HADR Overview

- Replication is done by log shipping
- Whole database is replicated (easy administration)

No special hardware or software needed, just standard TCP
Resources

- **HADR Wiki on IBM developerWorks**
  - Welcome page
    - Central guide to HADR materials
      - Covers all HADR aspects
      - Collection of wiki articles
      - Links to other resources, such as white papers
  - Performance Tuning page
    - Base of this presentation
Agenda

- Configuration
- Monitoring
- Diagnostics
TCP Tuning

- Recommended TCP window size:
  - sendRate * roundTripTime
    - Consider: bestSendRate * worstRoundTripTime
  - System default may not be optimal.
  - Smaller buffer may not make full use of network bandwidth.

- Set by DB2 registry variable
  - DB2_HADR_SOSNDBUF and DB2_HADR_SORCVBUF
  - Recommendation: Find optimal value using HADR simulator. The deploy to database.
TCP Tuning (continued)

- What if you don’t know the nominal bandwidth?
  - Start from 64KB, double buffer size until throughput no longer increases
    - Recommended minimal is 64KB
      - 16 log pages. Remote catchup state log read batch size.

- Enable TCP Window Scaling (RFC1323)
- Lift OS socket buffer limit if needed.
- Confirm your config
  - DB2 treat as soft error when request is not satisfied.
  - SOCK_SEND_BUF_REQUESTED
  - SOCK_RECV_BUF_REQUESTED
  - SOCK_SEND_BUF_ACTUAL
  - SOCK_RECV_BUF_ACTUAL
HADR State Transition

- Local catchup
  - After startup, standby first tries to read logs from local source.
  - Reads from log path, overflow path, and archive

- Remote catchup pending
  - After local catchup reaches local end of log, if there is no connection to primary, standby waits in this state.

- Remote catchup
  - Primary sends log pages to standby, reading from disk or archive

- Peer
  - Standby sends log pages to standby directly from log write buffer.
  - Primary writes a local copy concurrently
  - HADR sync mode applies only to peer state (except superAsync)
Synchronization Modes

- SYNC
- NEARSYNC
- ASYNC
- SUPERASYNC
SYNC mode

- Transactions on primary will commit only after logs have been written to disk on both primary and standby.
- Maximal data protection, with performance cost.
- After writing logs to local disk, primary sends a copy to standby. Primary will then wait for “log written” ack message from standby.
- Serial write and send on primary
- **In peer state,** any transaction committed on primary is guaranteed to have committed on standby too.
- **In peer state,** if a failover occurs, you will not lose any committed transaction.
NEARSYNC mode

- Transactions on primary will commit only after logs have been written to disk on primary and received into memory on standby.
- Protection nearly as good as SYNC mode. Performance is better than SYNC mode.
- When writing logs to local disk, primary also sends a copy to standby. Primary will then wait for “log received” ack message from standby.
- Parallel write and send on primary
- **In peer state**, you will lose data in a failover only if standby fails before it writes received log pages locally (very small window)
ASYNC mode

- Transactions on primary will commit only after logs have been written to local disk and sent to standby.
- Better performance, less data protection.
- When writing logs to local disk, primary also sends a copy to standby. Primary will go on as soon as send() call to TCP returns.
- **In peer state,** any transaction committed on primary is guaranteed to have been “sent” to standby.
- **In peer state,** if a failover occurs, logs sent but not yet received can be lost.
SUPERASYNC mode

- HADR pair never enters peer state.
  - State transition stops at remote catchup
- Transaction commit on primary has no dependency on log shipping.
- Slow network or standby will not slow down primary
- But standby can fall behind
  - Monitor log gap closely.
  - Failover can lose data in log gap.
- Role switch allowed in remote catchup state
  - Only allowed in peer state in other sync modes.
  - Check log gap before issuing takeover command
  - Role switch will stop transactions on primary, ship all logs and finish replay. No data loss.
  - Large gap will result in long takeover time
HADR Planning: Choosing a sync mode

- Step 1: Know Your Workload
  - Use DB2 log scanner to measure logging rate
- Step 2: Know Your Disks
  - Use HADR simulator to measure disk speed
- Step 3: Know Your Network
  - Use HADR simulator to measure network speed
- Step 4: Know Your Sync Modes
  - Use HADR calculator to estimate impact to primary workload under various HADR sync modes.

Details at
Peer wait limit

- HADR_PEER_WAIT_LIMIT (registry variable)
  - Wait limit for peer state log replication.
  - Default 0, meaning no limit.
  - Handles slow network
    - Cannot send out data, or ack message delayed.
  - Handles slow standby
    - Slow replay on standby causes “receive buffer/spool full”. Standby cannot receive more logs
    - Async mode: Primary sees “congestion”
    - Sync and nearsync mode: Primary may or may not see congestion. May send out a flush, then wait for ack.
Standby receive buffer and spool size

- Two ways to stage received log data
  - In memory buffer and on disk spool
- Absorbs primary load spike
  - Won’t help on sustained high workload.
  - Spooling recommended over buffering
    - Spooling supported from V10.1
    - Defaults to “automatic” on V10.5 and later
      - Automatic size: capacity of logprimary + logsecond log files
- Side effect
  - Takeover (forced and nonforced) must finish replaying all staged logs.
- Monitoring
  - STANDBY_RECV_BUF_PERCENT
  - STANDBY_SPOOL_PERCENT (V10.5 and later)
Configuration Recap

- TCP tuning
- HADR synchronization mode
- Peer wait limit
- Standby receive buffer and spool
Monitoring HADR, Interfaces

