

## Willkommen zum „IBM DB2 Newsletter“

Liebe Leserinnen und Leser,

ich freue mich, sie zur November Ausgabe des DB2 Newsletter zu begrüßen. Danke an alle, die sich bereits für den Newsletter angemeldet haben.

Unsere Kontaktadresse: [db2news@de.ibm.com](mailto:db2news@de.ibm.com)

Obwohl wir in der Oktober-Ausgabe angekündigt haben, über BCU und weitere Kennzahlen zu schreiben, haben wir diese Themen noch einmal verschoben und dafür eine Fortsetzung der Artikelserie über HADR und aus aktuellem Anlass Techtipps über DB2 und Unicode und neue Merkmale in DB2 aufgenommen. Die angekündigten Artikel sind aufgeschoben und nicht aufgehoben. Also bitten wir noch um etwas Geduld.

Ihr TechTeam

## Inhaltsverzeichnis

<b>WILLKOMMEN ZUM „IBM DB2 NEWSLETTER“</b> .....	<b>1</b>
<b>CHATS MIT DEM LABOR</b> .....	<b>2</b>
<b>ARTIKELSERIE: HADR II – HADR IM DETAIL</b> .....	<b>2</b>
BEGRIFFSBESTIMMUNG .....	2
FUNKTIONSWEISE .....	2
EINSCHRÄNKUNGEN .....	5
<b>TECHTIPP: DB2 UND UNICODE</b> .....	<b>6</b>
<b>TECHTIPP: BUFFERPOOL-HITRATIO</b> .....	<b>7</b>
<b>TECHTIPP: PASSWORT ÄNDERUNG IN DB2</b> .....	<b>7</b>
<b>SCHULUNGEN</b> .....	<b>8</b>
DB2 FÜR LUW - NEUE FUNKTIONEN IN VERSION 9.1 (CG09D1DE) .....	8
DB2 FÜR LINUX, UNIX, WINDOWS - NEUE FUNKTIONEN IN VERSION 9.2 (CG09D2DE) .....	8
<b>NEWSLETTER ARCHIV</b> .....	<b>9</b>
<b>ANMELDUNG/ABMELDUNG</b> .....	<b>9</b>
<b>AUSBLICK AUF WEITERE THEMEN</b> .....	<b>9</b>
<b>DIE AUTOREN DIESER AUSGABE:</b> .....	<b>9</b>

## Chats mit dem Labor

In regelmäßigen Abständen finden im Internet Chats mit dem DB2-Labor statt.

Der letzte Chat mit dem Labor fand am 07. November zum Thema: „DB2 Warehouse 9.5 - What's New in the Platform for High Performance DB2 Analytics“

Diese sind zu finden unter: <http://www.ibm.com/db2/labchats>

Eine Liste der bereits durchgeführten Chats ist zu finden unter:

<http://www-306.ibm.com/software/data/db2/9/labchats.html>

Die Präsentationen der Chats, können als pdf angeschaut und runtergeladen werden.

## Artikelserie: HADR II – HADR im Detail

### ***Begriffsbestimmung***

<b>Begriff</b>	<b>Beschreibung</b>
Primary	Die aktive Datenbank in einem HADR-Verbund
Standby	Die inaktive Datenbank in einem HADR-Verbund
Peer State	HADR auf Primary und Standby ist aktiv, beide Systeme kommunizieren miteinander

### ***Funktionsweise***

#### **Granularität**

HADR arbeitet auf Datenbank-Ebene. Innerhalb einer Instanz können mehrere Datenbanken (als Primary- und Standby-Datenbanken) existieren. Für jede Primary-Datenbank gibt es eine einzige Standby-Datenbank. Die Primary- und Standby-Datenbank müssen in verschiedenen Instanzen liegen. Die Instanzen dürfen auf demselben System oder auf einem Remote-System liegen.

