



# Achieving High Availability with DB2 HADR and TSAMP

Philip K. Gunning

Gunning Technology Solutions, LLC

# What is HADR?

- High Availability Disaster Recovery (HADR)
  - Introduced in DB2 V8.2
  - Log based replication using existing network infrastructure
  - Ported from Informix after acquired by IBM
  - Provides for high availability in same data center or remote data center
  - Many improvements over the years based on customer feedback and technology improvements

# HADR

- Bundled with all versions of DB2 except DB2 Express-C
- Easy to setup and monitor
- Provides additional flexibility for report only databases while providing a degree of high availability
- Multiple synchronous modes to choose some
- Synchronization mode should be chosen based on BUSINESS REQUIREMENTS
  - Many times the business doesn't know their requirements
  - That's where you provide recommendation based on analysis of application requirements

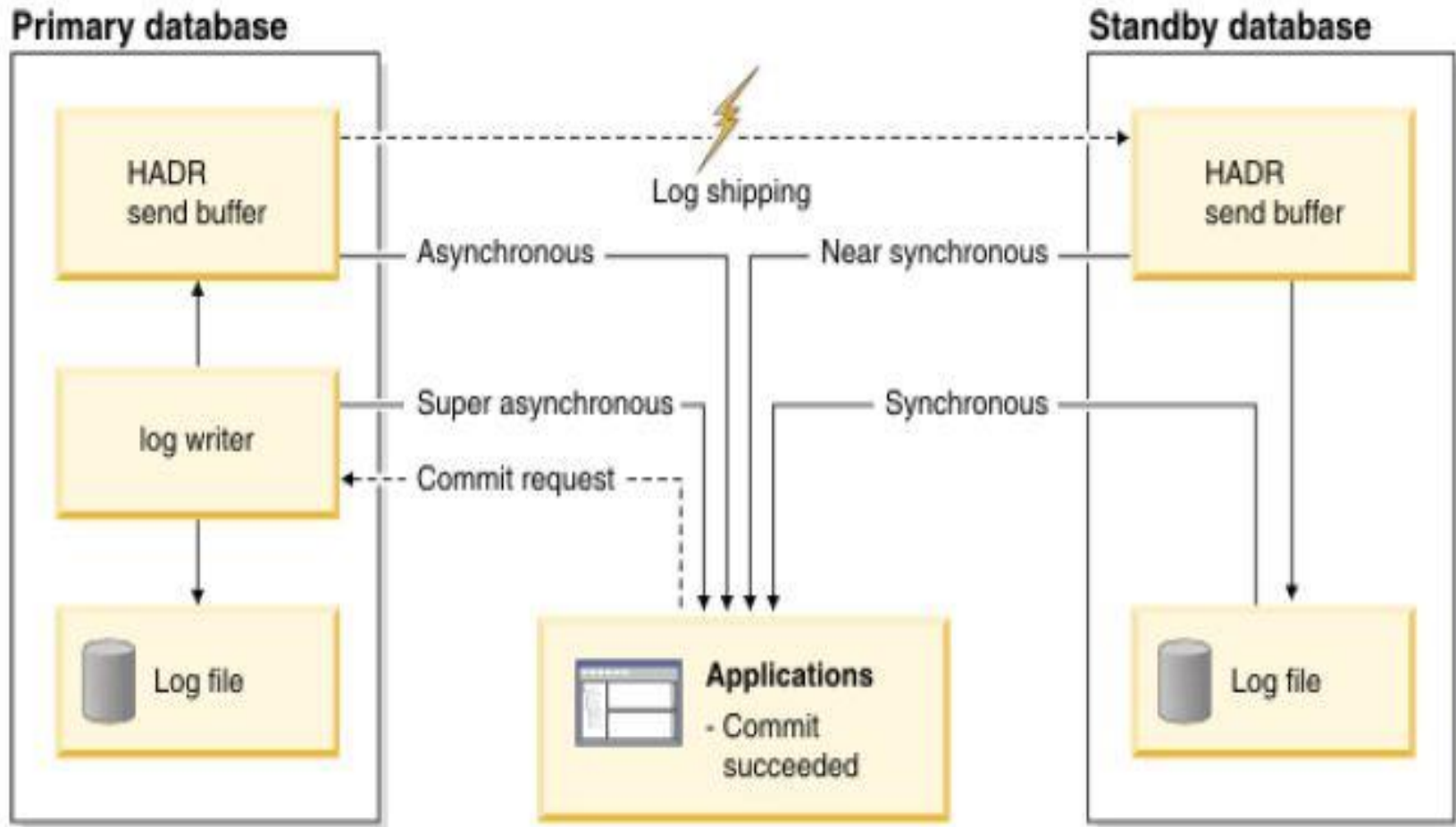
# HADR

- READ-ON-STANDBY (ROS) provided via registry variable in DB2 9.7
- Many customers use ROS to establish both an HA database and report only database
  - Eliminates need to do daily resotres, snapshots, etc to establish report only databases
  - Very convenient, flexible and efficient
- HADR Spooling added in DB2 10.1
- Multiple standby databases in DB2 10.1
  - Primary standby still known as standby
  - Additional “standby’s” known as auxilliaris

# HADR

- Delayed Replay introduced in DB2 10.1 provides protection against application errors
- Can have an auxiliary be delayed for a user-specified time behind the primary databases
- Enables auxiliary to be used to recover from application errors by bringing the auxiliary online before the rogue application error

