2psc_db2psc_HADB-rg Nominal=Online
        |- Online IBM.Application:db2_db2psc_db2psc_HADB-rs
                |- Online IBM.Application:db2_db2psc_db2psc_HADB-rs:node01
                '- Offline IBM.Application:db2_db2psc_db2psc_HADB-rs:node02
        '- Online IBM.ServiceIP:db2ip_192_168_142_100-rs
                |- Online IBM.ServiceIP:db2ip_192_168_142_100-rs:node01
                '- Offline IBM.ServiceIP:db2ip_192_168_142_100-rs:node02
Online IBM.ResourceGroup:db2_db2psc_node01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2psc_node01_0-rs
                '- Online IBM.Application:db2_db2psc_node01_0-rs:node01
Online IBM.ResourceGroup:db2_db2psc_node02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2psc_node02_0-rs
                '- Online IBM.Application:db2_db2psc_node02_0-rs:node02
Online IBM.Equivalency:db2_db2psc_db2psc_HADB-rg_group-equ
        |- Online IBM.PeerNode:node01:node01
        '- Online IBM.PeerNode:node02:node02
Online IBM.Equivalency:db2_db2psc_node01_0-rg_group-equ
        '- Online IBM.PeerNode:node01:node01
Online IBM.Equivalency:db2_db2psc_node02_0-rg_group-equ
        '- Online IBM.PeerNode:node02:node02
Online IBM.Equivalency:db2_private_network_0
        |- Online IBM.NetworkInterface:eth0:node01
        '- Online IBM.NetworkInterface:eth0:node02
db2psc@node01:/pot_hadr/05hadrros>
```


HADR and Read on Standby
Configure VIP for the Standby

★ `db2haicu` does not create VIP for the Read on Standby.

★ White Paper http://www-304.ibm.com/support/docview.wss?uid=swg27020912

- Explains the RSCT commands used to configure a VIP for the standby

```
export CT_MANAGEMENT_SCOPE=2

echo "-----------------------------------------------------"
echo "Creating Resource hadr_ros_01-rs"
echo "-----------------------------------------------------"
mkrsrc -f hadrros11.def IBM.Application

echo "-----------------------------------------------------"
echo "Creating Resource hadr_ros_02-rs"
echo "-----------------------------------------------------"
mkrsrc -f hadrros12.def IBM.Application

echo "-----------------------------------------------------"
echo "Equivalency hadr_ros_equiv between hadr_ros_01-rs and hadr_ros_02-rs"
echo "-----------------------------------------------------"
mkequ -p NoControl hadr_ros_equiv IBM.Application:hadr_ros_01-rs:node01,hadr_ros
_02-rs:node02

echo "-----------------------------------------------------"
echo "Create Resource ros_ip_192_168_142_200-rs"
echo "-----------------------------------------------------"
mkrsrc -f hadrros13.def IBM.ServiceIP

echo "-----------------------------------------------------"
echo "Create Resource Group hadr_ros-rg"
echo "-----------------------------------------------------"
mkrg hadr_ros-rg

echo "-----------------------------------------------------"
echo "Add ros_ip_192_168_142_200-rs to the hadr_ros-rg"
echo "-----------------------------------------------------"
addrgmbr -g hadr_ros-rg IBM.ServiceIP:ros_ip_192_168_142_200-rs

echo "-----------------------------------------------------"
echo "Relation between ros_ip_192_168_142_200-rs and hadr_ros_equiv"
echo "-----------------------------------------------------"
mkrel -p DependsOn -S IBM.ServiceIP:ros_ip_192_168_142_200-rs -G IBM.Equivalency
:hadr_ros_equiv hadr_ros_relation
```


HADR and Read on Standby
Verify Additional Cluster Resources

★ **lssam**