#### **Wie arbeitet HADR?**

Nachdem HADR erfolgreich aufgesetzt wurde, befinden sich beide Datenbanken im sogenannten *Peer State*.

Im *Peer State* versendet die Primary-Datenbank ihre anfallenden logischen Protokolle an die Standby-Datenbank. Der Zeitpunkt des Versendens hängt direkt mit dem hinausschreiben des Puffers der logischen Protokolle auf die Platte zusammen.

Der Puffer der logischen Protokolle bei Eintreten verschiedener Ereignisse auf Platte geschrieben. Dazu gehören:

- Der Protokoll-Puffer (logbufsz) ist voll
- Eine Transaktion sendet ein *commit*

- Ein automatisches *commit* wird ausgelöst (siehe *MINCOMMIT*-Konfigurationsparameter)
- Durch *Write Ahead Logging (WAL)* - es wird sichergestellt, dass keine veränderte Datenpage auf Platte geschrieben wird, bevor die Änderungsprotokolle geschrieben sind)
- Beim Deaktivieren der Datenbank
- Durch Ausführen des Kommando "*db2 archive log*"

Die Standby-Datenbank empfängt den Protokollpuffer und pflegt die Änderungen auf dem Standby-System ein. Da das Einpflegen der Protokolle über den Standard-Wiederherstellungs-Mechanismus der Datenbank läuft, werden alle geänderten Daten-Pages automatisch im Bufferpool zur Verfügung gestellt. Dies ist ein wichtiger Punkt bei einem Takeover oder Failover.

Der genaue Zeitpunkt der Übertragung und die Rückgabe der Kontrolle an die Applikation wird über den Konfigurations-Parameter *HADR\_SYNC\_MODE* bestimmt:

- **ASync**  
Die Primary-Datenbank versendet den Protokollpuffer gleichzeitig mit dem Schreiben auf Platte. Die Steuerung (erfolgreiches *Commit*) wird an die Applikation zurückgegeben, nachdem das lokale Schreiben der Protokolle erfolgreich war und die Sende-Operation durchgeführt wurde.
- **NEAR\_SYNC**  
Die Primary-Datenbank versendet den Protokollpuffer gleichzeitig mit dem Schreiben auf Platte. Die Steuerung (erfolgreiches *Commit*) wird an die Applikation zurückgegeben, nachdem das lokale Schreiben der Protokolle erfolgreich war und die Standby-Datenbank den Empfang quittiert hat.
- **SYNC**  
Die Primary-Datenbank versendet den Protokollpuffer **nachdem** das lokale Schreiben der Protokolle erfolgreich war. Die Steuerung (erfolgreiches *Commit*) wird an die Applikation zurückgegeben, nachdem die Standby-Datenbank das erfolgreiche Schreiben der Protokolle auf Platte quittiert hat.

## Welcher Sync-Modus soll gewählt werden?

Der *SYNC*-Modus bietet die höchste Sicherheits-Stufe. Im *Peer State* wird garantiert, dass alle auf dem Primary festgeschriebenen Transaktionen auch auf dem Standby festgeschrieben wurden. Da alle Aktionen (lokales schreiben, übertragen, fernes schreiben) seriell ausgeführt werden, können negative Auswirkungen auf die Applikations-Performance nicht ausgeschlossen werden.

Der *NEAR\_SYNC*-Modus ist der optimale Kompromiss zwischen höchster Datensicherheit und maximaler Applikations-Performance. Für die meisten Anwendungsfälle dürfte dieser Modus ausreichend sein. In diesem Modus besteht die potentielle Gefahr eines Verlusts von bereits abgeschlossenen Transaktionen, wenn sowohl Primary als auch Standby nahezu gleichzeitig ausfallen. Im Einzelnen müssen folgende Bedingungen zutreffen:

- Die Primary-Datenbank fällt aus, nachdem sie die Quittierung des Standby erhalten hat und diese Information an die Applikation zurückgegeben hat
- Die Standby-Datenbank fällt aus, bevor die bereits empfangen logischen Protokolle auf die Platte geschrieben werden konnten

