# Gespiegelte Datenbanken auch ohne Data Guard & Co.

**Kai Combüchen**
**BTN Versandhandel GmbH**
**Meine**

**Schlüsselworte**
ITIL, ITSCM, Disaster Recovery, Hochverfügbarkeit, Datenbankspiegelung, Standby, Oracle Standard Edition, Oracle Standard Edition One, Open Source

**Einleitung**
Die Kosten und Folgen eines Ausfalls der IT sind meist verheerend: Umsatzverluste, Verluste von Marktanteilen, Schadenersatzanforderungen, Image-Schäden bis hin zur Existenzgefährdungen sind keine Seltenheit. Zu ihrer Abwendung sind präventive Maßnahmen zu treffen.

Im Folgenden wird eine Lösung für die Hochverfügbarkeit von Oracle-Datenbanken vorgestellt, die bei einem mittelständischen Versandhandelsunternehmen realisiert worden ist. Zwar existieren für diesen Bereich kommerzielle Produkte wie Oracle Data Guard, Dbvisit standby, StandbyONE oder Libelle DBShadow, jedoch keine lizenzfreie, quelloffene und von der Community gestützte Lösung. Zuhörer und Interessierte sind daher aufgerufen, an der Weiterentwicklung und Verbreitung dieser Implementierung mitzuwirken.

**Service Continuity**
Wenn wir über die Hochverfügbarkeit von Datenbanken sprechen, dann ist dies in den allgemeineren Kontext des *Service Continuity* einzuordnen. In der IT Infrastructure Library (ITIL), dem De-facto-Standard für IT-Service-Management, ist hierzu der Prozess des *IT Service Continuity Managements* (ITSCM) definiert[1]: „Continuity Management is the process by which plans are put in place and managed to ensure that IT Services can recover and continue should a serious incident occur. It is not just about reactive measures, but also about proactive measures – reducing the risk of a disaster in the first instance."

Das ITSCM sieht dabei folgende Optionen für ein Recovery vor[2]:

- Cold Standby: Das Unternehmen kann ohne den ausgefallenen IT-Service 72 Stunden oder länger agieren.
- Warm Standby: Die zu tolerierende Ausfalldauer liegt im Bereich von 24 bis 72 Stunden.
- Hot Standby: Der ausgefalle IT-Service erfordert eine sofortige Wiederherstellung im Bereich von bis zu maximal 24 Stunden.

Während Cold und Warm Standby mittels Ausweich-Rechenzentren, Ersatz-Hardware oder Restore/Recovery-Verfahren realisiert werden können, erfordert das Hot Standby andere Lösungswege. Hier kommt die Spiegelung kritischer Systeme und Daten an geografisch getrennten Orten ins Spiel.

In einer vom Marktforschungsunternehmen Techconsult im Auftrag von HP Deutschland im Jahr 2013 durchgeführten Studie[3] hat die Hälfte der befragten Unternehmen mit 200 bis 500 Mitarbeitern angegeben, dass eine Stunde Ausfall der IT-Services zwischen 5.000 und 50.000 EUR kostet. Im

1 http://www.itil-itsm-world.com/itil-8.htm
2 http://www.itilnews.com/index.php?pagename=it_service_continuity_management
3 http://www.storage-insider.de/themenbereiche/management/compliance/articles/407377

Schnitt entstehen den Unternehmen durch Ausfälle Kosten von 240.000 EUR im Jahr. Dieses Volumen errechnet sich aus durchschnittlich 3 Ausfällen/Jahr x 3,9 Stunden x 20.278 EUR/Stunde. Als maximal zu verkraftende Ausfallzeit werden im Mittel 5 Stunden angegeben.