```
db2psc@node01:/pot_hadr/05hadrros> lssam
Online IBM.ResourceGroup:db2_db2psc_db2psc_HADB-rg Request=Online Nominal=Online
        |- Online IBM.Application:db2_db2psc_db2psc_HADB-rs
                |- Online IBM.Application:db2_db2psc_db2psc_HADB-rs:node01
                '- Offline IBM.Application:db2_db2psc_db2psc_HADB-rs:node02
        '- Online IBM.ServiceIP:db2ip_192_168_142_100-rs
                |- Online IBM.ServiceIP:db2ip_192_168_142_100-rs:node01
                '- Offline IBM.ServiceIP:db2ip_192_168_142_100-rs:node02
Online IBM.ResourceGroup:db2_db2psc_node01_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2psc_node01_0-rs
                '- Online IBM.Application:db2_db2psc_node01_0-rs:node01
Online IBM.ResourceGroup:db2_db2psc_node02_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2psc_node02_0-rs
                '- Online IBM.Application:db2_db2psc_node02_0-rs:node02
Offline IBM.ResourceGroup:hadr_ros-rg Nominal=Offline
        '- Offline IBM.ServiceIP:ros_ip_192_168_142_200-rs
                |- Offline IBM.ServiceIP:ros_ip_192_168_142_200-rs:node01
                '- Offline IBM.ServiceIP:ros_ip_192_168_142_200-rs:node02
Online IBM.Equivalency:db2_db2psc_db2psc_HADB-rg_group-equ
        |- Online IBM.PeerNode:node01:node01
        '- Online IBM.PeerNode:node02:node02
Online IBM.Equivalency:db2_db2psc_node01_0-rg_group-equ
        '- Online IBM.PeerNode:node01:node01
Online IBM.Equivalency:db2_db2psc_node02_0-rg_group-equ
        '- Online IBM.PeerNode:node02:node02
Online IBM.Equivalency:db2_private_network_0
        |- Online IBM.NetworkInterface:eth0:node01
        '- Online IBM.NetworkInterface:eth0:node02
Online IBM.Equivalency:hadr_ros_equiv
        |- Offline IBM.Application:hadr_ros_01-rs:node01
        '- Online IBM.Application:hadr_ros_02-rs:node02
db2psc@node01:/pot_hadr/05hadrros>
```

New Resource group created but still offline.

```
# chrg -o online hadr_ros-rg
```

```
Online IBM.ResourceGroup:hadr_ros-rg Nominal=Online
        '- Online IBM.ServiceIP:ros_ip_192_168_142_200-rs
                |- Offline IBM.ServiceIP:ros_ip_192_168_142_200-rs:node01
                '- Online IBM.ServiceIP:ros_ip_192_168_142_200-rs:node02
```

VIP is active on node02

HADR and Read on Standby
Check VIP on the Standby

★ **# ifconfig -a**

```
db2psc@node02:/etc/sysconfig/network> /sbin/ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0C:29:16:14:9F
          inet addr:192.168.142.102  Bcast:192.168.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe16:149f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:173660 errors:0 dropped:0 overruns:0 frame:0
          TX packets:196596 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88378446 (84.2 Mb)  TX bytes:20863301 (19.8 Mb)

eth0:0    Link encap:Ethernet  HWaddr 00:0C:29:16:14:9F
          inet addr:192.168.142.200  Bcast:192.168.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:148 errors:0 dropped:0 overruns:0 frame:0
          TX packets:148 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22412 (21.8 Kb)  TX bytes:22412 (21.8 Kb)
```


HADR and Read on Standby
Check VIP on the Primary

★ **# ifconfig -a**

```
db2psc@node01:/pot_hadr/05hadrros> /sbin/ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:50:56:31:74:BE
          inet addr:192.168.142.101  Bcast:192.168.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:fe31:74be/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:116255 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85855 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13454908 (12.8 Mb)  TX bytes:23931687 (22.8 Mb)

eth0:0    Link encap:Ethernet  HWaddr 00:50:56:31:74:BE
          inet addr:192.168.142.100  Bcast:192.168.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:107 errors:0 dropped:0 overruns:0 frame:0
          TX packets:107 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9076 (8.8 Kb)  TX bytes:9076 (8.8 Kb)
```


HADR and Read on Standby
Disable

```
db2psc@node01:/pot_hadr/05hadrros> db2 get dbm cfg | grep -i cluster
 Cluster manager                                     = TSA
db2psc@node01:/pot_hadr/05hadrros> ssh node02 db2 get dbm cfg | grep -i cluster
 Cluster manager                                     = TSA
db2psc@node01:/pot_hadr/05hadrros>
```

★ Run **db2haicu –disable** on the Standby. It locks the resources and removes TSA as a cluster manager of the DB2 server.