# HADR Synchronization Modes



# Synchronization Modes

- **hadr\_syncmode** – Database Configuration Parameter that controls the HADR synchronization mode
- Four possible values:
  - SYNC
  - NEARSYNC
  - ASYNC
  - SUPERASYNC

# Hadr\_syncmode

- **SYNC** Mode
- SYNC mode offers the best protection of data. Two on-disk copies of data are required for transaction commit. The cost is the extra time for writing on the standby and sending an acknowledgment message back to the primary. In SYNC mode, logs are sent to the standby only after they are written to the primary disk. Log write and replication events happen sequentially. The total time for a log write event is the sum of (primary\_log\_write + log\_send + standby\_log\_write + ack\_message). The communication overhead of replication in this mode is significantly higher than that of the other three modes.



# Hadr\_syncmode

- **NEARSYNC** Mode
- NEARSYNC mode is nearly as good as SYNC, with significantly less communication overhead. In NEARSYNC mode, sending logs to the standby and writing logs to the primary disk are done in parallel, and the standby sends an acknowledgement message as soon as it receives the logs in memory. On a fast network, log replication causes no or little overhead to primary log writing. In NEARSYNC mode, you will lose data if the primary fails and the standby fails before it has a chance to write the received logs to disk. This is a relatively rare "double failure" scenario. Thus NEARSYNC is a good choice for many applications, providing near synchronization protection at far less performance cost.

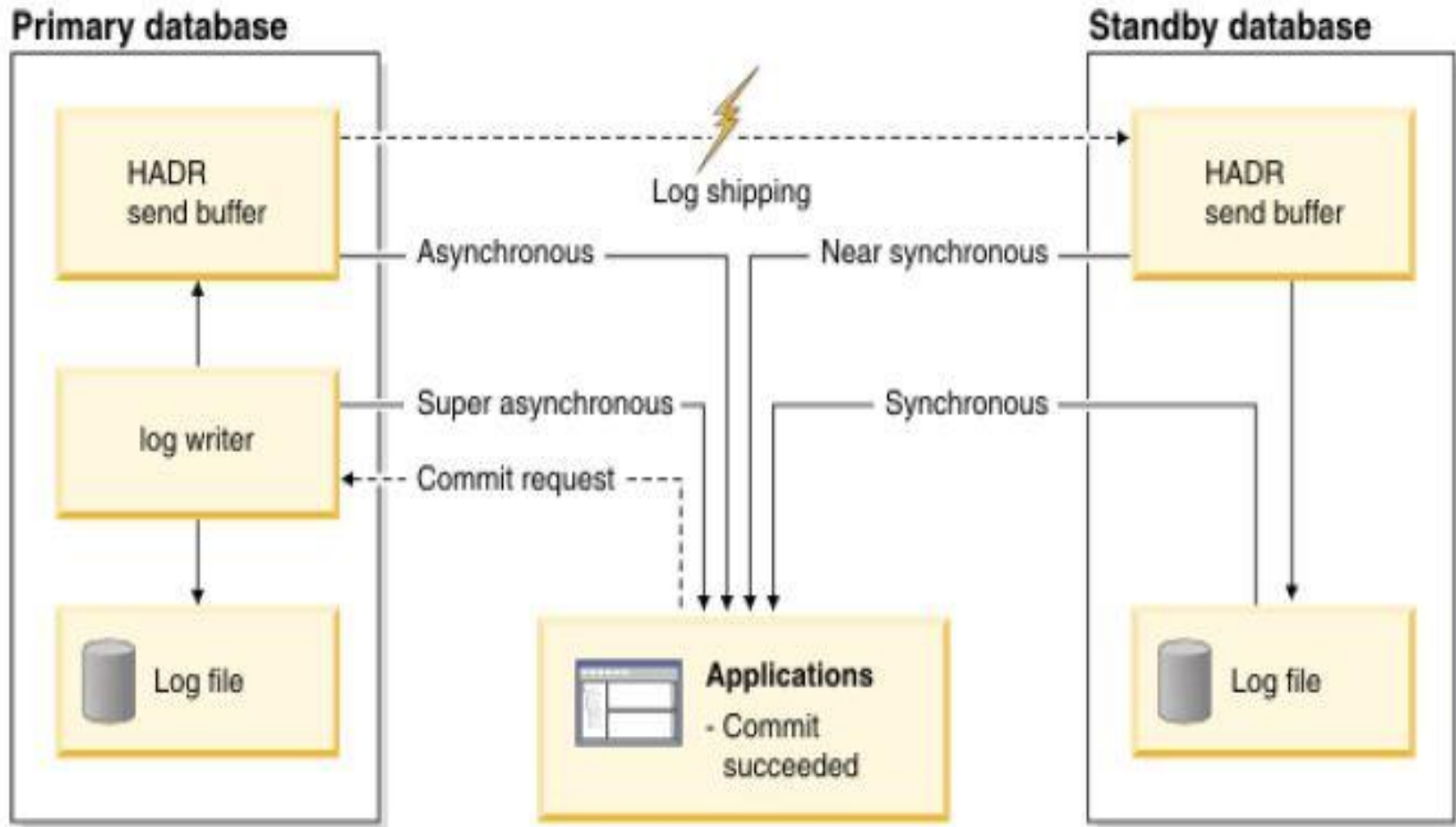
# Hadr\_syncmode

- **ASYNC** Mode
- In ASYNC mode, sending logs to the standby and writing logs to the primary disk are done in parallel, just like in NEARSYNC mode. Because ASYNC mode does not wait for acknowledgment messages from the standby, the primary system throughput is  $\min(\text{log write rate, log send rate})$ . ASYNC mode is well suited for WAN applications. Network transmission delay does not impact performance in this mode, but if the primary database fails, there is a higher chance that logs in transit will be lost (not replicated to standby).