Der *ASYNC*-Modus sollte nur verwendet werden, wenn die Applikations-Performance durch die beiden Sync-Modus nicht erreicht wird. In diesem Modus besteht die potentielle Gefahr eines Verlusts von bereits abgeschlossenen Transaktionen, wenn sowohl Primary als auch Standby nahezu gleichzeitig ausfallen. Im einzelnen müssen folgende Bedingungen zutreffen:

- Die Transaktion sendet ein *Commit*
- Während des Schreibens der logischen Protokolle auf dem primären System bricht die Netzwerk-Verbindung zur Standby-Datenbank ab
- Die Primary-Datenbank quittiert den erfolgreichen *Commit* an die Applikation
- Die Primary-Datenbank fällt aus

**Wichtig: Da es sich um eine Hochverfügbarkeits-Konfiguration handelt, führt der Ausfall der Standby-Datenbank (auch während einer offenen Transaktion) nicht zum Abbruch der Applikation. Die Primary-Datenbank arbeitet dann autark weiter (unabhängig vom Wert des Konfigurations-Parameters *HADR\_TIMEOUT*. Der Ausfall der Standby-Datenbank wird bei aktiven Applikationen in der Regel sofort erkannt - durch den Erhalt eines TCP-Fehlers beim Versenden der Transaktionen).**

## Failover oder Neustart des Primary?

Soll nach einem Ausfall der Primary-Datenbank versucht werden diese erneut anzustarten, oder soll gleich ein Failover auf die Standby-Datenbank durchgeführt werden.

Aus der HADR-Theorie gesehen, sollte jeder Ausfall der Primary-Datenbank zu einem Failover auf die Standby-Datenbank führen. In der Praxis hängt diese Frage jedoch von folgenden Faktoren ab:

- Welcher Sync-Modus wurde gewählt?
- Welche Ausfallzeiten der Applikation sind tollerierbar?

Der Vorteil von HADR ist die Möglichkeit in sekundenschnelle von Standby- auf Primary- oder zurück in den Standard-Betrieb zu schalten. Zusätzlich sind die Bufferpools bereits mit den aktuellen Daten gefüllt.

Bei grossen Datenbanken kann ein Neustart schon einmal mehrere Minuten dauern. Danach besteht immer noch das Problem, dass die Bufferpools nicht gefüllt sind. Eine Negative Performance, über einen gewissen Zeitraum, kann die Folge sein.

Im *SYNC*-Modus empfiehlt sich der sofortige Failover, da Transaktionsverluste ausgeschlossen sind, solange sich die HADR-Partner im *Peer*-Status befunden haben.

Auch für den *NEARSYNC*-Modus kann ein sofortiger Failover durchgeführt werden, da Transaktionsverluste ebenfalls ausgeschlossen sind, solange sich die HADR-Partner im `_Peer_Status` befunden haben.

Bei Verwendung des *ASYNC*-Modus wäre der Versuch eines Neustarts der Primär-Datenbank einem sofortigen Failover

## **Einschränkungen**

Für den Einsatz von HADR gelten nachfolgend beschriebene Einschränkungen.

### **Gleiche Hardware und gleiche DB2 Version**

Die Systeme, die den Primary und den Standby beherbergen, müssen mit der gleichen Hardware und mit identischer Betriebssystem-Version ausgestattet sein. Die DB2 Version darf nur während eines Versions-Upgrades abweichen.

### **Redirected Restore für Tablespaces**

Ein Tablespace-Restore wird nicht unterstützt.

### **Keine Unterstützung für DPF**

Eine Datenbank mit aktivem DPF (Database Partitioning Feature) kann nicht über HADR gesichert werden.

### **Eingeschränkte Replikation von LOBS**

Es werden nur solche LOBS repliziert, deren Veränderungen in den logischen Protokollen mitgeschrieben werden. Ob die Veränderungen von LOBS mitgeschrieben werden kann bei der Spaltendefinition angegeben werden.

