IBM Information On Demand 2008

June 3 - 6, 2008 The Hague, Netherlands

EUROPE, MIDDLE EAST & AFRICA IOD CONFERENCE

TSB-1362A

High-End OLTP Systems on DB2 HADR - Real Life Experiences

Stefan Krämer & Berni Schiefer

Housekeeping

- Please turn off your mobiles, blackberries and laptops
- Please remember this is a 'non-smoking' venue!
- Please remember to wear your badge at all times
- Your feedback is valued: please remember to complete your session evaluation forms

Agenda

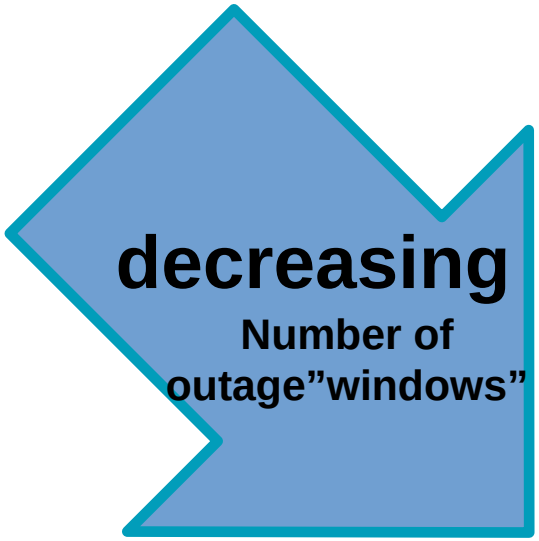
- Goals of this session
- HADR overview
- Current Environment and Throughput
- Congestion, Monitoring Congestion & Troubleshooting
- Other Important Configuration Considerations
- Summary

Goals of this session

- Share Real-Life data about built-in High Availability and Disaster Recovery Technology in DB2 in an extreme OLTP workload environment
- Understand how DB2 HADR technology can be used to provide High Availability in a production environment
- Learn best practices gained from deploying one of the world‘s leading HADR implementations
- Understand how to achieve immediate business benefits for your own DB2 implementation

The quest for business optimization – requirements

- Today's business requirements for ERP customers contain
  - Improved availability – even continuous availability if affordable
  - Rolling infrastructure upgrades with minimal outage requirements
  - Scalability to ensure unlimited data growth

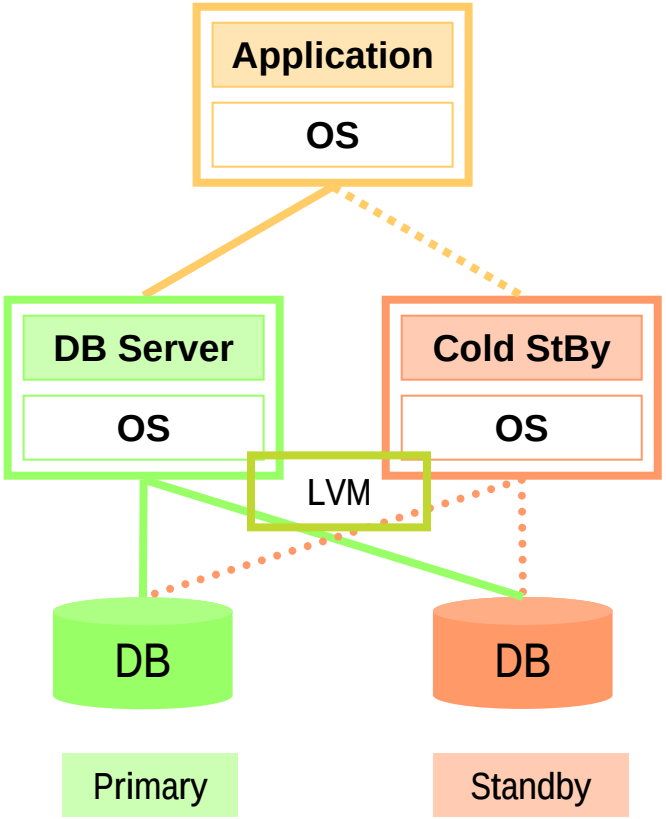


SLA

predefined w/ penalties

Traditional DB2 HA setup before HADR

- HACMP or TSA as software foundation for cluster management
- Industry Standard AIX Logical Volume Mirroring based implementation
  - LVM mirror with 50-90 km distance
  - LVM sometimes corrupting both storage copies
  - LVM mirror write consistency must be turned on – causing higher latency
  - Increasing failover execution time due to disk takeover procedures

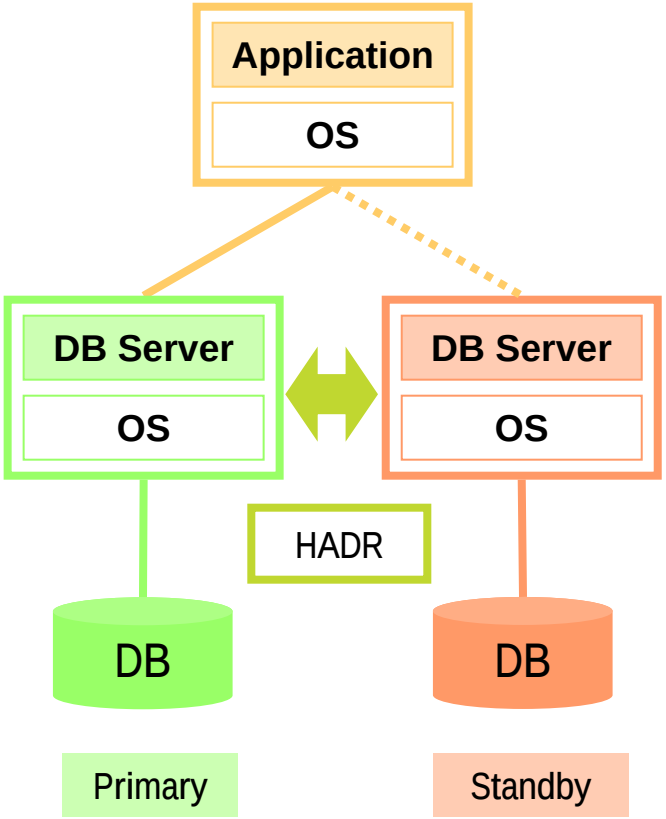


Design Goals of HADR

- Ultra-fast failover
- Easy administration
- Handling of site failures
- Negligible impact on performance
- Configurable degree of consistency
- Protection against storage corruption
- Software upgrades without interruption
- Very easy integration with HA-software
- Transparent failover and failback for applications (combined with “client re-route”)

