DB2 Offline-Standby-Backup

mit Libelle DBShadow® und DB2 UDB

Datenbank, Datenbankspiegel und Katastrophenschutz

Mit dem vorliegenden Whitepaper soll die Funktionsfähigkeit und das Zusammenspiel beider Produkte erläutert und validiert werden. Zu diesem Zweck wurden im November 2003 Tests im IBM Solution Partnership Center in Stuttgart und im Hause Libelle durchgeführt.

Die Kombination beider Produkte stellt für unsere gemeinsamen Kunden eine Lösung dar, die die Vorzüge der marktführenden Datenbank mit denen eines Datenbankspiegels und eines umfassenden Katastrophenschutzes für die Datenbank verbindet.

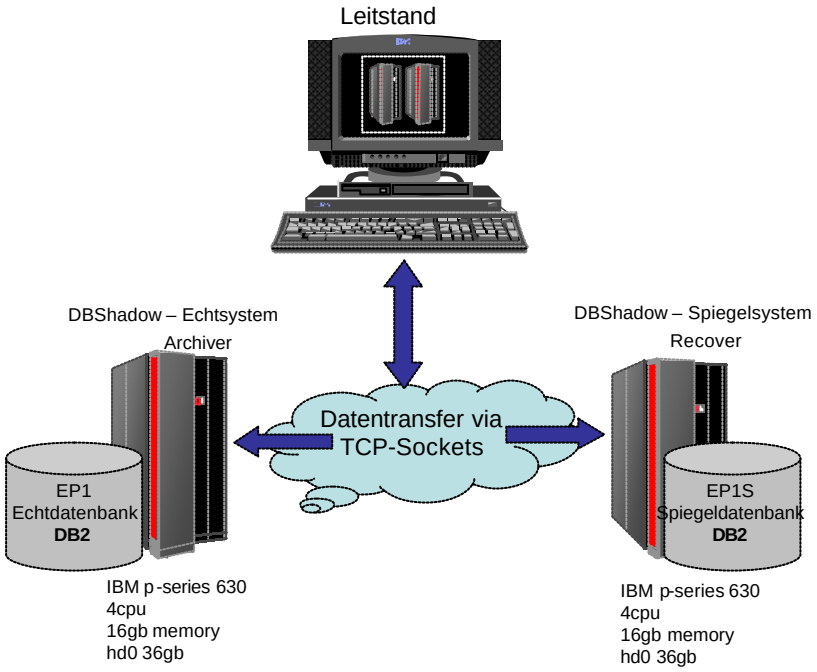
DB2 Universal Database ist das Herzstück jeder e-business Anwendung - von der Transaktionsverarbeitung über E-Commerce, Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), Supply Chain Management (SCM), Data Warehousing- und Business Intelligence-Anwendungen bis hin zu Dokumentenmanagement-Systemen. DB2 hilft Kunden, die ihr Geschäft auf e-business umstellen wollen, ihre Informationen wirkungsvoll einzusetzen. Es stellt die Leistung, Skalierbarkeit, Zuverlässigkeit und Verfügbarkeit bereit, die für anspruchsvolle e-business Anwendungen benötigt werden. Als objekt-relationale Datenbank erlaubt DB2 auch die Verwaltung multimedialer Datentypen über entsprechende DB2 Extender (XML, Bild, Audio und Video) sowie die Definition von benutzerdefinierten Datentypen und Funktionen.

DBShadow

DBShadow schützt Ihre Datenbanken vor logischen und physikalischen Ausfällen. Kernkonzept ist das zeitversetzte Spiegeln von Datenbanken.
Rund 70 % aller Ausfälle lassen sich auf Software- oder Benutzerfehler zurückführen. Hier versagen dann die klassischen Konzepte wie RAID-Technologien und Cluster-Lösungen. Denn im Falle eines Ausfalls muss hier zuerst eine Rücksicherung und ein Recovery durchgeführt werden, bevor ein Weiterarbeiten möglich ist.
Mit DBShadow® kann auch dann binnen weniger Minuten weitergearbeitet werden, wenn Datenbanken in Terabytegröße logisch oder physikalisch zerstört wurden.

Beide Produkte wurden in einer produktionsnahen Umgebung und unter realen Bedingungen getestet und haben diese Tests bestanden. Für unsere gemeinsamen Kunden sind solche Tests von hoher Bedeutung, erhöhen sie doch die Kundenzufriedenheit und geben Gewissheit bei der Entscheidungsfindung.