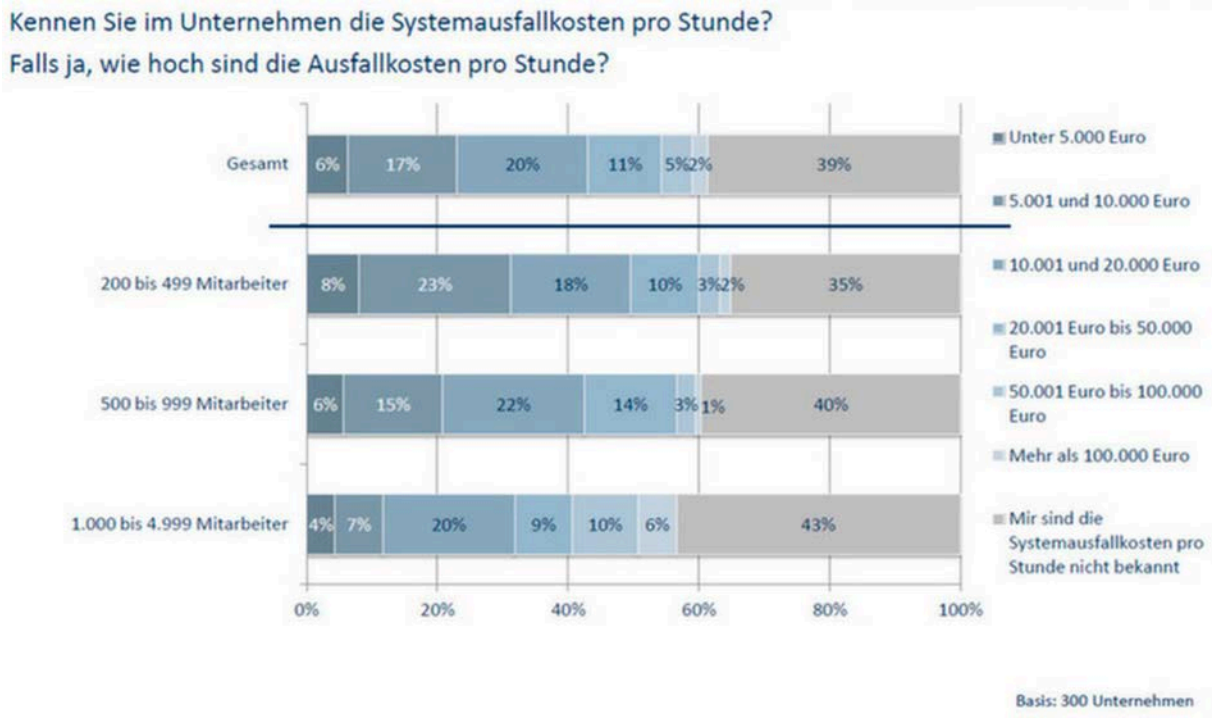


**Abb. 1: Kosten eines Systemausfalls (Quelle: Techconsult/HP)**

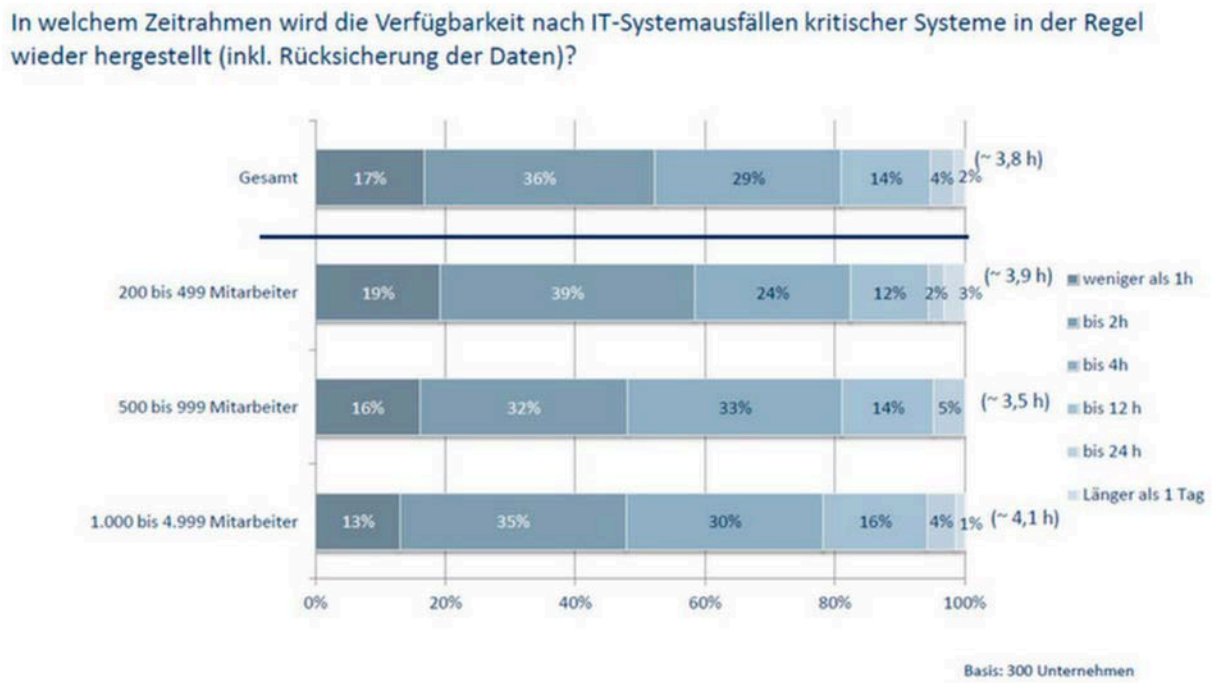


**Abb. 2: Zeitrahmen bis zur Wiederherstellung ausgefallener IT-Services (Quelle: Techconsult/HP)**

Der Studie zur Folge wissen dabei gut 40% der IT-Verantwortlichen gar nicht, welcher finanzielle Schaden ihnen durch diese Ausfälle entstanden ist, obwohl ¾ der befragten Unternehmen von Ausfällen geschäftskritischer Systeme für Warenwirtschaft, Fertigung oder Vertrieb betroffen waren.

Der Studie „Avoidable Cost of Downtime“ der CA Technologies aus dem Jahr 2011[4] ist zu entnehmen, dass deutsche Unternehmen aufgrund von Ausfällen ihrer IT-Services allein im Jahr 2010 mehr als 4 Milliarden Euro Umsatz einbüßten. Europaweit belief sich die Summe der ausgefallenen Arbeitsstunden auf über 37 Mio., die mittlere Ausfallzeit lag bei 14 Stunden. Ein großer Teil davon ließe sich durch kürzere Wiederherstellungszeiten vermeiden.