# Hadr\_syncmode

- **SUPERASYNC** Mode
- This mode has the shortest transaction response time of all synchronization modes but has also the highest probability of transaction losses if the primary system fails. The primary system throughput is only affected by the time needed to write the transaction to the local disk. This mode is useful when you do not want transactions to be blocked or experience elongated response times due to network interruptions or congestion. SUPERASYNC mode is well suited for WAN applications. Since the transaction commit operations on the primary database are not affected by the relative slowness of the HADR network or the standby HADR server, the log gap between the primary database and the standby database might continue to increase. It is important to monitor the log gap in this mode as it is an indirect measure of the potential number of transactions that might be lost should a true disaster occur on the primary system.

# HADR Synchronization Modes



# Configuring Databases for HADR

- Before you can use HADR you have to determine the SYNC mode you will use and then create a template of ports, service names, and IP addresses you will use
- You normally need to coordinate this with network support personnel or server ADMINS
- Properly size the STANDBY server or servers
  - The primary STANDBY should have same amount of RAM and CPU as the primary in the event of a failover or takeover

# Configuring Databases for HADR

- Before you can use HADR you have to determine the SYNC mode you will use and then create a template of ports, service names, and IP addresses you will use
- You normally need to coordinate this with network support personnel or server ADMINS
- Properly size the STANDBY server or servers
  - The primary STANDBY should have same amount of RAM and CPU as the primary in the event of a failover or takeover

# Configuring Databases for HADR

- Setup separate dedicated NETWORK for HADR PRIMARY to SECONDARY connection
- Open HADR ports in Firewalls else HADR will fail and can be difficult to diagnose

