looki looki, no hands

Sven Heidorn

Centrala Studiestödsnämnden

sven.heidorn@csn.se

Certified for IBM DB2 e-business software

Swedish DB2 User Group 07/01/24

Background

- Services where to be launched to a broader range of customers.
  - So
- Existing system needed to be scaled up and moved to a more stable platform.
  - And
- Success of the project was critical.

Objectives

- To build a stable and reliable ”hands off” database system for 24 * 7 operation using standard features
- Replace existing Windows based system with as little disruption as possible

Old solutions problem

- Windows constant patching
  - Single most cause for downtime.
  - Even though, it never gets any better.
- Cluster
  - Lots of hands on required after failover.
  - Performance at failover.
  - DBAs in the hands of the OS admins
- Replication services
  - Some work as services with cluster.
  - Others did not!

Design goals

- Automate
- Automate
- Automate
- Aim for no cluster
  - Get out of the hands of OS admins
- Simple monitoring
  - Get out of the hands of under trained operators
- Simple or no administration
  - Get the job out of our hands

Components Used

- DB2 Workgroup Unlimited Edition V8.2
  - With HADR feature
  - DataPropagator – SQL based
- Tivoli Storage Manager
- AIX
  - HACMP
  - Shell scripts

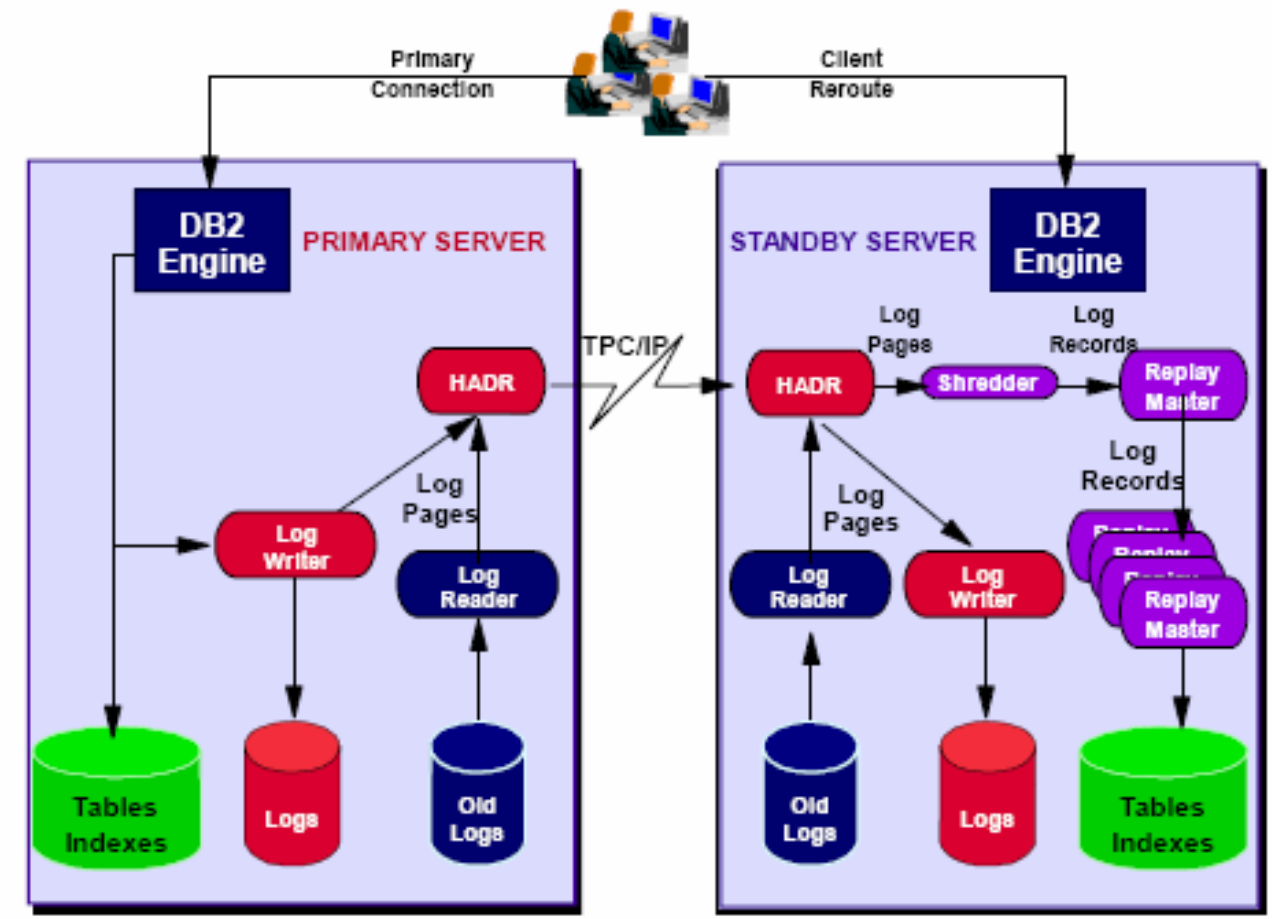
The cluster issue

- Once you have a cluster you are
  - Likely to be dependent of UNIX system admins
  - Everything gets more complicated
- Don’t be fooled by the sample scripts
  - It’s still not easy
  - They don’t contain all you might need
  - Documentation is poor if any
- Test, test and retest
  - make sure you know what happens in every situation

Instead of cluster?

- HADR
  - High Availability Disaster Recovery
  - Included in ESE
  - Otherwise charged feature
  - Does not need cluster for failover
- But….
  - Still needs cluster for automation of failover
- However
  - Will not need cluster in the near future

HADR


Primary
Connection
Client
Reroute
DB2
Engine
PRIMARY SERVER
HADR
TPC/IP
Log
Writer
Log
Pages
Log
Reader
Tables
Indexes
Logs
Old
Logs
STANDBY SERVER
DB2
Engine
Log
Pages
Log
Records
HADR
Shredder
Replay
Master
Log
Records
Log
Pages
Log
Reader
Log
Writer
Replay
Master
Old
Logs
Logs
Tables
Indexes

HADR

- Servers need to be identical down to the last bit.
  - for the part used by the DB2 system
- So need same..
  - File system structure
  - Naming
- For clarity..
  - Keep everything identical if possible
- If running DAS keep it down on standby system