| | |
|---|---|
| Europe | 37,160,146 |
| France | 13,881,520 |
| Spain | 8,310,721 |
| UK | 5,495,623 |
| Germany | 4,770,937 |
| Italy | 4,189,956 |
| Sweden | 1,273,375 |
| Finland | 899,780 |
| Denmark | 851,398 |
| Netherlands | 699,660 |
| Belgium | 692,511 |
| Norway | 629,806 |

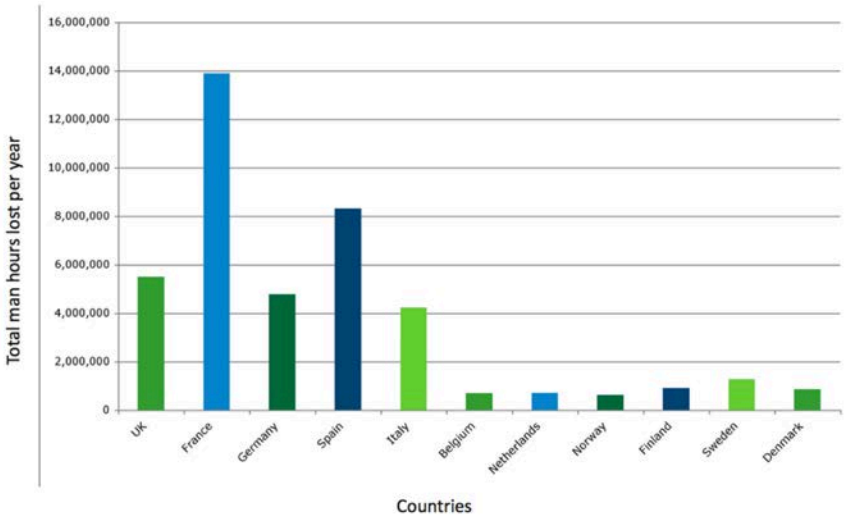


**Abb. 3: Im Jahr 2010 ausgefallene Arbeitsstunden in Folge ausgefallener IT-Services (Quelle: ca technologies)**

Diese Zahlen verdeutlichen, wie entscheidend die Verfügbarkeit auch für kleine und mittelständische Betriebe (KMU) ist, die gerade in Deutschland in hohem Maße vertreten sind. Da Inhaber und Geschäftsführer derartiger Unternehmen von Investitionen für mögliche Ausfälle, die hoffentlich nie eintreten, jedoch meist schwer zu überzeugen sind, sind kostengünstige, aber dennoch effektive Lösungen gefragt.