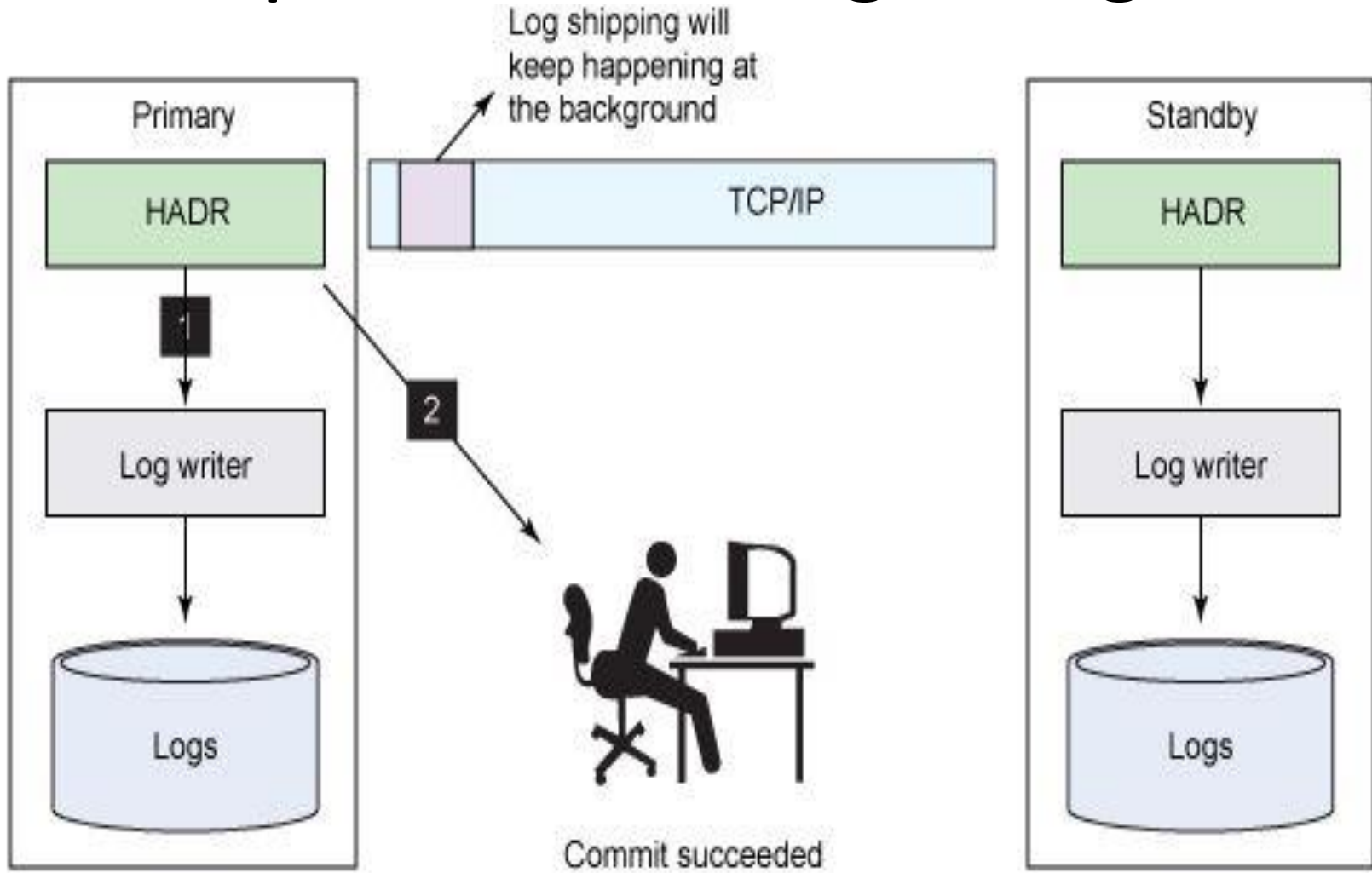


# HADR Design RECAP

- Before you can use HADR you have to determine the SYNC mode you will use and then create a template of ports, service names, and IP addresses you will use
- Create a diagram of the proposed architecture, so all involved have a common picture of the setup
  - Helps to avoid confusion and misconceptions!



# Sample HADR Design Diagram



# DB CFG HADR Parameters

- Before you can use HADR you have to determine the SYNC mode you will use and then create a template of ports, service names, and IP addresses you will use

# DB CFG HADR Parameters -- Primary

HADR local host name (HADR\_LOCAL\_HOST) = 10.221.37.1

HADR local service name (HADR\_LOCAL\_SVC) = db2h\_DB2\_1

HADR remote host name (HADR\_REMOTE\_HOST) = 10.221.37.2

HADR remote service name (HADR\_REMOTE\_SVC) = db2h\_DB2\_2

HADR instance name of remote server (HADR\_REMOTE\_INST) = DB2

HADR timeout value (HADR\_TIMEOUT) = 120

HADR log write synchronization mode (HADR\_SYNCMODE) = SUPERASYNC

HADR peer window duration (seconds) (HADR\_PEER\_WINDOW) = 600

# ETC/SERVICES Entries -- Primary

- db2h\_DB2\_1 58101/tcp #DB2  
HADR GCPROD Port
- db2h\_DB2\_2 58102/tcp #DB2  
HADR GCPROD Port

# DB CFG HADR Parameters -- Standby

HADR local host name (HADR\_LOCAL\_HOST) = 10.221.37.2

HADR local service name (HADR\_LOCAL\_SVC) = db2h\_DB2\_2

HADR remote host name (HADR\_REMOTE\_HOST) = 10.221.37.1

HADR remote service name (HADR\_REMOTE\_SVC) = db2h\_DB2\_1

HADR instance name of remote server (HADR\_REMOTE\_INST) = DB2

HADR timeout value (HADR\_TIMEOUT) = 120

HADR log write synchronization mode (HADR\_SYNCMODE) = SUPERASYNC

HADR peer window duration (seconds) (HADR\_PEER\_WINDOW) = 600

# ETC/SERVICES Entries -- Standby

- db2h\_DB2\_1 58101/tcp #DB2  
HADR Port gcprod
- db2h\_DB2\_2 58102/tcp #DB2  
HADR Port gcprod

# DB2 Registry Settings -- Primary

[e] DB2PATH=C:\Program Files\IBM\SQLLIB  
[i] DB2\_STANDBY\_ISO=UR  
[i] DB2\_HADR\_ROS=ON  
[i] DB2\_CAPTURE\_LOCKTIMEOUT=ON  
[i] DB2\_CREATE\_DB\_ON\_PATH=YES  
[i] DB2\_SKIPINSERTED=yes  
[i] DB2\_USE\_ALTERNATE\_PAGE\_CLEANING=on  
[i] DB2\_EVALUNCOMMITTED=yes  
[i] DB2\_SKIPDELETED=yes  
[i] DB2INSTPROF=C:\ProgramData\IBM\DB2\DB2COPY1  
[i] DB2COMM=TCPIP  
[i] DB2\_PARALLEL\_IO=\*  
[g] DB2\_EXTSECURITY=YES  
[g] DB2\_COMMON\_APP\_DATA\_PATH=C:\ProgramData  
[g] DB2SYSTEM=CW-DB01  
[g] DB2PATH=C:\Program Files\IBM\SQLLIB  
[g] DB2INSTDEF=DB2  
[g] DB2ADMINSERVER=DB2DAS00

# HADR Database Snapshot -- Primary

## HADR Status

Role = Primary

State = Remote catchup

Synchronization mode = SuperAsync

Connection status = Connected , 03/14/2014 09:56:39.738078

Heartbeats missed = 0

Local host = 10.221.37.1

Local service = db2h\_DB2\_1

Remote host = 10.221.37.2

Remote service = db2h\_DB2\_2

Remote instance = DB2

timeout(seconds) = 120

Primary log position(file, page, LSN) = S0338748.LOG, 13150, 00004D16BD4B6CFF

Standby log position(file, page, LSN) = S0338748.LOG, 12650, 00004D16BD2C2084

Log gap running average(bytes) = 15109



# HADR db2pd -gc\_prod -hadr -- Primary

Database Partition 0 -- Database GC\_PROD -- Active -- Up 10 days 09:42:00 -- Date  
2014-03-24-19.37.51.674000

## HADR Information:

Role	State	SyncMode	HeartBeatsMissed	LogGapRunAvg (bytes)
Primary	RemoteCatchup	SuperAsync	0	17746

ConnectStatus	ConnectTime	Timeout
Connected	Fri Mar 14 09:56:39 2014 (1394805399)	120