Aber: LOBS deren Größe 1 GB übersteigt, werden generell nicht protokolliert. Eine Replikation von LOBS größer 1GB ist somit nicht möglich.

### **Aktivitäten die nicht repliziert werden**

Durch HADR werden nur diejenigen Aktivitäten repliziert, die über die logischen Protokolle mitgeschrieben werden. Dazu gehören auch das Anlegen eines Tablespace und von Bufferpools.

Nicht repliziert werden u. a.

- Datenbank-Konfigurationsparameter
- Änderungen die mit *NOT LOGGED INITIALLY* durchgeführt werden
- UDF-Bibliotheken
- Änderungen im History-File
- LOAD-Operationen mit Option *COPY NO*

## Einschränkungen für die Standby-Datenbank

Für die Standby-Datenbank gelten zusätzlich, da diese im Rollforward Pending-Modus gehalten wird, folgende Einschränkungen:

- Keine Archivierung der logischen Protokolle
- Kein Backup möglich
- Kein Applikationszugriff möglich

## TechTipp: DB2 und Unicode

Bei der Umstellung einer Datenbank von den "normalen" Zeichensätzen (SBCS) auf Unicode ist die Sortierung (Collating Sequence) aus Performancegründen genau zu prüfen. DB2 erlaubt verschiedene Varianten, aus denen sorgfältig ausgewählt werden muß:

- UTF-8 mit Identity (=binärer Sortierung)  
Dies ist der Default für Unicode und wird meist verwendet (z.B. bei SAP). Die Performance entspricht weitestgehend der von SBCS. Allerdings steht damit keine landesspezifische Sortierung mehr zur Verfügung.
- UTF-8 mit UCA400\_\* (=aufwändige Sortierung)  
Die Sortierung der Daten ist mit UCA400 extrem ausgefeilt. Dies ist das, was eigentlich von Kunden oder Anwendern gewünscht wird. Es kann jedoch zu extremen Performanceproblemen bei der Verwendung von UCA400 kommen. Ein erhöhter CPU Bedarf um den Faktor 10 kann durchaus auftreten. Die Implementierung von UCA400 in DB2 9.5 ist wesentlich gegenüber DB2 9.1 verbessert. Der Overhead ist trotzdem immer noch dramatisch hoch. Hier muß also sorgfältig zwischen Funktionalität und Aufwand abgewogen werden.

Die Probleme wurden vom Labor erkannt und daher steht mit DB2 9.5 eine weitere Sortierung für Unicode zur Verfügung:

- UTF-8 mit SYSTEM\_csid\_territory  
Mit dieser Einstellung wird für die landesspezifischen Zeichen die Sortierung erhalten und für alles andere wird auf die binäre Sortierung zurückgegriffen. Dies erlaubt die Nutzung von Unicode mit der vom SBCS gewohnten Sortierung und eine sehr gute Performance.

Wichtig: Diese Einstellungen lassen sich ausschließlich beim Anlegen der Datenbank vornehmen!

Näheres dazu findet man hier:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.cmd.doc/doc/r0001941.html>

## TechTipp: Bufferpool-Hitratio

Nach der Zusammenstellung des letzten DB2 NL mit dem kleinen Skript zum BPHR bekamen wir die Info, das BP-Hit auch mittels den Table-Funktionen abgefragt werden kann. Dazu kann folgendes SQL-Statement verwendet werden:

```
select substr(bp_name,1,20) as BP_NAME,  
       int (( 1 - (decimal(pool_data_p_reads) /  
                nullif(pool_data_l_reads,0) )) * 100) as data_hit_ratio,  
       int (( 1 - (decimal(pool_index_p_reads) /  
                nullif(pool_index_l_reads,0) )) * 100) as index_hit_ratio,  
       int (( 1 - (decimal(pool_data_p_reads + pool_index_p_reads) /  
                nullif( (pool_data_l_reads + pool_index_l_reads),0) )) * 100)  
       as BP_hit_ratio,  
       int (( 1 - (decimal(pool_async_data_reads + pool_async_index_reads) /  
                nullif( (pool_async_data_reads + pool_async_index_reads +  
                direct_reads),0) )) * 100) as Async_read_pct,  
       int (( 1 - (decimal(direct_writes) / nullif(direct_reads,0) )) *  
                100) as Direct_RW_Ratio  
from table (snapshot_bp ('<DBNAME>', -1) ) as snapshot_bp ;
```