Bei den Tests kam folgendes Environment zum Einsatz:



Im Folgenden werden die Aktivitäten im IBM SPC erläutert. Analog wurde im Hause Libelle mit AIX 4.3.3, AIX 5.1 und DB2 Enterprise Server Edition 7.2 getestet und gleiche Ergebnisse erzielt.

Technisches Environment

Zum Einsatz kamen zwei IBM eServer p630 mit 4 CPUs und einem Hauptspeicher von 16 GB. Beide Maschinen waren mit einer internen Festplatte mit einer Kapazität von 32 GB ausgerüstet. Das installierte Betriebssystem war AIX 5.2.0.10 (64bit).
DBShadow Release 4.0.9 wurde in der Business Edition installiert.
Die eingesetzte Datenbank war eine DB2 UDB Enterprise Server Edition Version 8.1.2
Die Kommunikation beider Maschinen wurde über TCP/IP realisiert:

Name	IP-Adresse	
P630-3	172.17.40.17	(Echtsystem)
P630-4	172.17.40.23	(Spiegelsystem)

Vorbereitung zum Test

Die Datenbankinstallation wurde unter dem Verzeichnis /usr/opt/db2_08_01 auf der p630-3 durchgeführt und es wurde ein Response-File (db2ese.rsp) erzeugt. Mit Hilfe dieses Response-Files wurde auf der zweiten Maschine (p630-4) silent installiert.
Der Instance User heisst db2inst1(default), sein Home Verzeichnis ist /home/db2inst1.
Für die Tests wird natürlich eine Datenbank benötigt, welche mit Hilfe eines Skirpts der Firma Libelle (siehe Anhang) erstellt wurde. Die Struktur der Datenbank wurde einem SAP-System nachempfunden.
Es wurden „system managed“ und „database managed“ Tablespaces verwendet.
Die Datenbank EP1 wurde in dem Local Database Verzeichnis /db2/EP1 angelegt.