Die Ursachen für den Ausfall von IT-Services lassen sich basierend auf den im BSI-Grundschutzkatalog aufgeführten Gefährdungen in 6 Kategorien einteilen[5]:

- Elementare Gefährdungen (Feuer, Wasser, Naturkatastrophen)
- Höhere Gewalt
- Organisatorische Mängel
- Menschliche Fehlhandlungen
- Technisches Versagen
- Vorsätzliche Handlungen

Wenn es um die Wiederherstellung eines ausgefallenen IT-Services geht, spielen zwei weitere Kennzahlen eine wichtige Rolle: RTO und RPO. RTO (Recovery Time Objective) ist die Dauer, die nach einem Ausfall eines Services vergeht, ehe das Unternehmen wieder Zugriff darauf hat. RPO

4 http://www.ca.com/~/media/files/articles/avoidable_cost_of_downtime_part_2_ita.aspx

5 https://www.it-on.net/sites/default/files/news/3013_10_praesentation_ausfallzeit_in_der_it.pdf

(Recovery Point Objective) ist der erreichte Wiederanlauf-Zeitpunkt oder anders ausgedrückt die Dauer des tolerierten Datenverlustes.[6]

Beide Größen stehen in unmittelbarem Zusammenhang zueinander und haben erhebliche Auswirkungen auf die Kosten der damit verbundenen Umsetzung: Je kleiner RTO und RPO, desto höher die Kosten, vgl. Abb. 4. Hier gilt es also, im Unternehmen für den jeweiligen Service ein Optimum zu finden.

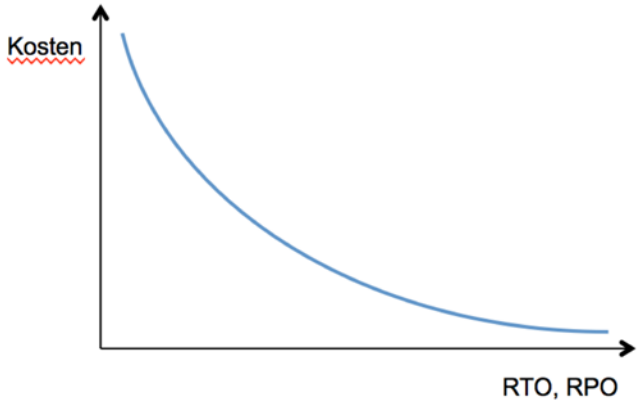


**Abb. 4: Recovery Time Objective (RTO) und Recovery Point Objective (RPO) im Verhältnis zu den Kosten**

**Kommerzielle Produkte**
Kommt das Gespräch auf hochverfügbare Oracle-Datenbanken, so fällt i. d. R. der erste Blick auf Oracle Active Data Guard. Allerdings handelt es sich hier um eine Option zur Enterprise Edition (EE) und ist damit in doppelter Hinsicht kostspielig: Zum einen ist die EE selbst um ein Vielfaches teuer als die Standard Edition (SE) und die Standard Edition One (SE One), zum anderen kommen weitere Lizenz- und Supportgebühren für Active Data Guard hinzu. Hier sind auch in einer kleinen Ausbaustufe schnell sechsstellige Beträge nur für die Oracle-Software fällig, für KMUs scheidet diese Variante daher meist aus.

Drittanbieter adressieren dieses Segment und bieten Produkte an, die sich auch mit der SE und der SE One kombinieren lassen. Zu nennen sind hier u. a. DBshadow der in Stuttgart ansässigen Libelle AG, Dbvisit Standby des gleichnamigen Anbieters aus Neuseeland und StandbyOne der up to data Professional Services GmbH aus Rheinland-Pfalz.

**Physical vs. Logical Standby**
Bei der Erstellung und dem Betrieb einer Standby-Datenbank gibt es zwei Prinzipien, die sich grundsätzlich voneinander unterscheiden: die physische und die logische Replikation.

Eine Physical Standby Database stellt – wie der Name schon ausdrückt – eine physische 1:1-Kopie der primären Datenbank dar. Sie befindet sich permanent im Recovery-Modus und wird durch das Applizieren der Änderungsvektoren aus den Redo-Log-Einträgen der Primär-Datenbank synchron gehalten („Redo Apply“). Im Failover-Fall kann eine Physical Standby Database schnell für die Nutzung zur Verfügung gestellt werden. Mit dem Übernehmen der letzten Änderungen der Primär-Datenbank sind sowohl RTO als auch RPO in der Regel klein. Im Normalbetrieb ist die Standby-Datenbank jedoch i. d. R. nicht für die Anwender verfügbar, bspw. für Read-only-Abfragen.