Wobei <DBNAME> durch ihren Namen der Datenbank ersetzt werden muss.

Diese Statement wurde aus eine Developer Artikel „The DB2 for Linux, UNIX, and Windows DBA Checklist“ entnommen.

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0404snow/index.html>

## TechTipp: Passwort Änderung in DB2

Mit dem APAR IY94490 wurde eine neue Funktionalität in DB2 UDB 8.1 FP5 und 9.1 FP3 eingeführt, die sehr nützlich in HACMP Umgebungen sein kann.

### Problem:

DB2 hat eine Funktionalität, die es dem Benutzer ermöglicht sein Passwort zu ändern ("change password feature").

Dies bedeutet, ein Benutzer kann, bei der Anmeldung an die Datenbank (connect) sein Passwort folgermassen ändern:

```
db2 connect to dbname user user1 using password1 new password2 confirm  
password2
```

Dabei wird das Passwort nur auf der aktuellen Node geändert. Wenn jedoch ein Takeover stattfindet, kann sich der Benutzer nicht länger an die Datenbank anmelden.

Es gibt derzeit keine technische Möglichkeit, die DB2 nutzen könnte um das Passwort auf allen Nodes im Cluster zu ändern.

Betroffen von diesem Problem sind Kunden, die weder LDAP noch NIS verwenden um die /etc/passwd auf allen Nodes aktuell zu halten.

### Lösung / Workaround:

DB2 hat mit APAR IY94490 eine neue Funktionalität eingeführt: "Disable the change password feature".

Durch setzen der **DB2AUTH** Registervariable (db2set) auf DISABLE\_CHGPASS wird diese Funktionalität aktiviert.

APAR IY94490 ist gefixt ab 8.1 FP15 und V9 FP3. Somit kann diese neue Funktionalität ab sofort genutzt werden.

### Beispiel:

```
db2set DB2AUTH=DISABLE_CHGPASS  
db2start
```

```
db2 connect to sb user newton using xxxx new yyyy confirm yyyy
SQL30083N Attempt to change password for user id "newton" failed with security
reason
"17" ("UNSUPPORTED FUNCTION"). SQLSTATE=08001
```

db2diag.log:

```
2007-02-06-09.01.59.696655-300 I4857C513 LEVEL: Warning
PID : 57268 TID : 1 PROC : db2agent (SB)
INSTANCE: xxxxxxxx NODE : 000 DB : SB
APPHDL : 0-7
FUNCTION: DB2 UDB, bsu security, sqlxSlsSystemAuthenticate, probe:125
MESSAGE : ZRC=0x805C0144=-2141454012=SQLEX_CHGPASS_UNSUPPORTED
"Can not change password" DATA #1 : String, 62 bytes
change password disabled via DB2AUTH registry variable. User:
DATA #2 String with size, 6 bytes newton
```

Hinweis: In HACMP existiert ein Tool (clpasswd) mit dem der Benutzer das Passwort auf allen Nodes ändern kann. Das "Disable the change password feature" von DB2 stellt sicher, daß der Benutzer auf dieses Tool oder andere passende Vorgehensweisen zurückgreift.