Hier die Ausgabe des Kommandos „db2 get db cfg“:

```
System Database Verzeichnis

 Number of entries in the Verzeichnis = 1

Database 1 entry:

 Database alias                       = EP1
 Database name                        = EP1
 Local database Verzeichnis             = /db2/EP1
 Database release level               = a.00
 Comment                              =
 Verzeichnis entry type                 = Indirect
 Catalog database partition number    = 0

       Database Configuration for Database

 Database configuration release level                   = 0x0a00
 Database release level                                 = 0x0a00

 Database territory                                     = US
 Database code page                                      = 819
 Database code set                                       = ISO8859-1
 Database country/region code                           = 1

 Dynamic SQL Query management           (DYN_QUERY_MGMT) = DISABLE

 Discovery support for this database       (DISCOVER_DB) = ENABLE

 Default query optimization class         (DFT_QUERYOPT) = 5
 Degree of parallelism                      (DFT_DEGREE) = 1
 Continue upon arithmetic exceptions   (DFT_SQLMATHWARN) = NO
 Default refresh age                   (DFT_REFRESH_AGE) = 0
 Number of frequent values retained     (NUM_FREQVALUES) = 10
 Number of quantiles retained            (NUM_QUANTILES) = 20

 Backup pending                                         = NO

 Database is consistent                                 = NO
 Rollforward pending                                    = NO
 Restore pending                                        = NO

 Multi-page file allocation enabled                     = NO

 Log retain for recovery status                         = RECOVERY
 User exit for logging status                            = YES

 Data Links Token Expiry Interval (sec)      (DL_EXPINT) = 60
 Data Links Write Token Init Expiry Intvl(DL_WT_IEXPINT) = 60
 Data Links Number of Copies             (DL_NUM_COPIES) = 1
 Data Links Time after Drop (days)        (DL_TIME_DROP) = 1
 Data Links Token in Uppercase                (DL_UPPER) = NO
 Data Links Token Algorithm                   (DL_TOKEN) = MAC0

 Database heap (4KB)                            (DBHEAP) = 1200
 Size of database shared memory (4KB)  (DATABASE_MEMORY) = AUTOMATIC
 Catalog cache size (4KB)              (CATALOGCACHE_SZ) = (MAXAPPLS*4)
 Log buffer size (4KB)                        (LOGBUFSZ) = 8
 Utilities heap size (4KB)                (UTIL_HEAP_SZ) = 5000
 Buffer pool size (pages)                     (BUFFPAGE) = 1000
 Extended storage segments size (4KB)    (ESTORE_SEG_SZ) = 16000
 Number of extended storage segments   (NUM_ESTORE_SEGS) = 0
 Max storage for lock list (4KB)              (LOCKLIST) = 100

 Max size of appl. group mem set (4KB) (APPGROUP_MEM_SZ) = 30000
 Percent of mem for appl. group heap   (GROUPHEAP_RATIO) = 70
 Max appl. control heap size (4KB)     (APP_CTL_HEAP_SZ) = 128
```

```
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) =
(SHEAPTHRES)
 Sort list heap (4KB)                         (SORTHEAP) = 256
 SQL statement heap (4KB)                     (STMTHEAP) = 2048
 Default application heap (4KB)             (APPLHEAPSZ) = 256
 Package cache size (4KB)                   (PCKCACHESZ) = (MAXAPPLS*8)
 Statistics heap size (4KB)               (STAT_HEAP_SZ) = 4384

 Interval for checking deadlock (ms)         (DLCHKTIME) = 10000
 Percent. of lock lists per application       (MAXLOCKS) = 10
 Lock timeout (sec)                         (LOCKTIMEOUT) = -1

 Changed pages threshold                  (CHNGPGS_THRESH) = 60
 Number of asynchronous page cleaners     (NUM_IOCLEANERS) = 1
 Number of I/O servers                     (NUM_IOSERVERS) = 3
 Index sort flag                               (INDEXSORT) = YES
 Sequential detect flag                        (SEQDETECT) = YES
 Default prefetch size (pages)           (DFT_PREFETCH_SZ) = 32

 Track modified pages                          (TRACKMOD) = OFF

 Default number of containers                             = 1
 Default tablespace extentsize (pages)    (DFT_EXTENT_SZ) = 32

 Max number of active applications             (MAXAPPLS) = AUTOMATIC
 Average number of active applications        (AVG_APPLS) = 1
 Max DB files open per application             (MAXFILOP) = 64

 Log file size (4KB)                           (LOGFILSIZ) = 1000
 Number of primary log files                  (LOGPRIMARY) = 3
 Number of secondary log files                (LOGSECOND) = 2
 Changed path to log files                    (NEWLOGPATH) =
 Path to log files                                         = /db2/EP1/log_dir/NODE
0000/
 Overflow log path                       (OVERFLOWLOGPATH) =
 Mirror log path                           (MIRRORLOGPATH) =
 First active log file                                     = S0000472.LOG
 Block log on disk full                  (BLK_LOG_DSK_FUL) = NO
 Percent of max active log space by transaction(MAX_LOG) = 0
 Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

 Group commit count                            (MINCOMMIT) = 1
 Percent log file reclaimed before soft chckpt (SOFTMAX) = 100
 Log retain for recovery enabled               (LOGRETAIN) = RECOVERY
 User exit for logging enabled                  (USEREXIT) = ON

 Auto restart enabled                        (AUTORESTART) = ON
 Index re-creation time                         (INDEXREC) = SYSTEM (RESTART)
 Default number of loadrec sessions     (DFT_LOADREC_SES) = 1
 Number of database backups to retain    (NUM_DB_BACKUPS) = 12
 Recovery history retention (days)      (REC_HIS_RETENTN) = 366

 TSM management class                     (TSM_MGMTCLASS) =
 TSM node name                             (TSM_NODENAME) =
 TSM owner                                    (TSM_OWNER) =
 TSM password                              (TSM_PASSWORD) =
```

Die Datenbank wurde für das Archivieren der Logfiles konfiguriert. Zum Einsatz kam dabei ein von der Firma Libelle bereitgestellter USEREXIT, welcher die zu archivierenden Logfiles in das Verzeichnis /db2/EP1/log_arch kopierte.
Nun wurden die DBShadow® Server-Prozesse auf beiden Maschinen installiert und vom GUI des DBShadow®, dem sogenannten Leitstand, aus die Konfiguration angelegt (trdb2.cnf). Dabei wird ein Timedelay zwischen Echtdatenbank und Spiegeldatenbank definiert, d.h. bei logischen Fehlern auf der Echtdatenbank können diese durch ein Point-in-Time Recovery behoben werden.