---

6 http://www.itwissen.info

Bei der Logical Standby Database erfolgt die Synchronisation über den SQL-Layer („SQL Apply“). Die Datenbank selbst kann eine gänzlich andere physikalische Struktur haben als die Primär-Datenbank. Die Datenbank ist im Normalbetrieb für Read-only-Operationen geöffnet und kann damit z. B. für Reporting-Zwecke genutzt werden. Sie erfordert allerdings einen höheren administrativen Aufwand, ferner ist nicht sichergestellt, dass sämtliche Änderungen auf der primären Seite auch automatisch auf der sekundären Seite erfolgen.

Für Disaster-Recovery-Szenarien stellt die Physical Standby Database die geeignetere Variante dar.

**O$^2$DM – Open Oracle Database Mirror**

Als das Thema IT Service Continuity vor einigen Jahren bei der BTN Versandhandel GmbH aufgegriffen wurde, ging es vor allem darum, die immer geschäftskritischer gewordene ERP-Anwendung, aber auch das operative Data Warehouse gegen Ausfälle zu sichern. Die Systemlandschaft im Kurzüberblick:

- ERP-Anwendung: Individualsoftware, Technologie-Stack bestehend aus JSF für das Web-Frontend (PrimeFaces), JEE für den Middle-Tier (TomEE), MyBatis als Persistence Layer und Oracle SE One zur Datenhaltung
- Data Warehouse: Oracle Discoverer Web (Ablösung für 2015 geplant), Advanced Replication und Oracle SE One
- Windows Server 2012
- ca. 100 Endanwender an drei Standorten
- Office-Zeiten: montags bis samstags von 07:00 bis 20:00 Uhr
- darüber hinaus umfangreiche Nacht- und Wochenend-Jobs, so dass sich de-facto ein 7x24-Betrieb ergibt

Die Stakeholder – in diesem Fall die geschäftsführenden Inhaber – waren zwar von der Funktionalität des Active Data Guard angetan, lehnten eine Einführung aufgrund der damit verbunden Lizenz- und Support-Gebühren jedoch ab. Stattdessen entschied man sich für eine Inhouse-Entwicklung mit folgendem Anforderungsprofil:

- Duplizierung der ERP- und der Data-Warehouse-Anwendung inkl. der zugehörigen Datenbanksysteme auf einem geografisch getrennten Server
- einfache, kostengünstige, wartungsarme, aber dennoch effiziente Implementierung
- RTO < 1 Stunde
- RPO < 10 Minuten
- Weiternutzung der SE-One-Lizenzen
- manuelles Failover durch DBAs
- keine operative Nutzung der Standby-Datenbanken im Normalbetrieb
- unveränderliche Weiternutzung mit künftigen DBMS-Versionen
- mögliche Nutzung unter Linux ohne größere Anpassungen
- keine Berücksichtigung von Real Application Cluster (RAC), Automatic Storage Management (ASM) und Oracle Managed Files (OMF)

Mit Blick auf das im Hause vorhandene Know-how entschied man sich für eine Implementierung basierend auf RMAN und Java. Das Grundprinzip, das sich auch in den o. g. kommerziellen Produkten in ähnlicher Weise wiederfindet, ist einfach und schnell beschrieben: Die auf der Primär-Datenbank anfallenden Archive Logs werden auf den Standby-Server transportiert und dort auf einem zuvor erstellten Duplikat der Primär-Datenbank appliziert. Der Normalbetrieb – hier am Beispiel der ERP-Anwendung, dasselbe gilt für die Data-Warehouse-Lösung – ist in Abb. 5 dargestellt.