HADR Principles

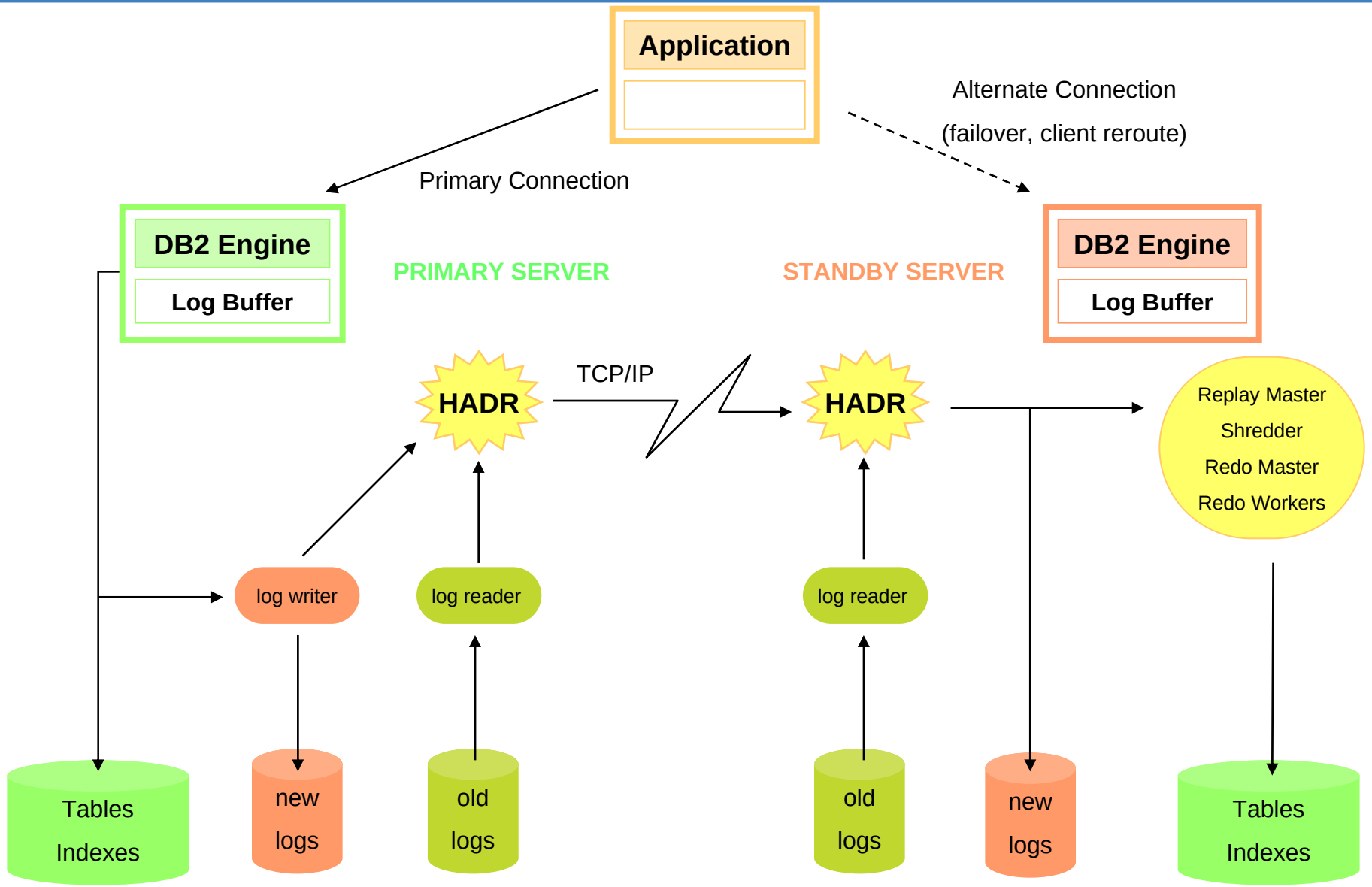
- Two active machines
  - Primary
    - Processes transactions
    - Ships log records (not log files) to the other machine
  - Standby
    - Cloned from the primary
    - Receives and stores log records from the primary
    - Re-applies the log records
- If the primary fails, the standby can take over the transactional workload
  - Standby becomes the new primary
- If the failed machine becomes available again, it can be automatically resynchronized
  - The old primary becomes the new standby
- Operation modes:
  - Asynchronous
  - Near-synchronous
  - Synchronous