LocalHost	LocalService
10.221.37.1	db2h_DB2_1

RemoteHost	RemoteService	RemoteInstance
10.221.37.2	db2h_DB2_2	DB2

PrimaryFile	PrimaryPg	PrimaryLSN
S0338748.LOG	5055	0x00004D16BB517AA1

StandByFile	StandByPg	StandByLSN
S0338748.LOG	4014	0x00004D16BB407011

# HADR Database Snapshot -- Standby

## HADR Status

Role = Standby

State = Remote catchup

Synchronization mode = SuperAsync

Connection status = Connected , 03/14/2014 09:56:41.042773

Heartbeats missed = 0

Local host = 10.221.37.2

Local service = db2h\_DB2\_2

Remote host = 10.221.37.1

Remote service = db2h\_DB2\_1

Remote instance = DB2

timeout(seconds) = 120

Primary log position(file, page, LSN) = S0338747.LOG, 5370, 00004D16B6832E27

Standby log position(file, page, LSN) = S0338747.LOG, 5370, 00004D16B6832E27

Log gap running average(bytes) = 529199

# HADR db2pd – gc\_prod –hadr -- Standby

Database Partition 0 -- Database GC\_PROD -- Active Standby -- Up 376 days 07:15:45 -- Date 2014-03-24-19.26.22.802000

## HADR Information:

Role	State	SyncMode	HeartBeatsMissed	LogGapRunAvg (bytes)
Standby	RemoteCatchup	SuperAsync	0	491834

ConnectStatus	ConnectTime	Timeout
Connected	Fri Mar 14 09:56:41 2014 (1394805401)	120

ReplayOnlyWindowStatus	ReplayOnlyWindowStartTime	MaintenanceTxCount
Inactive	N/A	0

LocalHost	LocalService
10.221.37.2	db2h_DB2_2

RemoteHost	RemoteService	RemoteInstance
10.221.37.1	db2h_DB2_1	DB2

PrimaryFile	PrimaryPg	PrimaryLSN
S0338747.LOG	9862	0x00004D16B79BE3C5

StandByFile	StandByPg	StandByLSN	StandByRcvBufUsed
-------------	-----------	------------	-------------------

# DB2 Registry Settings -- Standby

[e] DB2PATH=C:\Program Files\IBM\SQLLIB  
[i] DB2\_STANDBY\_ISO=UR  
[i] DB2\_HADR\_ROS=ON  
[i] DB2\_CREATE\_DB\_ON\_PATH=YES  
[i] DB2\_SKIPINSERTED=YES  
[i] DB2\_USE\_ALTERNATE\_PAGE\_CLEANING=YES  
[i] DB2\_EVALUNCOMMITTED=YES  
[i] DB2\_SKIPDELETED=YES  
[i] DB2INSTPROF=C:\ProgramData\IBM\DB2\DB2COPY1  
[i] DB2COMM=TCPIP  
[i] DB2\_PARALLEL\_IO=\*  
[g] DB2\_EXTSECURITY=YES  
[g] DB2\_COMMON\_APP\_DATA\_PATH=C:\ProgramData  
[g] DB2SYSTEM=CW-DB02  
[g] DB2PATH=C:\Program Files\IBM\SQLLIB  
[g] DB2INSTDEF=DB2  
[g] DB2ADMINSERVER=DB2DAS00

# SNAPHADR Administrative View in DB2

## 9.7

- Sample output – (DB2 9.7)

```
DBNAME  HADR_ROLE  HADR_STATE  
HADR_SYNCMODE HADR_CONNECT_STATUS
```

---

```
GC_PROD PRIMARY  REM_CATCHUP  SUPERASYNC  
CONNECTED
```

1 record(s) selected.

# MON\_GET\_HADR Table Function

- Sample output – (DB2 10.1 and above)

```
HADR_ROLE STANDBY_ID HADR_STATE
PRIMARY_MEMBER_HOST -----
----- PRIMARY 1 PEER
hostP.ibm.com PRIMARY 2 REMOTE_CATCHUP
hostP.ibm.com PRIMARY 3 REMOTE_CATCHUP
hostP.ibm.com STANDBY_MEMBER_HOST -----
--- hostS1.ibm.com hostS2.ibm.com hostS3.ibm.com 3
record(s) selected.
```

# HADR Shutdown and Startup Log Messages

- [..\ipdevco\hadrmessages.txt](#)

# HADR LOG SPOOLING

- **LOG Spooling** introduced in DB2 10.1 to resolve HADR receive buffer full issues with slow STANDBY which blocks primary in all but SUPERASYNC mode
- **hadr\_spool\_limit** DB CFG parameter
  - As of DB2 10.5 set to **AUTOMATIC** by default
  - Value of 0 turns it off
- Ensure disk space for active logs is large enough when using log spooling



# Other HADR Tuneables

- DB2\_HADR\_BUF\_SIZE
- HADR\_TIMEOUT
- HADR\_PEER\_WINDOW
- DB2\_HADR\_PEER\_WINDOW\_WAIT\_LIMIT

# DB2\_HADR\_BUF\_SIZE

- Log receive on standby writes logs into this receive buffer as a producer. Log replay reads the logs as a consumer. If the consumer is too slow, the buffer fills up, and the standby cannot receive any more logs. For SYNC and NEARSYNC modes, chances are that the primary can still send out one log flush when the standby buffer is full. This flush will be buffered along the network pipeline from the primary to the standby, in various OS or network buffers. But the standby database is not able to receive it into its receive buffer. The primary will then be blocked, waiting for an acknowledgment message. For ASYNC mode, the primary will keep sending logs until the network pipeline fills up, then it will get network congestion error when trying to send more logs. For SYNC and NEARSYNC, if the network pipeline can not absorb a flush, the primary will also see congestion.
- Setting the **DB2\_HADR\_BUF\_SIZE** registry variable to the size of expected workload spikes configures your HADR standby log receive buffer to absorb spikes in the primary workload.