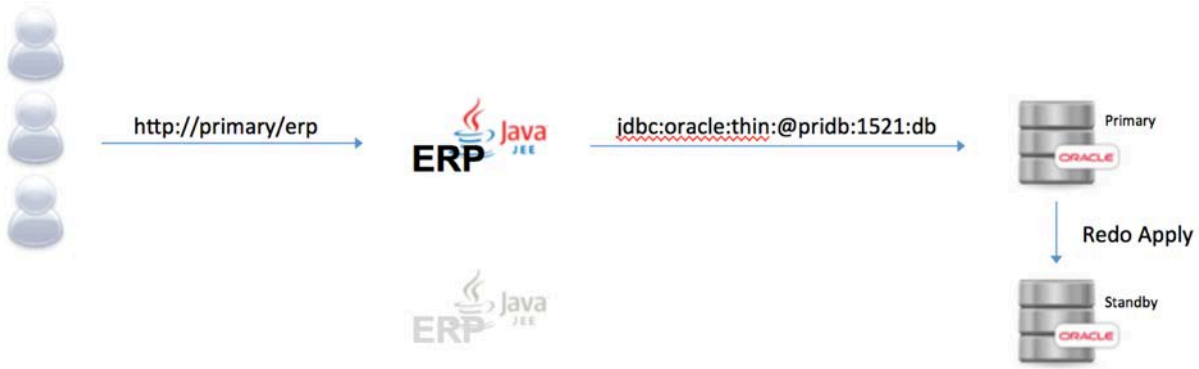


**Abb. 5: Normalbetrieb**

Ist die ERP-Anwendung gestört, verbinden sich die Anwender über die Angabe einer alternativen URL mit dem Standby-Server, auf dem dieselbe JEE-Anwendung installiert ist, vgl. Abb. 6.


**Abb. 6: Ausfall der ERP-Anwendung**

Fällt die Datenbank selbst aus, wird das Duplikat auf dem Standby-Server geöffnet und in der JEE Data Source die JDBC-URL angepasst, vgl. Abb. 7.

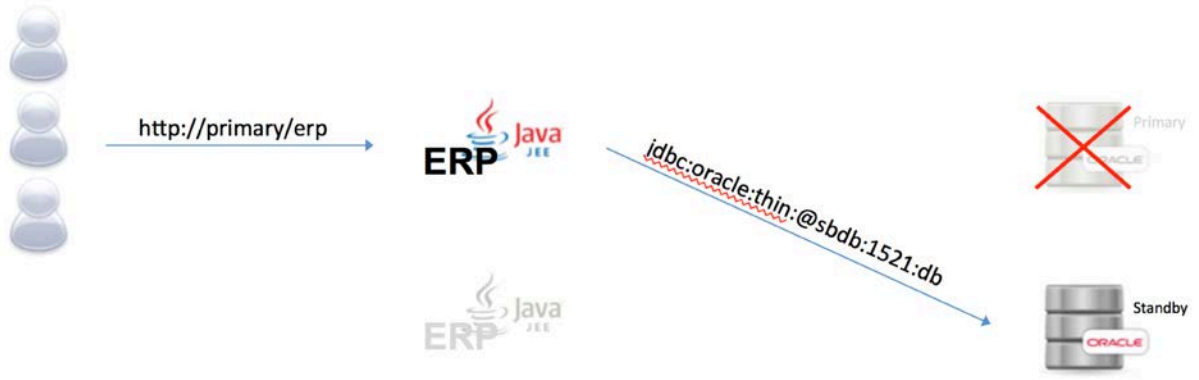


**Abb. 7: Ausfall des Datenbanksystems**

Um die Standby-Umgebung aufzubauen, wird auf der Standby Site zunächst die erforderliche Verzeichnisstruktur und ein Password File erstellt. Aus dem pfile der Primär-Datenbank wird ein spfile für die Standby Site erstellt, mit dem die Standby-Instance im Nomount-Status hochgefahren wird:

```
create spfile from pfile='...';
rman target sys/<password> nocatalog;
set dbid <dbid der Primär-Datenbank>;
startup nomount;
```

Aus einem vollständigen Backup der Primär-Datenbank wird auf der Standby Site mittels RMAN ein Duplikat aufgebaut:

```
restore controlfile from autobackup;
alter database mount;
restore database;
```

Dabei können die Data Files durchaus auf anderen Disks oder in anderen Verzeichnissen liegen, ein entsprechendes Mapping erfolgt mittels `alter database rename datafile`. Nach dem Restore bleibt die Datenbank im Mount-Status, so dass mit dem permanenten Recovery begonnen werden kann.