HADR Requirements / Restrictions

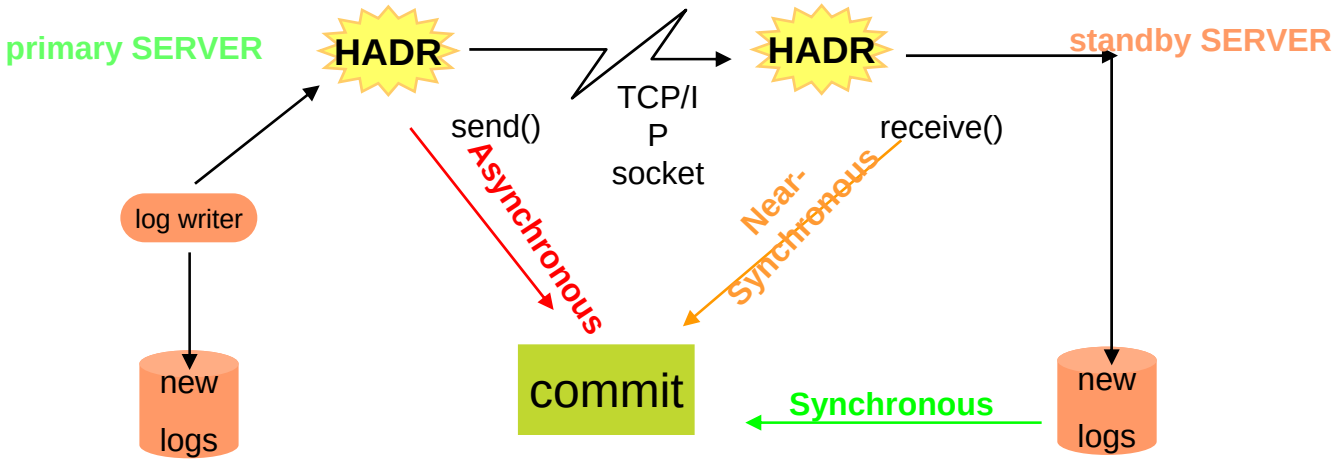
- Requirements
  - Primary and standby databases must have the same name, the same operating system and DB2 levels and the same bit-ness (32-bit vs 64-bit)
- Restrictions
  - Not supported with Data Partitioning Feature (DPF), raw I/O for transaction logs, or redirected restore
  - Rolling upgrades refer to Fix Pack upgrades only
  - Non-logged operations are not replicated:
    - Updates to DB/DBM cfg and registry
    - LOBs > 1GB or LOBs marked as NOT LOGGED
    - NOT LOGGED INITIALLY operations
  - Log archiving and self-tuning memory manager (STMM) available on the primary database
  - No CONNECT access to standby database

HADR Process Overview



HADR Synchronization modes

- A transaction on the primary machine receives a positive return code on its commit attempt when ... :
  - Synchronous: all relevant log records are externalized to disk on the standby machine. Writes to disk on primary before sending to standby.
  - Near Synchonous: all relevant log records are received by the standby database. Log pages are sent to the network before writing to primary disk.
  - Asynchronous: all relevant log records have been sent to the standby machine.



HADR Setup Procedure

- Prepare and clone the primary database and configure for HADR
  - Backup or Split Mirror copy
- Create the standby database from the clone
- Configure the standby database for HADR
- Configure client reroute if desired
- Start-up HADR

Implementation Architecture with DB2 HADR

- DB2 HADR implemented since July 2006
- AIX 5.3, HACMP 5.4, DB2 9.1 FP3
- One 2Gb Etherchannel network for SAP Application Servers, Tivoli Storage Manager and between HADR nodes
- We use scripted IP address takeover, instead of DB2 Client Reroute to avoid SAP Application Server Split Brain Problem
- Virtual IP addresses for DB2 database server and SAP enqueue/message server are controlled via HACMP
- The DB2 resource group in HACMP contains the additional NFS resources required for SAP's /sapmnt/<SID>/global and others
- SAP's Replicated Enqueue Server is running on the Standby Server

OLTP DB2 Database Layout/Configuration

- DMS file containers, autoresize enabled
- 200+ table spaces using 16K or 32K pagesize
- 100+ GB for /db2/<SID>/log_dir
- STMM is currently not used