Danach erfolgt ein initialer Kopiervorgang der Datenbank auf das Spiegelsystem (p630-4) bei dem die Online Backup/Restore Funktionalität von DB2 genutzt wurde.
Die Spiegeldatenbank, die nun als EP1S katalogisiert ist, befindet sich in einem „Rollforward pending = Database“ Zustand.
Die DBShadow® Prozesse Archiver und Recover sind nun aktiv.
Der DBShadow® Archiver Prozess kopiert die archivierten Logfiles von /db2/EP1/log_arch auf dem Echtsystem (p630-3) nach /db2/EP1/log_dir/NODE000 (Path to Logfiles) auf dem Spiegelsystem (p630-4).
Der DBShadow® Recover Prozess steuert das entsprechende Rollforward Recovery in Abhängigkeit vom definierten z. Zt. gültigen Timedelay.

Testszenario

1. Stoppen der DBShadow® Prozesse Archiver und Recover
2. Stoppen der db2 Instance auf dem Spiegelsystem
3. Datensicherung auf dem Spiegelsystem mit Betriebssystem Mitteln (tar, cp –r etc.)
   - sichern des Home Verzeichnisses des Instance users /home/db2inst1 und des Datenbank Verzeichnisses /db2/EP1, dieses beinhaltet das „Path to Logfiles“ Verzeichnis und alle Container der „database managed“ tablespaces
4. Starten der db2 Instance auf dem Spiegelsystem
5. Starten der DBShadow® Prozesse Archiver und Recover
6. Notumschaltung – Dieser Zustand wird erreicht über das Stoppen der db2 Instance auf der Echtdatenbank (Fehlersimulation) und durch die Auslösung der Notumschaltung im Leitstand von DBShadow®. Auf der Spiegeldatenbank wird ein „Rollforward to end of logs“ ausgeführt. Danach ist die Datenbank geöffnet und steht mit dem Originalnamen (EP1) für die Produktion zur Verfügung. Die DBShadow® Prozesse Archiver und Recover werden nach der Notumschaltung gestoppt und sind nicht mehr aktiv.

   Es traten **keine** Probleme und Fehler auf.

   Bisher wurde der Offline Backup nicht benötigt da die aufgetretene Fehlersituation auf dem Echtssystem mit DBShadow®- und DB2-Mitteln behoben werden konnte.

   Es gibt allerdings auch Fehlersituationen in denen man nicht umhin kommt, auf den (hoffentlich vorhandenen) Offline Backup zurückzugreifen. Das könnte zum Beispiel ein ungewolltes Löschen einer Tabelle oder ein falscher „Point-in-Time“ Recovery sein.

7. Stoppen der db2 Instance auf dem Spiegelsystem
8. Löschen der durch das Öffnen der Spiegeldatenbank veränderten oder neu geschriebenen Online-Logfiles (evtl. anhand des Timestamps). Dies ist notwendig, da die DB2-Datenbank z. T. in die verarbeiteten Logfiles erweitert und die Transaktionsnummern nicht mehr zu den Archivfiles der Echtdatenbank passen.
9. Einspielen des Offline Backups (siehe 3.) auf dem Spiegelsystem
10. Starten der db2 Instance auf dem Echtsystem
11. Starten der db2 Instance auf dem Spiegelsystem – danach befindet sich die Datenbank wie erwartet im „Rollforward pending = Database“ Zustand und ist wieder als EP1S katalogisiert.
12. Zurücksetzen des „Copy _back“ parameters auf FALSE über Steuerung/Direkte Steuerung/Spiegeldatenbank auf dem DBShadow® Leitstand
13. Starten der DBShadow® Prozesses Recover

14. Ermitteln des nächsten benötigten Logfiles für die Spiegeldatenbank und Abgleich mit der Echtdatenbank. Sollten diese Logfiles auf dem Echtsystem nicht mehr vorhanden sein, so müssen diese aus einem früheren Backup zurückgespielt werden! Der Test geht natürlich davon aus, dass ein funktionierendes Backup & Recovery Konzept implementiert ist!
15. Eintragen der Nummer des nächsten benötigten Logfiles über das Leitstand-Menue DBShadow ✍ Steuerung ✍ Direkte Steuerung ✍ Echtdatenbank.
16. Starten der DBShadow® Prozesses Archiver
17. Der Ausgangszustand wurde wieder erreicht!

Es traten **keine** Probleme und Fehler auf. Die technische Funktionsfähigkeit ist damit validiert und wird bestätigt.