  - Applies to systems with HADR managed tools catalog
  - Only start DAS when system is primary

Think redundancy

- Make sure there is a alternative for everything
  - Dual logging
  - Log archiving
  - Diag archiving
  - HADR of cource
  - TSM
    - This can be a showstopper…..

Instance

- Keep instance clean
- Separate filesystems for
  - Instance
  - Dumps & Diags
  - Diag archives
  - Logs
  - Mirror logs
  - Log archives
  - Data, Indexes & temp

Logs

- Keep primary on separate disk
  - In its own filesystem
    - Placed on the outer edge of the disk
  - Not in instance where it ends up by default
- Let DB2 archive logs directly to TSM
  - Make sure you have access to TSM at all time
  - Remember, this can be a showstopper
- Separate log_archive directory in case TSM archiving fails
  - db cfg FAILARCHPATH
  - This will give some relieve if TSM fails
  - Make sure you get an alarm if filesystem starts to fill up
- Mirror logs (dual logging) on separate disk
  - In its own filesystem
    - Placed on the outer edge of the disk
  - If no mirror and primary fails, you are dead!
  - DB2 deletes these logs once they are no longer needed

Dump & Diag

- Place diagnostic data in separate filesystem
  - Not in instance where it ends up by default
  - update dbm cfg using DIAGPATH = <path>
- Separate diag_archive directory
- Use db2diag to archive the db2diag.log
  - db2diag –archive <your archive path> <your diag path> /db2diag.log
- Use cron to execute regulary
  - 0 3 * * 0 <your script path>/ ArchDiag.ksh
  - task center will do the trick to but not as reliable
- Use TSM to copy off and delete archived diag logs
  - Save # of copies or # of days
- Use script to delete trap files on regular basis

Dump & Diag: sample

This script will work with several instances archiving to the same archive path.

```
#!/bin/ksh
#set -x
#############################################################################
#######  ArchDiag                                                     #######
#######  Archive db2diag.log                                          #######
#######                                                               #######
#######  © 2007 - Sven Heidorn                                        #######
#############################################################################
ARCHPARH=<your archive path>
# Get path to dump directory
DIAGPATH=$(db2 get dbm cfg | grep "DIAGPATH" | awk '{print $7}')
# get inst owner
INSTOWN=`whoami`
# Archive diag log
db2diag -archive $ARCHPATH/$INSTOWN $DIAGPATH/db2diag.log
```