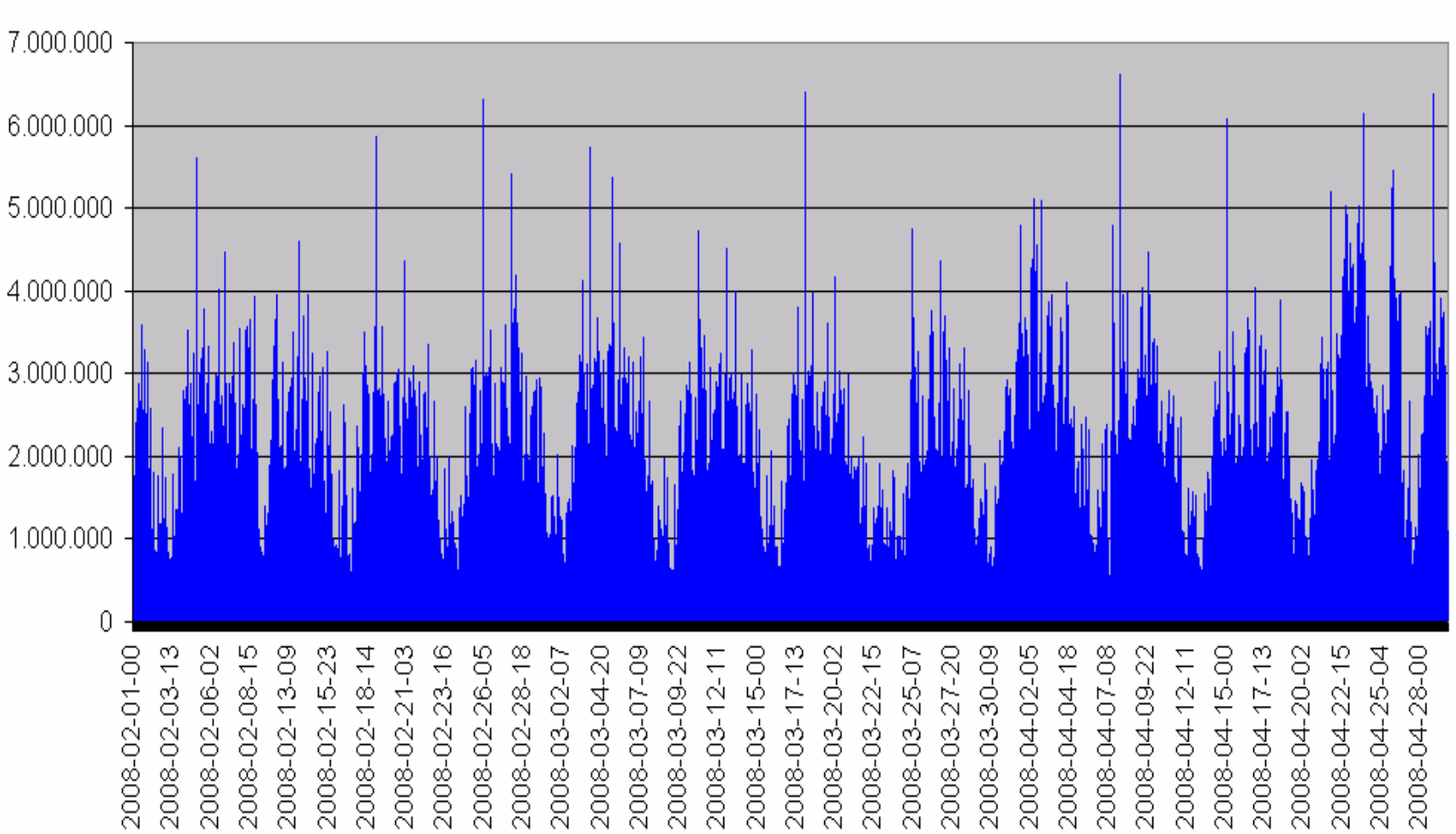
Current System Sizing and SAP Workload

- Our System sizes for standalone DB servers
  - 128-256 GB Ram (RAM = 2-3% of used DB size)
  - 16-48 Power5 CPUs, p595 @ 1.9 GHz
  - 7-12 TB allocated for tables/indexes, on disk up to 15 TB
- Workload
  - SAP R/3 4.7, up to 30 Million SAP dialog steps per week
  - 3700 highly concurrent / active SAP users in peak hours
  - A maximum of 30 GB DB2 logs / hour, 375 GB DB2 logs / day
- Top objects within the HADR based systems
  - 18 tables > 100 GB; 120 tables > 10 GB (typical DB)
  - Indexes with > 400 GB for single tables
  - Total index space of up to 5 TB per DB

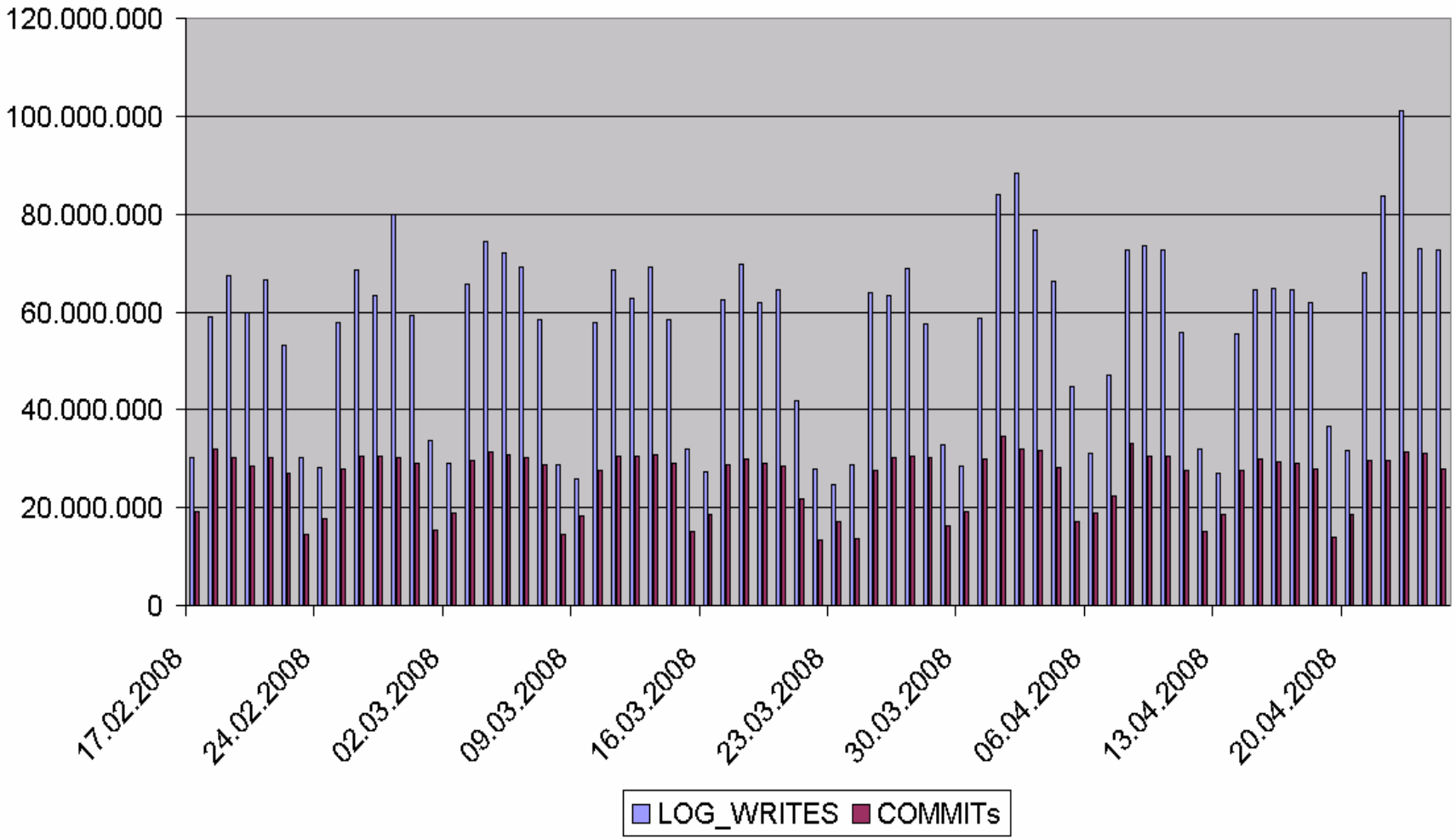
Failover time with HADR

- Unplanned Failover time
  - less than 5 minutes, typically < 1 minute
- Planned Failover time
  - less than 5 minutes, typically < 1 minute
- Standby catchup performance
  - 50+ GB/h