Wie zu sehen ist, wird bei diesen Schritten auf die Verwendung von Kommandos wie `backup conrolfile for standby`, `duplicate target database for standby`, `create controlfile for standby` oder `alter database mount standby database` verzichtet. Zumindest einer der o. g. Hersteller tut dies nicht, was nach Auffassung des Autors jedoch eine Lizenzverletzung darstellt, da hier auf Funktionen der Data-Guard-Option zurückgegriffen wird, die der Oracle Licensing Information zur Folge ausschließlich in der EE genutzt werden dürfen[7].

Anders als bei den zuvor genannten kommerziellen Lösungen wurde der Einfachheit halber der Prozess zur laufenden Spiegelung der Datenbank-Änderungen auf dem Standby-Server implementiert. In der Primär-Datenbank wird lediglich ein Schema `oodm` mit einer einzigen Tabelle `STANDBY_ARCHIVED_LOG` erstellt, die die auf der Standby Site bereits applizierten Archive Logs vorhält (Abb. 8).

| SAL_RECID | SAL_ARCHIVELOG_FILE | SAL_SEQUENCE# | SAL_FIRST_CHANGE# | SAL_FIRST_TIME | SAL_NEXT_CHANGE# |
|---|---|---|---|---|---|
| 275675 | G:\FLASH_RECOVERY_AREA\BTNP1\ARCHIVELOG\2014_09_29\01_MF_1_214295_B2LM9OKG_.ARC | 214295 | 3613948605261 | 2014-09-29 00:00:00 | 3613948606452 |
| 275676 | G:\FLASH_RECOVERY_AREA\BTNP1\ARCHIVELOG\2014_09_29\01_MF_1_214296_B2LMYKMO_.ARC | 214296 | 3613948606453 | 2014-09-29 00:00:00 | 3613948617497 |
| 275677 | G:\FLASH_RECOVERY_AREA\BTNP1\ARCHIVELOG\2014_09_29\01_MF_1_214297_B2LNMM0_.ARC | 214297 | 3613948617498 | 2014-09-29 00:00:00 | 3613948621015 |
| 275678 | G:\FLASH_RECOVERY_AREA\BTNP1\ARCHIVELOG\2014_09_29\01_MF_1_214298_B2LOB0C4_.ARC | 214298 | 3613948621016 | 2014-09-29 00:00:00 | 3613948624077 |
| 275679 | G:\FLASH_RECOVERY_AREA\BTNP1\ARCHIVELOG\2014_09_29\01_MF_1_214299_B2LOZ802_.ARC | 214299 | 3613948624078 | 2014-09-29 00:00:00 | 3613948626876 |

**Abb. 8: Tabelle STANDBY_ARCHIVED_LOG**

Die zentrale Java-Klasse besteht aus einer `main()`-Methode, die als Windows-Service gestartet wird. In einer Endlos-Loop werden folgende Schritte durchlaufen:

- Öffnen der Verbindung zur Primär-Datenbank
- Auslösen eines `ALTER SYSTEM SWITCH LOGFILE`, damit auf der Primär-Datenbank ein Wechsel der Redo-Log-Gruppe erfolgt und der Archiver-Prozess ein neues Archive Log erstellt.

[7] http://docs.oracle.com/cd/B28359_01/license.111/b28287.pdf

- Abgleich der zum Spiegelungs-Service gehörenden Tabelle `STANDBY_ARCHIVED_LOG` mit der Performance-View `V$ARCHIVED_LOG`, um festzustellen, ob seit dem letzten Apply-Durchlauf neue Archive Logs vorliegen.
- Die neuen Archive-Log-Files werden mit Hilfe der `FileChannel`-Klasse in ein Staging-Verzeichnis auf dem Standby-Server kopiert.
- Alle so empfangenen Files werden zunächst über das RMAN-Command `CATALOG ARCHIVELOG` im Catalog des zur Standby-Datenbank gehörenden Control-Files registriert.
- Im nächsten Schritt wird ein Recovery bis zur jüngsten SCN vorgenommen, das RMAN-Command hierzu lautet `RECOVER DATABASE UNTIL SCN`.
- Abschließend werden die Archive-Log-Files über das Command `DELETE ARCHIVELOG` aus der Staging Area entfernt.
- Die zuvor genannte Tabelle wird um die applizierten Archive Logs ergänzt.
- Vor dem erneuten Durchlaufen der Loop wird das Java-Programm über die `wait()`-Methode für eine konfigurierbare Zeitspanne suspendiert. Bei BTN ist dieser Wert auf 10 Minuten gesetzt, um den gewünschten RPO zu erzielen.