The following is an sample script to execute the script above from cron as root for multiple instances. The cron sample entry provided in the comment below would run the script every Sunday at 3 AM. You need to modify the script below to conform to your environment.

```
#!/bin/ksh
#set -x
#############################################################################
#    RunArchDiag
#    Script to run db2diag.log archiving
#    0 3 * * 0 <your script path>/RunArchDiag
#############################################################################
su - <inst1_owner> -c <your script path>/ArchDiag
su - <inst2_owner> -c <your script path>/ArchDiag
```

create database

- Define the tablespaces
  - Syscatspace
    - DMS
    - Place it, not in instance!
  - Userspace1
    - DMS
    - Place it, not in instance!
    - Consider dropping
  - Tempspace1
    - SMS
    - Place it, not in instance!
    - Consider renaming, ex. tempspace4k

Tablespaces

- Use DMS
  - For everything
    - raw or file
  - Except tempspace
    - use SMS
- Use Auto resize
  - With maxsize!
  - For syscatspace maxsize none may be appropriate
- Disable file system cashing
  - Memory is better used in bufferpool
- Automatic storage
  - Less control
  - DB2 decides where to place containers

Automatic Maintenance

- Use for online activities
  - Runstats
  - Backup
- Report offline activities
  - Reorg
  - Use if you have an offline window
- Define a window
  - Anytime is probably not a good idea

Data Propagator

- Tricky with failover
  - Use dynamic scripts for start and stop
    - Script should figure out what to start or stop
    - Capture
    - Apply
    - Monitor
- Call scripts from HACMP scripts
  - Use same approach as for HADR
- This was a major problem with MSCS

Tools catalog

- Separate database
  - Full automation needs HADR
  - Still debating if this is right approach
  - Will it work?
- Contains “scheduling system”
  - This is not the same after failover
  - Needs to be updated by HACMP scripts
- Cron might be a better choice

In case of failure

- Do not autostart DB2
  - If or when the failing node reboots you are likely to get 2 primary's
    - Split brain – not good!
    - db2iauto -off InstName - recommended with HACMP
  - Run STOP HADR command before starting DB2 on the failing node
    - Then start HADR as appropriate
  - DB2 can be stopped with HADR “active”
    - Used during rolling upgrades

Keep yourself informed

- Scan db2diag.log for errors
  - Use db2diag command
    - Db2diag –readfile –level “error, severe” –history 10m
  - send output if any by E-mail
- Keep track of tablespace filesystem utilization
  - Use scripts with snapshot functions
  - Warn before DB2 reaches the maxsize
  - Make sure ALTERs don’t breach filesystem size
    - Keep filesystem size larger than sum(TS max size)

Keep yourself informed: sample

The following script scans the last 10 minutes of the diag log for errors and pipes the output to a temporary file. The file existence test is due to that there might not exist a diag log if it has been archived recently.

```
#!/bin/ksh
#set -x
#############################################################################
#######  DiagHist                                                     #######
#######  Log last 10 minutes error messages to file.                  #######
#######                                                               #######
#######  © 2007 - Sven Heidorn                                        #######
#############################################################################
# Get path to dump directory
DIAGPATH=$(db2 get dbm cfg | grep "DIAGPATH" | awk '{print $7}')
if [[ -s $DIAGPATH/db2diag.log ]] then
  db2diag -readfile -level "error, severe" -history 10m > /tmp/diaghist.tmp
fi
```

The following is a sample to run the above script and mail the output if any. You need to modify the script below to conform to your environment.

```
#!/bin/ksh
#set -x
#############################################################################
#######  SendDiag                                                     #######
#######  Mail last 10 minutes error messages to ....                  #######
#############################################################################
HOSTNAME=$(echo `hostname` | awk '{print toupper($1)}')
su - <inst_owner> -c <your script path>/DiagHist.ksh
if [[ -s /tmp/diaghist.tmp ]] then
  mail -s Errors-DB2-$HOSTNAME-<instance> mr.mr@mail.com < /tmp/diaghist.tmp
  rm /tmp/diaghist.tmp
fi
```

Future enhancements

- Tivoli System Automation
  - Cluster manager
  - Switch from HACMP to TSA
  - Should be included with HADR as of FP12
    - AIX, earlier for Linux
- Floating IP addresses
  - No risk of split brain scenario in HADR
  - Very good for WebSphere data sources
  - Managed by TSA
- Use standby node resources when standby
  - Use unused capacity for acceptance testing in separate instance
  - Let TSA force acceptance testing instance off during failover

Questions