Thoughput today – db2 log writes per hour



Thoughput today – log writes and commits per day



DB2 HADR Configuration Details

- LOGBUFSZ is currently 64 MB for large systems
  - To avoid log full situations
  - All small rollbacks occur within the DB2 log buffer – no log reads
- HADR SYNC Mode is used due to business governance
- HADR_PEER_WINDOW is used.
  - New DB CFG parameter formally introduced with DB2 9.5
  - This been made available for DB2 V9.1 as early customer program
  - Set to 120 sec but you should evaluate the correct duration for yourself
  - Scenario: 1) the primary is falling out of peer state and more transactions are being processed, 2) Takeover to the standby -> potentially transactions can be lost
  - The Peer Window is providing a safe failover feature for automated takeovers

Thoughput today – summary

- The slides presented show stable system performance after tuning
- The following slides provide hints for monitoring and troubleshooting

Insight: What is HADR Congestion ?

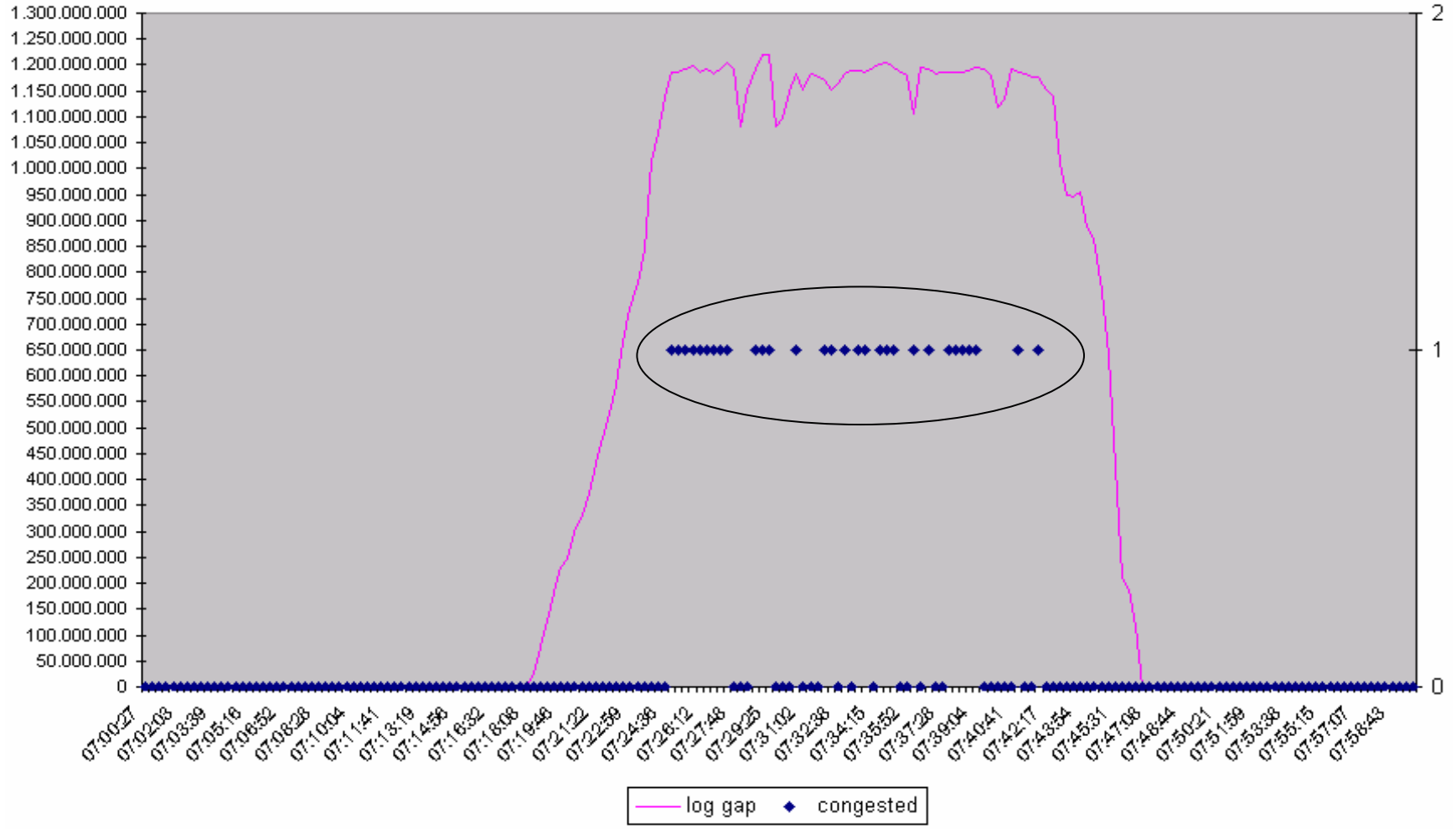
- If the log replay on secondary is - for some reason - slower than the log stream on the primary, the HADR state can be getting "Congested"
- Example output 'db2 get snapshot for db on $sid |grep -p HADR':

```
HADR Status
Role = Primary
State = Peer
Synchronization mode = Sync
Connection status = Congested, 08/30/2007 13:08:10.067957
Peer window end = 08/30/2007 13:09:39.000000 (1188479379)
Peer window (seconds) = 90
Heartbeats missed = 0
Local host = deadb137
Local service = DB2_HADR
Remote host = deadb138
Remote service = DB2_HADR
Remote instance = db2eb7
timeout(seconds) = 30
Primary log position(file, page, LSN) = S0163540.LOG, 254685, 000046763D47ACB7
Standby log position(file, page, LSN) = S0163540.LOG, 186686, 000046762CADB85F
Log gap running average(bytes) = 272.184.902
```

Insight: What is HADR Congestion ?