```
db2psc@node01:/pot_hadr/05hadrros> db2 get dbm cfg | grep -i cluster
 Cluster manager                                     =
db2psc@node01:/pot_hadr/05hadrros>
```

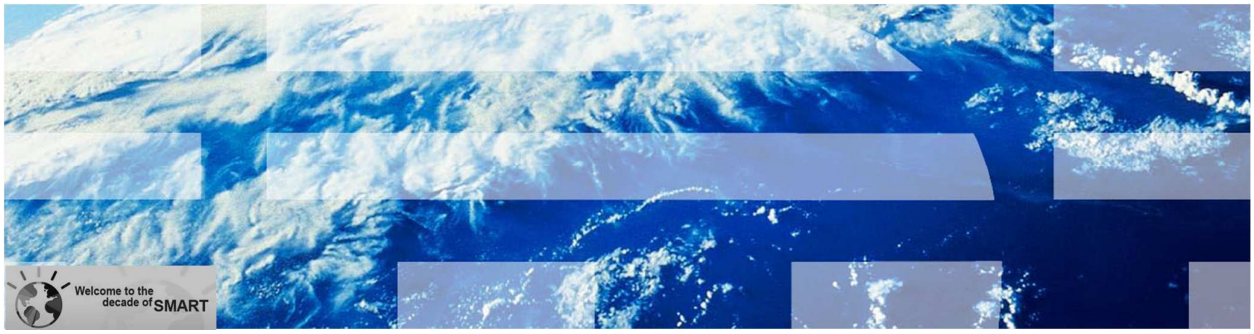
This is unset now

★ Run **db2haicu –disable** on the Primary. It locks the resources and removes TSA as a cluster manager of the DB2 server.

```
db2psc@node02:/pot_hadr/05hadrros> db2haicu -disable
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu).
```

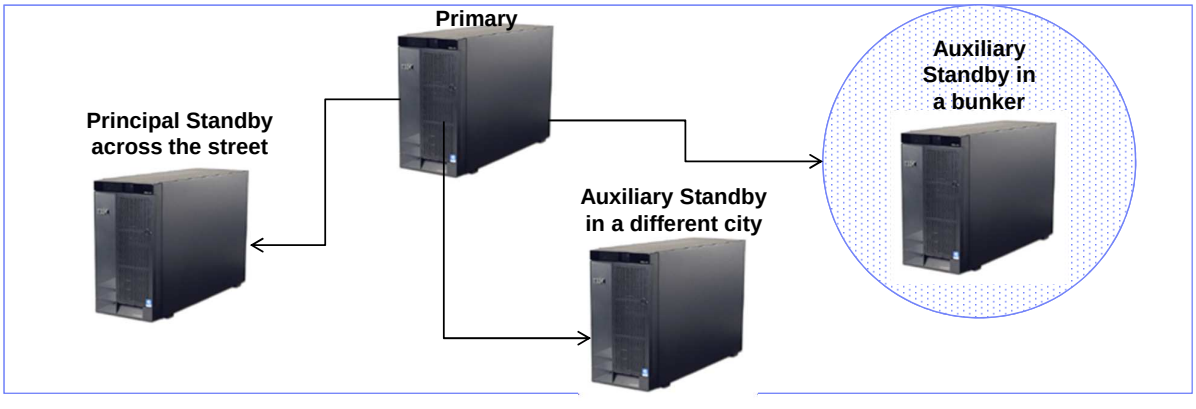
★ **# export CT_MANAGEMENT_SCOPE=2**
 # stoprpdomain –f db2ha

High Availability Disaster Recovery & Multiple Standbys



HADR and Multiple Standbys Architecture

- ★ Standby is now known as Principal Standby and supports any synchronization mode
- ★ Up to two Auxiliary Standbys are supported.
- ★ Auxiliary Standbys supports only SUPERASYNC mode
- ★ Standbys are fed directly from the Primary (No daisy chaining)
- ★ Read on Standby is supported on all Standbys



HADR and Multiple Standbys
Configuration

- ★ Take full backup of the database on the Primary.
- ★ Restore database on the Auxiliary Standbys
- ★ Update DB CFG parameters on the Standbys for HADR
  - `hadr_local_host`
  - `hadr_local_svc`
  - `hadr_remote_host`
  - `hadr_remote_inst`
  - `hadr_remote_svc`
  - **`hadr_target_list`**
- ★ Additionally `HADR_SYNCMODE` needs to be set to `SUPERASYNC` on the Auxiliary Standbys
- ★ Start HADR on all Standbys
- ★ Start HADR on Primary

HADR and Multiple Standbys
SUPERASYNC Mode

- ★ SUPERASYNC mode is available in DB2 10.1, 9.7 FP 5, 9.5 FP 8
- ★ SYNC, NEARSYNC and ASYNC mode can lead to a back pressure on the Primary due to slow network or if Standby is not able to keep up
- ★ SUPERASYNC mode prevents back pressure by never entering peer state
  - Standby moves from local catch-up to the remote catch-up only
  - HADR will always ship primary's on-disk logs or archived logs
- ★ Committed considered successful as soon as the log buffer has been written to the disk on the Primary
  - Does not wait for a log buffer `send()` to the standby database
- ★ Best performance but least protection
- ★ Normal TAKEOVER is supported even though the Standby state is remote catch-up pending

HADR and Multiple Standbys
HADR_TARGET_LIST

- DB CFG parameter HADR_TARGET_LIST used to define all Standbys
- Lists up to 3 hosts with port number delimited by '|'
  - Example: Node02:51801|Node03:51602:Node04:51802
- All Standbys must have this parameter defined
- This parameter on Standbys must include the current Primary
- The list defined on the Primary has first entry for the Principal Standby with up to two auxiliary standby listed after it.
- After a TAKEOVER, standbys listed in the target list are automatically reconfigured for
  - HADR_REMOTE_HOST
  - HADR_REMOTE_SVC
  - HADR_REMOTE_INST

HADR and Multiple Standbys
Monitor HADR