Andreas Schulz
Support Manager
Libelle Informatik GmbH

Jörg Peinelt
Partner Technical Sales Support
IBM Software Group

Anhang

Skript zum Erzeugen der Datenbank

```
#!/usr/bin/ksh
#--------------------------------------------------------------------------#
# mkdb2 :                                                                  #
# ----------                                                               #
#                                                                          #
#   Create DB2 database .                                                  #
#                                                                          #
#                                                                          #
#                                                                          #
#--------------------------------------------------------------------------#



#-------------------------------------------------------------------
# create database
#-------------------------------------------------------------------
set -x
SID=EP1
DB2H=/d1/db2/${SID}

db2 stop dbm force
db2 start dbm
db2  drop database $SID
db2  create database $SID  \
     on $DB2H
db2  <<HERE
    connect to $SID

   create tablespace PSAPBTABD pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata5/PSAPBTABD.container000' 4000  )

   create tablespace PSAPBTABI pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata2/PSAPBTABI.container000' 4000  )

   create tablespace PSAPCLUD pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata3/PSAPCLUD.container000' 4000 )

   create tablespace PSAPCLUI pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata3/PSAPCLUI.container000' 4000 )

   create tablespace PSAPDDICD pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata2/PSAPDDICD.container000' 4000 )

   create tablespace PSAPDDICI pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata2/PSAPDDICI.container000' 4000 )

   create tablespace PSAPDOCUD pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata3/PSAPDOCUD.container000' 4000 )

   create tablespace PSAPDOCUI pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata1/PSAPDOCUI.container000' 4000 )

   create tablespace PSAPEL46CD pagesize 4k   \
     managed by database     \
     using (  file '${DB2H}/sapdata4/PSAPEL46CD.container000' 4000 )

   create tablespace PSAPEL46CI pagesize 4k   \
```

```
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPEL46CI.container000' 4000 )

create tablespace PSAPES46CD pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata1/PSAPES46CD.container000' 4000 )

create tablespace PSAPES46CI pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata4/PSAPES46CI.container000' 4000 )


create tablespace PSAPLOADD pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPLOADD.container000' 4000 )

create tablespace PSAPLOADI pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPLOADI.container000' 4000 )


create tablespace PSAPPOOLD pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata4/PSAPPOOLD.container000' 4000 )


create tablespace PSAPPOOLI pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata1/PSAPPOOLI.container000' 4000 )


create tablespace PSAPPROTD pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPPROTD.container000' 4000 )


create tablespace PSAPPROTI pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPPROTI.container000' 4000 )


create tablespace PSAPSOURCED pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPSOURCED.container000' 4000 )


create tablespace PSAPSOURCEI pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata3/PSAPSOURCEI.container000' 4000 )


create tablespace PSAPSTABD pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata6/PSAPSTABD.container000' 4000 )


create tablespace PSAPSTABI pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata3/PSAPSTABI.container000' 4000 )


create tablespace PSAPUSER1D pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPUSER1D.container000' 4000 )


create tablespace PSAPUSER1I pagesize 4k   \
  managed by database     \
  using (  file '${DB2H}/sapdata2/PSAPUSER1I.container000' 4000 )
```

```
   create tablespace PSAPTEMP   \
     managed by system       \
     using (     '${DB2H}/sapdatat/TEMP4'           )

   quit

HERE

#-----------------------------------------------------------------
# set logretain
#-----------------------------------------------------------------

db2  connect to EP1

db2  update db cfg for EP1  \
     using  logretain recovery

#     using  logretain on

#-----------------------------------------------------------------
# dummy backup of the database
# (because of backup pending)
#-----------------------------------------------------------------

db2  force application all
sleep 10
db2  <<HERE

     backup database EP1        \
            to '/d1/db2/EP1/backup/save'
            #to '{DB2H}/backup/save'
HERE
#-----------------------------------------------------------------
# set userexit
#-----------------------------------------------------------------

db2  connect to EP1

db2  update db cfg for EP1  \
     using  userexit on
db2  update  db cfg for EP1 \
     using newlogpath '/d1/db2/EP1/log_dir'
#     using newlogpath '${DB2H}/log_dir'

db2 stop dbm force
sleep 10
db2 start dbm

#-------------------- end mkdbEP1 ----------------------------------
```