- Congestion is detected by the HADR Sender process, if it cannot send a network packet
- It can occur for a few microseconds and will not be noticed
- There is user impact if congestion builds up for a longer time period. Result: commit times will increase dramatically
- Cyclic DB snapshots are a good base for investigation
- The number of Congestion events can be exactly measured via db2 trace on the HADR process
- A trace can also show the flush size from the log buffer to the network
  - size of network transmit operations
  - number of congestions events

Example: Log gap and Congestion by snapshot monitoring



HADR Congestion – Possible Causes (1)

- Network overload, therefore review capacity utilization
- Network buffers are too small - TCP/IP Tuning
  - Many of the general network options ("no" command) are only active if option use_isno is disabled. If use_isno is enabled, the AIX default, the settings must be configured at the interface level.
  - New in DB2 9.1 FP5: DB2_HADR_SOSNDBUF and DB2_HADR_SORCVBUF registry variables
- HADR standby receive buffer is too small
  - The default receive buffer size is calculated to be 2 x LOGBUFSZ.
  - Increase DB2_HADR_BUF_SIZE to prevent the log gap to reac the size of the receive buffer.
  - APAR IZ10679 will allow more than 2GB LOGBUFSZ

HADR Congestion – Possible Causes (2)

- LOG replay on the standby is too slow ?
  - Issues with page cleaning on primary and/or standby (SOFTMAX)
    - Configuration may cause no constant page cleaning on the standby
    - Pagecleaning may work differently on the standby, as no sorting/temp activity is needed.
    - Good experience with "low“, e.g. aggressive softmax, For example log space represented by softmax = 2-3 % of the total buffer pool size on very large systems
  - Too many rollforward agents on the standby (eventually APAR **IZ03423** may help)
    - A system with 96 logical CPU's will cause 96 DB2REDOW to be created. Out tests have shown best performance with up to 16 redo workers. (DB2BPVARS configuration example)
    - To test: deactivate standby until you log gap reached a specific point. Let the standby catch up, measure time, ensure logs are read through local catchup.

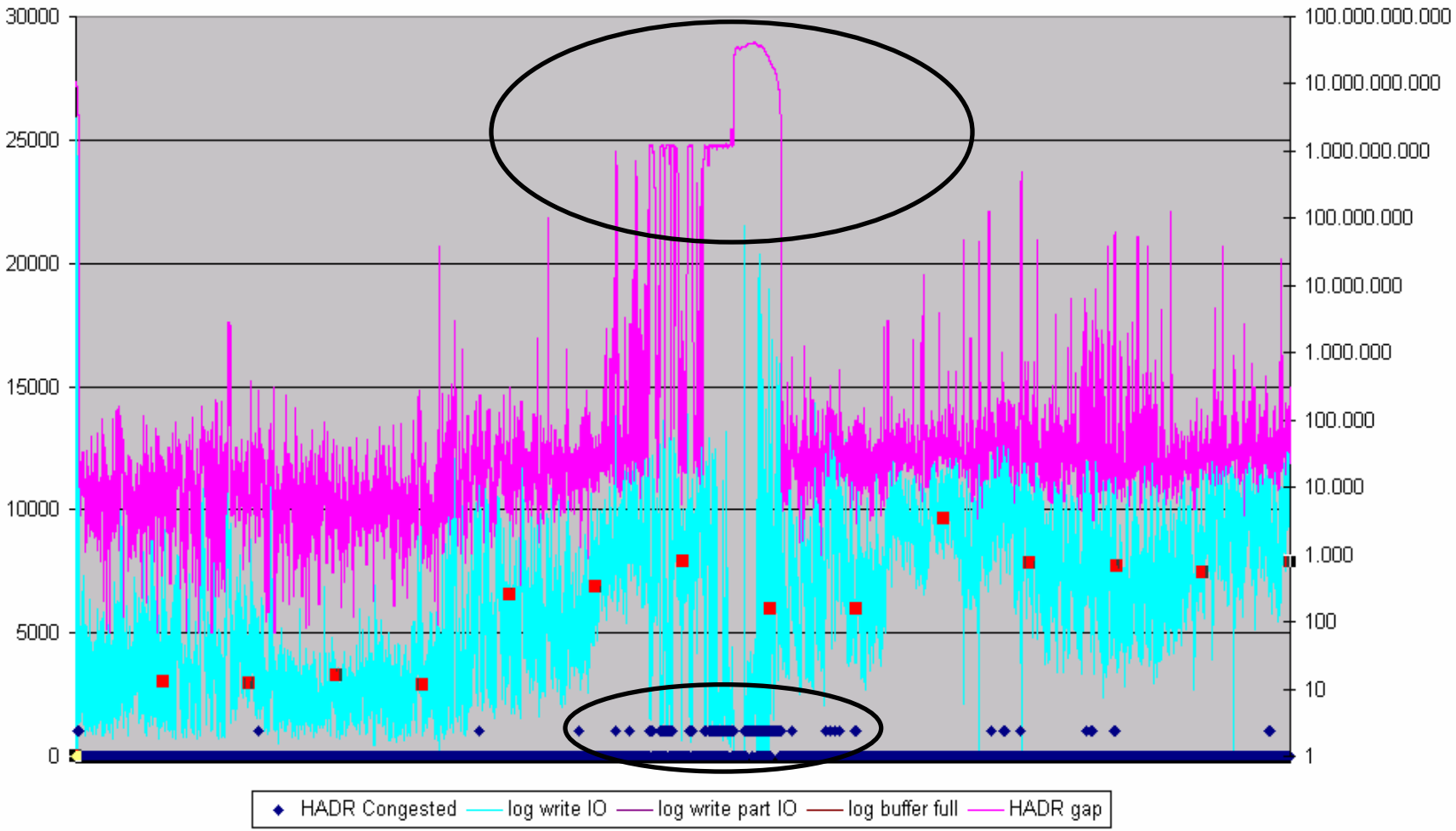
HADR Congestion – Possible Causes (3)