```
db2psc@node01:/pot_hadr/06hadrmulttarget> db2pd -hadr -d hadb | more

Database Member 0 -- Database HADB -- Active -- Up 0 days 00:09:47 -- Date 06/02
/2012 12:07:41

                            HADR_ROLE = PRIMARY
                        REPLAY_TYPE = PHYSICAL
                      HADR_SYNCMODE = NEARSYNC
                         STANDBY_ID = 1
                      LOG_STREAM_ID = 0
                         HADR_STATE = PEER
                  PRIMARY_MEMBER_HOST = node01
                     PRIMARY_INSTANCE = db2psc
                        PRIMARY_MEMBER = 0
                  STANDBY_MEMBER_HOST = node02
                     STANDBY_INSTANCE = db2psc
                        STANDBY_MEMBER = 0
                  HADR_CONNECT_STATUS = CONNECTED
             HADR_CONNECT_STATUS_TIME = 06/02/2012 11:57:55.438830 (1338652675)
  STANDBY_REPLAY_ONLY_WINDOW_ACTIVE = N

                            HADR_ROLE = PRIMARY
                        REPLAY_TYPE = PHYSICAL
                      HADR_SYNCMODE = SUPERASYNC
                         STANDBY_ID = 2
                      LOG_STREAM_ID = 0
                         HADR_STATE = REMOTE_CATCHUP
                  PRIMARY_MEMBER_HOST = node01
                     PRIMARY_INSTANCE = db2psc
                        PRIMARY_MEMBER = 0
                  STANDBY_MEMBER_HOST = node03
                     STANDBY_INSTANCE = db2psc
                        STANDBY_MEMBER = 0