# DB2\_HADR\_BUF\_SIZE

- Monitor with db2pd command
  - Reports percentage of buffer used
  - If 0 means nothing to replay, if 100% means buffer full and CONGESTION state likely
- As of DB2 10.1 and above spooling should be used to prevent this from occurring versus tuning buffer size

# HADR\_TIMEOUT

- Default is 120 seconds
- If a database does not receive a heartbeat message from its partner database within the number of seconds set by the value of the **hadr\_timeout** value, the database considers the connection down and closes the TCP connection.
- Recommendation – Use the default and adjust as necessary

# HADR\_PEER\_WINDOW

- The **hadr\_peer\_window** database configuration parameter enables failover operations with no data loss if the primary failed within a specified time—the peer window. When the HADR primary database is in peer state, logs are sent directly from the in-memory log buffer when the logs are written to the local disk. If the connection is broken, the primary database blocks logging for **hadr\_peer\_window** seconds and no transaction is committed. If the primary database then fails within this window, we know that there are no committed-but-not-replicated transactions and therefore there is no data loss during the failover to the standby database.
- Recommendations – **USE WITH CAUTION** since PRIMARY will be blocked for **hadr\_peer\_window** seconds if connection lost

# DB2\_HADR\_PEER\_WINDOW\_WAIT\_LIMIT

- Setting the DB2\_HADR\_PEER\_WAIT\_LIMIT registry variable allows you to prevent primary database logging from blocking because of a slow or blocked standby database.
- Can be used to cause primary to resume logging if in peer state and logging blocked
- However, if used with HADR\_PEER\_WINDOW then database logging will be blocked for DB2\_HADR\_PEER\_WINDOW\_WAIT\_LIMIT + HADR\_PEER\_WINDOW duration

# HADR Setup Steps

- Update primary DB CFG parameters
- Update /ETC/SERVICES ports and service names on primary
- Make DB2 HADR registry settings
- Deactivate and Activate the database or stop and start the instance for settings to take effect
- Catalog database and standby node

# HADR Setup Steps

- Backup the primary database online include logs
- Ship backup to standby server
- Make DB2 registry settings
- Update /ETC/SERVICES ports and service names on standby
- Catalog the database and primary node
- Restore the database to the standby instance without rolling forward
- Update the HADR DB CFG parameters
- Stop/Start the standby instance
- Activate the standby database
- Start hadr on the standby
- Activate the primary database
- Start hadr on the primary



# HADR Setup Steps

- Alternatively, use the DB2 Control Center (9.7 and below) or IBM Data Studio to setup the primary and standby to include starting HADR
- Sit back, relax, and monitor!