## Schulungen

Folgende Schulungen werde angeboten:

### ***DB2 für LUW - Neue Funktionen in Version 9.1 (CG09D1DE)***

Sie können die neuen Funktionen und Erweiterungen der DB2 für Linux, UNIX, Windows für die Version 9.1 implementieren und kennen die Auswirkungen auf Ihr System. Sie können abschätzen, wie die Neuerungen in Ihrer Datenbankumgebung Vorteile bringen und diese einsetzen.

- Migrations- und Installationsüberlegungen
- Datenbank- und Anwendungserweiterungen
- SQL Änderungen
- Selbstoptimierendes Speichermanagement (STMM)
- Label-basierende Zugriffskontrolle (LBAC)
- 'Large-RID' Unterstützung
- Datenkomprimierung
- Entwicklungsoberfläche - Developer Workbench
- XML - Überblick

Link zur Anmeldung:

[http://www-05.ibm.com/services/learning/de/ta-iris.nsf/\(ExtCourseNr\)/CG09D1DE](http://www-05.ibm.com/services/learning/de/ta-iris.nsf/(ExtCourseNr)/CG09D1DE)

Ort: München

Datum: 10.12.-12.12.2007

### ***DB2 für Linux, UNIX, Windows - Neue Funktionen in Version 9.2 (CG09D2DE)***

Sie können die neuen Funktionen und Erweiterungen der DB2 für Linux, UNIX, Windows Version 9.1 implementieren und kennen die Auswirkungen auf Ihr System. Sie können abschätzen, wie die Neuerungen in Ihrer Datenbankumgebung Vorteile bringen.

- Migrations- und Installationsüberlegungen
- Datenbank- und Anwendungserweiterungen
- SQL Änderungen
- Selbstoptimierendes Speichermanagement (STMM)
- Label-basierende Zugriffskontrolle (LBAC)
- 'Large-RID' Unterstützung

- Datenkomprimierung
- Entwicklungsoberfläche - Developer Workbench
- XML - Überblick

Link zur Anmeldung:

[http://www-05.ibm.com/services/learning/de/ta-iris.nsf/\(ExtCourseNr\)/CG09D2DE](http://www-05.ibm.com/services/learning/de/ta-iris.nsf/(ExtCourseNr)/CG09D2DE)

Ort: Stuttgart Degerloch  
Datum: 28.02.-29.02.2008

## Newsletter Archiv

Alte Ausgaben vom DB2-NL sind nun zum Nachlesen im Archiv von BYTEC zu finden:

<http://www.bytec.de/prd/sft/ibm/ibm-nlarchiv.php>

## Anmeldung/Abmeldung

Sie erhalten diesen Newsletter bis zur 3ten Ausgabe ohne Anmeldung. Wenn Sie weiterhin diesen Newsletter empfangen wollen, schicken Sie Ihre Anmeldung mit dem Subjekt „ANMELDUNG“ an [db2news@de.ibm.com](mailto:db2news@de.ibm.com).

## Ausblick auf weitere Themen

Wie bereits in der Oktober Ausgabe angekündigt: Was ist BCU und Erweiterungen zum KENNZAHLEN.sh (APPR).

## Die Autoren dieser Ausgabe:

Doreen Stein	IT-Spezialist für DB2 UDB, IBM Software Group; <a href="mailto:djs@de.ibm.com">djs@de.ibm.com</a>
Volker Fränkle	IT-Specialist for Informix Dynamic Server and DB2 UDB; IBM Software Group
Wilfried Hoge	IT-Architect, TechSales, IBM Software Group

## Reviewer und Ideenlieferanten:

Frank Berghofer	IT-Spezialist für DB2 UDB, IBM Software Group
Diana Gfrörer	Consulting IT Specialist Information Management SWG IM TechSales
Gerhard Müller	IT-Spezialist, IBM Software Group
Nela Krawez	IT-Spezialist
Peter Schurr	IT-Spezialist, IBM Software Group
Martina Lang	Techn.Support DMS, Speciality: IBM DB2 UDB Level 2 Support