- I/O too slow?
  - How to detect slow I/O on the standby?
    - You can expect, if the disk setup of the standby is on a different storage subsystem, but identical with the primary DB server, it is unlikely to have I/O issues.
      The standby needs to perform a lot less of read I/O than the primary, which most cases reflects the majority of OLTP workload.
    - Investigate disk I/O on the primary!
- Reduction of serialized operations on the standby
  - APAR **IZ14163** PROVIDE INCREASED CONCURRENCY TO PARALLEL RECOVERY FOR LOG RECORDS ASSOCIATED WITH ADDING AN EXTENT TO A TABLE.

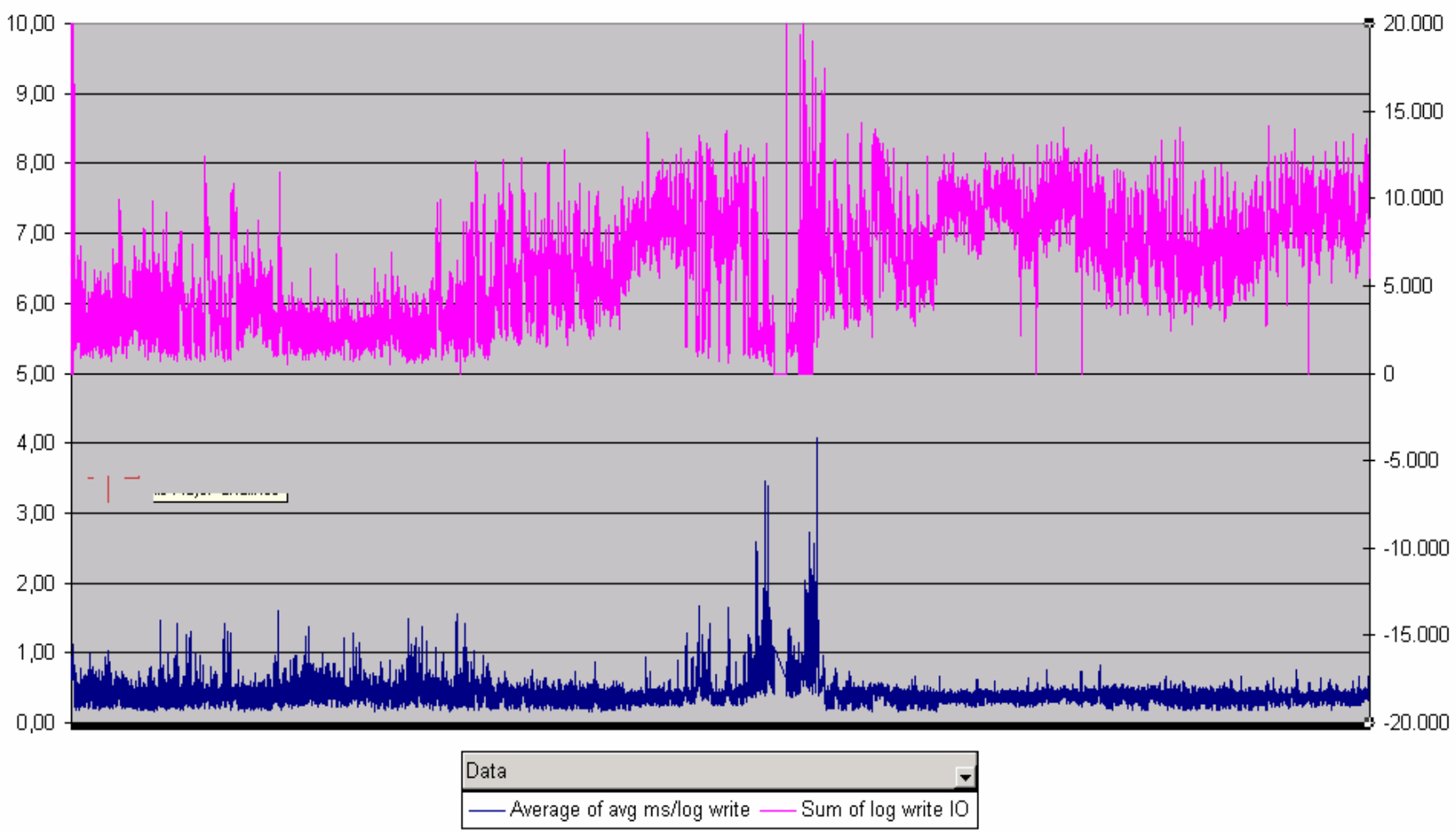
HADR Congestion – Monitoring Instructions

- AIX base monitoring
  - nmon, collection interval 60 seconds
- DB2 base monitoring
  - Primary: DB snapshots into table, every 60-300 seconds
  - Standby: DB snapshots into flat file, every 60-300 seconds
  - HADR related information on primary and standby: every 30-60 seconds into flat file
- DB2 troubleshooting
  - DB + Application snapshots on the primary
  - db2pd –hadr / db2pd –ihadr (primary + standby, may require service password)
  - db2trc on -m "*.*.HDR.*.*" -l 512M
  - db2pd –pages (bufferpool) / -latches / -stacks