# HADR

- A look at DB2 HADR in practice

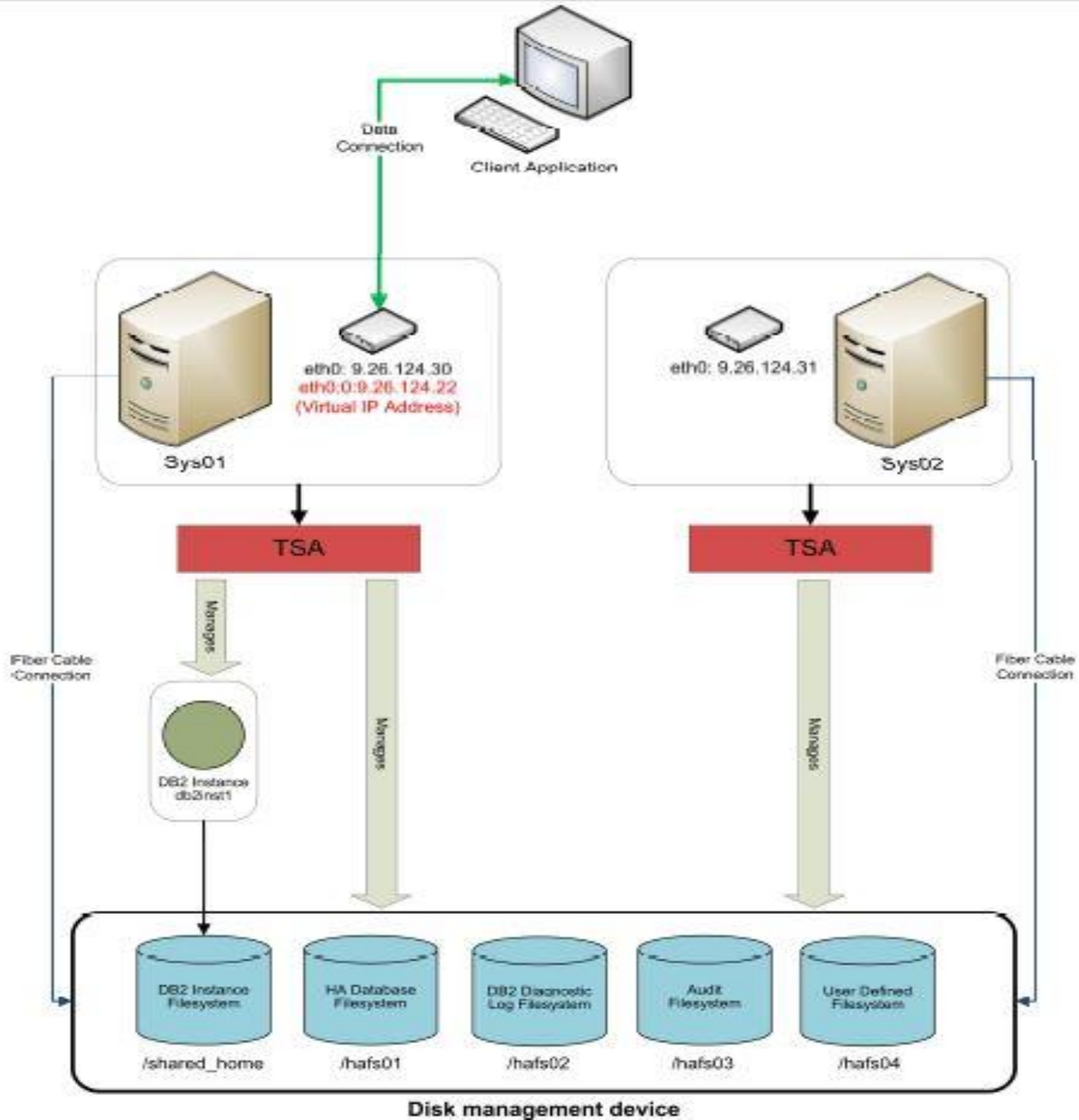
# Clustering



- Everyone wants it but few want to take the time to understand it and support it properly
- Achieve High Availability and Reliability if properly implemented
- Disaster waiting to happen if not properly implemented and understood

# TSAMP

- Tivoli System Automation for Multiplatforms (TSAMP)
- Bundled with DB2 since DB2 9.5
- Comes with all editions of DB2 except DB2 Express-C
- Seeing widespread implementation in 9.7 and above
  - Uses IBM Reliable Scalable Cluster Technology (RSCT) under the covers



# TSAMP Prerequisites

- TSAMP Prerequisites should be met by successfully running db2prereqchk
- Run preprnode once on each node in the cluster
- TSAMP installed (automatically on UNIX when DB2 installed, manual on Windows)
- Shared storage
- Shared filesystems / mount points
- Primary node and secondary node
- DB2 installed at same level on both primary and secondary node
- Same DB2 instance owner id, group and group id on both system
- Same major number for volume groups
- /ETC/SERVICES updated to contain DB2 service names and ports
- These are the main requirements, follow the whitepaper or DB2 HA Redbook, SG 247363

# TSAMP

- DB2 TSAMP clustering program bundled with as **db2haicu**
- **Use in conjunction with the DB2 TSAMP setup whitepaper**
- **Diagram and template what you will need for the cluster in advance**
- **You need inputs to db2haicu in advance**
- **Unlike HADR, TSAMP is at the instance level**

# TSAMP

- db2haicu is a menu-driven script that you use to setup a cluster
- Defines the cluster domain and defines cluster resources
- db2haicu can also use an XML file as input
- Typical cluster setup is active – passive with one active node and a passive node ready for failover



# TSAMP

- db2haicu inputs:
- Shared storage paths and filesystems
- Mount points not set to auto mount
- IP address of primary and secondary node (server)
- Virtual IP address to use for the cluster
- Quorum IP address
- DB2 TSAMP clustering program bundled with as **db2haicu**
- **Use in conjunction with the DB2 TSAMP setup whitepaper**
- **Diagram and template what you will need for the cluster in advance**
- **You need inputs to db2haicu in advance**

# db2haicu

172.17.2.13 - PuTTY

db2haicu determined the current DB2 database manager instance is 'db2inst8'. The cluster configuration that follows will apply to this instance.

db2haicu is collecting information on your current setup. This step may take some time as db2haicu will need to activate all databases for the instance to discover all paths ...

When you use db2haicu to configure your clustered environment, you create cluster domains. For more information, see the topic 'Creating a cluster domain with db2haicu' in the DB2 Information Center. db2haicu is searching the current machine for an existing active cluster domain ...

db2haicu found a cluster domain called 'KABLE\_domain' on this machine. The cluster configuration that follows will apply to this domain.

Retrieving high availability configuration parameter for instance 'db2inst8' ...

The cluster manager name configuration parameter (high availability configuration parameter) is not set. For more information, see the topic "cluster\_mgr - Cluster manager name configuration parameter" in the DB2 Information Center. Do you want to set the high availability configuration parameter?