- Monitoring interfaces
  - `db2pd -hadr`
  - Table function MON_GET_HADR
  - Deprecated: database snapshot (CLP and API)

- `Db2pd` can only run on database host machine
  - Light weight. Text output
  - Recommended during takeover

- Table function accessible from any SQL interface
  - Remote access from client
  - Works on standby only when reads on standby is enabled.
Monitoring HADR, Remote database

- Primary and standby exchange info via heartbeat
  - Report info about the remote database
    - Info is delayed up to heartbeat interval
    - HEARTBEAT_INTERVAL is reported in monitoring

- Multiple standbys visible only on the primary
  - Each standby only reports on itself and the primary
  - The primary reports on itself and all standbys
Monitoring HADR, Role and state

- **HADR_ROLE**
  - Primary/standby/standard
  - Also reported as "HADR database role" in db config (available when database is online or offline)

- **HADR_STATE: PEER is good**
  - superAsync mode never enters peer, monitor log gap instead.

- **HADR_CONNECT_STATUS:**
  - CONNECTED / DISCONNECTED / CONGESTED
  - **CONGESTED**: Cannot deliver data to TCP for send

- **HADR_CONNECT_STATUS_TIME**
  - start time of the current HADR.ConnectSTATUS.

- Standby tablespace status: db2pd -tablespaces
  - Replay error can bring a tablespace offline. Subsequent replay skips this tablespace.
  - Avoid surprise at takeover time.
Monitoring HADR, Log position

- Primary log position: PRIMARY_LOG_POS
- Standby receive position: STANDBY_LOG_POS
- Standby replay position: STANDBY_REPLAY_LOG_POS
- “POS” is byte offset. Logging rate = delta(pos) / delta(time)

- HADR_LOG_GAP: running average of (PRIMARY_LOG_POS - STANDBY_LOG_POS)
- STANDBY_RECV_REPLAY_GAP: running average of (STANDBY_LOG_POS - STANDBY_REPLAY_POS)
Monitoring HADR, Logging rate

- When HADR is enabled: compute from PRIMARY_LOG_POS
- When HADR is not enabled
  - V10.1 and later: CURRENT_LSO from table function MON_GET_TRANSACTION_LOG
  - Earlier releases:
    - LOG_WRITES (number of log pages written) field from table function SNAP_GET_DB
    - Or "Log pages written" field from "db2 get snapshot for database" command
- Detailed analysis: DB2 log scanner
  
Monitoring HADR, Log write size and time

- **Log write**
  - table function `MON_GET_TRANSACTION_LOG`:
    - Number of pages written: `LOG_WRITES`
    - Number of write calls to OS: `NUM_LOG_WRITE_IO`
    - Time spent writing (milliseconds): `LOG_WRITE_TIME`
      - Net disk IO time (excluding HADR overhead)
  - “db2 get snapshot for database” command:
    - Number of pages: "Log pages written"
    - Number of write calls to OS: "Number write log IOs"
    - Time spent writing: "Log write time (sec.ns)"
  - Log write metrics are applicable to primary and standby
    - **P and S independently track their log write metrics.**
Monitoring HADR, HADR impact

- HADR impact to primary database logging
  - Is logger waiting on HADR now?
    - LOG_HADR_WAIT_CUR
  - Accumulated HADR wait time
    - LOG_HADR_WAIT_ACCUMULATED
    - LOG_HADR_WAIT_COUNT
  - HADR overhead per log write
    - Delta(LOG_HADR_WAIT_ACCUMULATED) / Delta(LOG_HADR_WAIT_COUNT)
      - Compare to disk write time
      - Example: disk time 10ms per write, with 2ms HADR wait.
    - LOG_HADR_WAIT_RECENT_AVG (only reported by db2pd)
Monitoring HADR, Buffer and spool

- **STANDBY_RECV_BUF_PERCENT**
  - How much of DB2_HADR_BUF_SIZE is being used
  - 100% is bad unless spooling is enabled

- **STANDBY_SPOOL_PERCENT (V10.5 and later)**
  - How much of hadr_spool_limit is being used
  - 100% is bad.

- **HADR_FLAGS**
  - **STANDBY_RECV_BLOCKED (V10.5 and later)**
    - Caused by recv buf full (when spooling not enabled)
    - Or spooling limit reached
    - Or standby log device full
  - **STANDBY_LOGDEVICE_FULL (V10.5 and later)**
Diagnostic: Identifying Bottleneck

- First determine if it is an HADR problem
  - LOG_HADR_WAIT_CUR, LOG_HADR_WAIT_ACCUMULATED, LOG_HADR_WAIT_COUNT

- if (STANDBY_RECV_BLOCKED)
  
  this is a slow standby case.

  if (STANDBY_LOGDEVICEFULL)
    standby log device too small. Enlarge it.
  else
    standby replay is too slow. Tune replay or upgrade hardware.

- else

  Most likely a slow network case.
  Measure network speed to confirm.
  Tune or upgrade network if confirmed.
  Or use a less demanding HADR sync mode.

In rare cases, cause is slow standby log write
  Measure standby disk speed and log write size to confirm.
  Tune or upgrade disk if confirmed.
Multiple Standbys (starting V10.1)

- Treat as multiple primary-standby pairs
- Watch out for
  - Network bottleneck on primary
  - Logging device/archive bottleneck on primary
    - Remote catchup reads from log device/archive
HADR on pureScale (starting V10.5)

- Treat as multiple primary-standby pairs
- Watch out for
  - Network bottleneck on standby replay member
  - Standby replay speed
  - Standby member to SAN (GPFS) interface
The End

- Q and A