Example: log write, gap, congestion



Example: standby log writes and response time



Deactivating the DB2 HADR Standby

- Deactivating the standby can impact the primary if the buffer pool is large and there are many dirty pages
  - The standby buffer pool is cleaned in the beginning of the standby deactivation
  - Optimizations have been made to optimize page cleaning on the standby
- DB2 development is working on a new tool to flush the bufferpool before the DBA issues a "deactivate db"
  - Achieves minimal deactivation time, even if you have large buffer pools

Impact of Long Running Transactions

- Long running transactions – infrequent commit
  - may impact: page cleaning
  - deactivation / re-activation of the standby

  - Take DB and application snapshots in case of troubleshooting

Table/Index REORG Considerations

- Reorgs are important for performance, most important for indexes.
  - HADR Considerations
    - runtime, transaction duration
    - logging volume
- Index Reorgs
  - Always use reorg indexes command with “allow write access” (read access is the default)
- Table reorgs
  - run traditional reorgs as far as it is possible (table+indexes < user log space)
  - inplace reorgs always with "notruncate table"
- Table moves to new table spaces
  - The online table move stored procedure is available for SAP Customers. Please consult SAP note 1039544 for details, and see SAP note 362325 for instructions regarding DB6CONV

DMS FILE Containers and CIO

- If DMS file containers are being used on AIX, it is very important to use CIO and jfs2
  - if systems are large (64+ GB Ram)
  - to prevent file concurrency issues (inode locks)
  - issues within the file system cache (AIX vmo settings, lrud daemon)
- CIO allows better scalability under high workload
- Ensure you are current on AIX maintenance
  - E.g. AIX 5.3 TL6 SP4 or later
    - currently we are evaluating/verifying AIX 5.3 TL8 SP1

AIX Configuration Examples

- Most important AIX "no" (network options)
  - use_isno 0 – or configure per network interface
  - tcp_recvspace 256K
  - tcp_sendspace 256K
  - sb_max 1280K
- Use Jumboframes if possible

DB2 Configuration Examples

- Configure rollforward worker processes:
  - db2set DB2BPVARS=/db2/db2lab/sqllib/db2bpvars.cfg
  - Enter the following line in /db2/db2lab/sqllib/db2bpvars.cfg:
    PREC_NUM_AGENTS=3
    - Reduces number of parallel recovery agents
- HADR Receive buffer (depending on system size), e.g. :
  - db2set DB2_HADR_BUF_SIZE=41940
  - up to 2 GB
- SOFTMAX (LSN pagecleaning)
  - is representing a small portion of the bufferpool on large systems
  - e.g. 128 GB bufferpool
    LOGFILSIZ * SOFTMAX = 2-3 % of the bufferpool

Important Other DB2 Configuration Parameters

- LOCKLIST – in this example 500+ MB
- MAXLOCKS not higher than 22 (effectively 44% of the locks for one agent under DB2 V8 and DB2 V9 on AIX/64 bit)
- LOG SPACE – majority with primary log files
- MAX_LOG – limit the maximum transaction size, e.g. 5 GB
- NUM_LOG_SPAN – force off orphaned transactions, e.g. set it to 80% of LOGPRIMARY
- SOFTMAX / CHNGPGS_THRESH – threshold and LSN gap triggers should each fire at 50%, if possible. LSN is most critical for standby page cleaning.

DB2 9.5 Enhancements

- DB2 9.5 contains many enhancements relevant to this environment, including
  - Reduced footprint from the threaded engine architecture
  - Automatic compression
  - MDC rollout enhancements
  - Optimizer improvements
  - db2pd enhancements
- Specifically for HADR we have enhanced DB2 with
  - Integration of TSA (more on this on the next slide)
  - DB CFG: HADR_PEER_WINDOW, HADR_TIMEOUT
    - Including DB2_HADR_PEER_WAIT_LIMIT registry variable **(V9.1 FP4)**

HADR Enhancements in DB2 9.5

- For Linux and AIX, Tivoli System Automation for Multiplatforms (TSA MP) is now bundled with and installed by DB2

- db2haicu utility is used to specify the cluster manager configuration
  - Create a cluster domain
    - Define local DB2 instance resource groups on each node
      - Note that the default behavior is to restart the instance in place by default
    - Define quorum device for the domain
    - Define DB2 HADR database resource across the cluster, online on the primary and offline on the standby
  - Define the network interface to the cluster domain
    - Optionally, define network resource for a floating virtual IP

- db2pd –ha shows the cluster manager status

Summary

- Using DB2 HADR allows you to maintain excellent application performance and maximum database availability
  - Proven in large-scale production environment
- DB2 HADR is simple to configure
  - Further improvements have been delivered in DB2 9.5
- Tuning on application level remains the most important thing to do. With good application setup, there is little tuning work required at the DB2 level.
- We have provided both diagnostic procedures and tuning tips to assist you in your production environments

Thank you !

Feedback and questions are welcome!

Stefan Krämer info@skc-group.com

Berni Schiefer schiefer@ca.ibm.com

Relevant APARs to Consider in High-END Configs

- **IZ19059** - Tablespace status "reorg in progress" remains on HADR takeover
- **IZ10679** - SETTING DB2_HADR_BUFF_SIZE TO VALUE OF 2 Gb OR LARGER RESULTS INT THE HADR CONNECTION BEING DROPPED
- **IZ14163** - PROVIDE INCREASED CONCURRENCY TO PARALLEL RECOVERY FOR LOG RECORDS ASSOCIATED WITH ADDING AN EXTENT TO A TABLE.
- **IZ07906** - DB2 V9 FP5 - SLOW DEACTIVATION OF A DATABASE DUE TO TOO MANY DIRTY PAGES IN BUFFERPOOL
- **DB2 V9 FP5** (PCR030310) - Ability to tune TCP send/receive buffer size. Mini-feature for reporting hadr buffer usage through db2pd

IBM Information

> > > On Demand

June 3 - 6, 2008 The Hague, Netherlands

EUROPE, MIDDLE EAST & AFRICA IOD CONFERENCE

2008