The following are valid settings for the high availability configuration parameter:

- 1.TSA
- 2.Vendor

Enter a value for the high availability configuration parameter: [1]

# TSAMP Quorum Definition

- A quorum definition is required to enable TSAMP to decide which node to use as a tiebreaker during a node failure. TSAMP supports a 'network quorum'.
- A network quorum (or network tiebreaker) is a pingable IP address that is used to decide which node in the cluster will serve as the active node during a site failure, and which nodes will be offline. Note that the machine hosting this IP address does not need any particular software or operating system level installed; its primary requirement is that it can be pinged from all nodes in the cluster, and must remain pingable in the case of cluster node failures
- db2haicu will prompt you for the IP address

# db2haicu

- Upon successful completion of defining all the resources to db2haicu, the cluster will be started and active
- Use `db2pd -ha` or `lssam` command to monitor status of HA for the instance

# Issam output

```
$ Issam
Online IBM.ResourceGroup:db2_db2inst8_0-rg Control=MemberInProblemState
Nominal=                               Online
  |- Online IBM.Application:db2_db2inst8_0-rs
    |- Online IBM.Application:db2_db2inst8_0-rs:dist-db2-t1
    '- Offline IBM.Application:db2_db2inst8_0-rs:dist-db2-t2
  |- Online IBM.Application:db2mnt-db2home_db2inst8_db2-rs
Control=MemberI                          nProblemState
  |- Online IBM.Application:db2mnt-db2home_db2inst8_db2-rs:dist-
db                                         2-t1
  '- Failed offline IBM.Application:db2mnt-db2home_db2inst8_db2-
rs                                         :dist-db2-t2
  |- Online IBM.Application:db2mnt-db2home_db2inst8_db2data-rs
Control=Mem                               berInProblemState
  |- Online IBM.Application:db2mnt-db2home_db2inst8_db2data-
rs:dis                                     t-db2-t1
  '- Failed offline IBM.Application:db2mnt-db2home_db2inst8_db2dat          a-
rs:dist-db2-t2
  '- Online IBM.ServiceIP:db2ip_172_17_3_160-rs
    |- Online IBM.ServiceIP:db2ip_172_17_3_160-rs:dist-db2-t1
    '- Offline IBM.ServiceIP:db2ip_172_17_3_160-rs:dist-db2-t2
Online IBM.Equivalency:db2_db2inst8_0-rg_group-equ
  |- Online IBM.PeerNode:dist-db2-t1:dist-db2-t1
  '- Online IBM.PeerNode:dist-db2-t2:dist-db2-t2
Online IBM.Equivalency:db2_public_network_0
  |- Online IBM.NetworkInterface:en0:dist-db2-t1
```

# db2pd -ha output

[db2pdhaout.txt](#)

```
$ db2pd -ha
```

```
      DB2 HA Status
```

```
Instance Information:
```

```
Instance Name           = db2inst8
```

```
Number Of Domains       = 1
```

```
Number Of RGs for instance = 1
```

```
Domain Information:
```

```
Domain Name             = KABLE_domain
```

```
Cluster Version         = 3.1.4.4
```

```
Cluster State           = Online
```

```
Number of nodes         = 2
```

```
Node Information:
```

```
Node Name               State
```

```
-----
```

```
dist-db2-t2            Online
```

```
dist-db2-t1            Online
```

# TSAMP

- Conduct failure testing
  - Network
  - Storage
  - Server
- Learn TSAMP and RSCT commands to use to monitor status of cluster, stop cluster, and move cluster to secondary node for maintenance or other reasons
- Learn how to know if the cluster has failed over and what to do to get it back to the primary
- Test all of the above and document

# Summary

- DB2 HADR what it is, how it works and how to implement and monitor it
- DB2 HADR Best practices
- New features in DB2 10.5
- Described how to define, setup and integrate TSAMP clustering
- Provided DB2 HADR and TSAMP references and best



# HADR and TSAMP References

- Whitepaper - DB2 HADR Multiple Standbys  
[http://public.dhe.ibm.com/software/dw/data/dm-1206hadrmultiplestandby/HADR Multiple Standbys\\_jn20.pdf](http://public.dhe.ibm.com/software/dw/data/dm-1206hadrmultiplestandby/HADR_Multiple_Standbys_jn20.pdf)
- DBA HA Redbook:  
<http://www.redbooks.ibm.com/redbooks/pdfs/sg247363.pdf>

# HADR and TSAMP References

- DB2 HADR Best Practices

[https://www.ibm.com/developerworks/community/wikis/home?lang=en\\_US#!/wiki/Wc9a068d7f6a64434aece0d297ea80ab1/page/High%20Availability%20Disaster%20](https://www.ibm.com/developerworks/community/wikis/home?lang=en_US#!/wiki/Wc9a068d7f6a64434aece0d297ea80ab1/page/High%20Availability%20Disaster%20)

- Setup HADR with Data Studio

<http://www.ibm.com/developerworks/data/tutorials/dm-1003optimhadr/index.html?ca=dat>

# HADR and TSAMP References

- DB2 HADR Simulator – whitepaper  
<http://www.ibm.com/developerworks/data/library/techarticle/dm-1310db2luwhadr/dm-1310db2luwhadr-pdf.pdf>
- DB2 and TSAMP Setup Whitepaper  
<https://www.ibm.com/developerworks/data/library/long/dm-0909hasharedstorage/>
- (Note there is a 10.1 version of the above)

A large, bright yellow sunburst graphic with multiple sharp points radiating from a central point, positioned behind the "THANK YOU" text.

**THANK YOU**

# Achieving High Availability with DB2 HADR and TSAMP

Philip K. Gunning  
Gunning Technology Solutions, LLC  
[pgunning@gts1consulting.com](mailto:pgunning@gts1consulting.com)  
[www.gts1consulting.com](http://www.gts1consulting.com)

# DB2 Books by Phil