                            HADR_ROLE = PRIMARY
                        REPLAY_TYPE = PHYSICAL
                      HADR_SYNCMODE = SUPERASYNC
                         STANDBY_ID = 3
                      LOG_STREAM_ID = 0
                         HADR_STATE = REMOTE_CATCHUP
                  PRIMARY_MEMBER_HOST = node01
                     PRIMARY_INSTANCE = db2psc
                        PRIMARY_MEMBER = 0
                  STANDBY_MEMBER_HOST = node04
                     STANDBY_INSTANCE = db2psc
                        STANDBY_MEMBER = 0
                  HADR_CONNECT_STATUS = CONNECTED
```

HADR and Multiple Standbys
Log Spooling

- DB CFG parameter `HADR_SPOOL_LIMIT` used for log spooling
- Spools log records arriving from the Primary for replay later
- Allows replay to fall behind the log shipping
- Decouples log replay on the Standby from receiving of the log data from the Primary
- Logs are spooled in the active log directory
- Supported with any synchronization modes
- Logs are deleted automatically when no longer required
- Takeover may take longer as draining of the spool must take place before takeover completes
- Log spooling will absorb load spikes in logging from the primary without causing a back pressure to the primary

HADR and Multiple Standbys
Time Delayed Apply on the Standby

- The objective of this feature is to protect against application error.
- HADR delayed replay helps prevent data loss due to the errant transactions.
- To implement HADR delayed replay, we need to set the HADR_REPLAY_DELAY parameter on the Auxiliary Standby database.
- Delayed replay intentionally keeps the standby database at a point in time that is earlier than that of the primary database by delaying replay of logs on that standby.
- If an errant transaction is executed on the primary, you have until the configured time delay has elapsed to take action to prevent the errant transaction from being replayed on the standby.
- A TAKEOVER command on a standby with replay delay enabled will fail. You must first set the HADR_REPLAY_DELAY configuration parameter to 0 and then deactivate and reactivate the standby to pick up the new value, and then issue the TAKEOVER command.
- The delayed replay feature is supported only in SUPERASYNC mode.

HADR and Multiple Standbys
Time Delayed Apply on the Standby (continued…)

- Look at the current STANDBY_REPLAY_LOG_TIME parameter from the $ db2pd –hadr –d hadb

```
db2psc@node03:/pot_hadr/06hadrmulttarget> ssh node01 $PWD/mult17
                STANDBY_REPLAY_LOG_TIME = 06/02/2012 18:10:27.000000 (1338675027)
                STANDBY_REPLAY_LOG_TIME = 06/02/2012 18:10:27.000000 (1338675027)
                STANDBY_REPLAY_LOG_TIME = 06/02/2012 18:10:27.000000 (1338675027)
db2psc@node03:/pot_hadr/06hadrmulttarget>
```

- Enable HADR_REPLAY_DELAY and HADR_SPOOL_LIMIT
- Run application and monitor STANDBY_REPLAY_LOG_TIME parameter again.

```
db2psc@node03:/pot_hadr/06hadrmulttarget> ssh node01 $PWD/mult17
                STANDBY_REPLAY_LOG_TIME = 06/02/2012 19:43:29.000000 (1338680609)
                STANDBY_REPLAY_LOG_TIME = 06/02/2012 18:10:27.000000 (1338675027)
                STANDBY_REPLAY_LOG_TIME = 06/02/2012 19:43:29.000000 (1338680609)
db2psc@node03:/pot_hadr/06hadrmulttarget>
```

- The time delay apply is evident from the row count

```
db2psc@node03:/pot_hadr/06hadrmulttarget> ./mult15
Host = node01  PS_HISTORY Row Count =        74364
Host = node02  PS_HISTORY Row Count =        74364
Host = node03  PS_HISTORY Row Count =        70209
Host = node04  PS_HISTORY Row Count =        74364
db2psc@node03:/pot_hadr/06hadrmulttarget>
```

node03 is behind.

Grazie (ITALIAN) धन्यवाद (HINDI) Merci (FRENCH) ありがとうございました (JAPANESE) Obrigado (BRAZILIAN PORTUGUESE) 多谢 (SIMPLIFIED CHINESE)

Thank You

多謝 (TRADITIONAL CHINESE) Gracias (SPANISH) Спасибо (RUSSIAN) நன்றி (TAMIL) ขอบคุณ (THAI) Danke (GERMAN) شكراً (ARABIC)

Presentation By:

Vikram Khatri

IBM Senior Certified IT Specialist for DB2 Migrations
IBM Advanced Technical Expert DB2 for Clusters
IBM Certified Database Administrator for DB2 UDB V8.1 for Linux UNIX and Windows
IBM Certified Solutions Expert for DB2 UDB V8.1 Family Application Development
Project Management Professional Certified
IBM Plateau Level 1 Inventor
vikram.khatri@us.ibm.com