Die RMAN-Befehle werden über die `Process`-Klasse als externe Betriebssystem-Commands aufgerufen:

```
public static void command(String command) {
  boolean err = false;
  try {
    Process process = new ProcessBuilder(command.split(" ")).start();
    BufferedReader results =
            new BufferedReader(new InputStreamReader(process.getInputStream()));
    String s;
    while ((s = results.readLine()) != null) {
      log.info("Result from external O/S program: " + s);
    }
    BufferedReader errors =
            new BufferedReader(new InputStreamReader(process.getErrorStream()));
    while ((s = errors.readLine()) != null) {
      log.error("Error from external O/S program: " + s);
      err = true;
    }
  } catch(Exception e) {
    throw new RuntimeException(e);
  }
  if (err) {
    throw new OSExecuteException("Errors executing " + command);
  }
}
```

Sämtliche veränderbaren Konfigurationseinstellungen sind in einer Java-Property-Datei abgelegt:

```
<properties>
  <entry key="runInterval">600000</entry>
  <entry key="waitAfterSwitchLogfile">10000</entry>
  <entry key="primary.hostname">priserver</entry>
  <entry key="primary.sid">pridb</entry>
  <entry key="primary.port">1521</entry>
  <entry key="primary.username">oodm</entry>
  <entry key="primary.password">... </entry>
  <entry key="standby.rman.username">sys</entry>
  <entry key="standby.rman.password">...</entry>
   ...
</properties>
```

Um die Ergebnisse der ausgeführten RMAN-Commands im Bedarfsfall überprüfen und eventuell aufgetretene Exceptions nachvollziehen zu können, gibt es zwei Logging-Mechanismen:

- Zum einen werden die Log-Ausgaben der extern aufgerufenen RMAN-Commands in die Dateien `catalog.log`, `recover.log` sowie `delete.log` geschrieben. Damit diese Dateien beim erneuten Durchlauf nicht überschrieben werden, werden die Vorgänger durch Anhängen eines Timestamps umbenannt.

```
Starting recover at 29-SEP-14
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=156 devtype=DISK

starting media recovery

archive log thread 1 sequence 214303 is already on disk as file ...
archive log filename=... thread=1 sequence=214303
media recovery complete, elapsed time: 00:00:42
Finished recover at 29-SEP-14

Recovery Manager complete.
```

- Zum anderen werden über try/catch-Blöcke gefangene Exceptions mittels des LOG4J-Frameworks in ein XML-File geschrieben, das entweder direkt eingesehen oder über ein XSL-Stylesheet und XSLT-Transformation in eine HTML-Darstellung überführt und so komfortabel im Browser angezeigt werden kann:

```
<entry time='27.09.14 14:26:24.265' priority='DEBUG' ndc='null'><thread>main</thread>
  <class>de.btn.dbmirror.DBMirror</class>
  <method>recoverDatabaseUntilSCN</method>
  <line>362</line>
  <message>Archived redo logs successfully recovered.</message>
</entry>
```

**Erfahrungen und Verbesserungsideen**

Die Implementierung von O$^2$DM hat nur gut eine Woche gedauert und ist seit mehreren Jahren produktiv und nahezu störungsfrei im Einsatz. Es gab bisher lediglich die Situation, dass die Archive-Log-Files der Primary Site durch zu frühes Löschen nach der RMAN-Sicherung auf der Standby Site nicht zur Verfügung standen und damit zu einem „Archive Log Gap“ führten. Hier half nur das erneute Aufsetzen der Standby-Datenbank.

Anfänglich wurde die Lösung mit 10gR2 genutzt, mittlerweile mit 11gR2. Für 2015 ist eine Migration auf 12c mit einer Single-Tenant-Architektur (anstelle einer non-CDB) geplant, Probleme sind auch hier jedoch nicht zu erwarten, da das Verfahren des Redo-Apply weiterhin angewendet werden kann.

**Der Gedanke der freien Software**

Wie eingangs erwähnt und dem gewählten Namen O$^2$DM/Open Oracle Database Mirror zu entnehmen ist, möchte der Autor die Lösung der Community zur Nutzung und Weiterentwicklung zu Verfügung stellen. Getreu dem Motto, dass es nichts gibt, was man nicht noch besser machen kann, könnte so mit den Ideen und dem Engagement weiterer Entwickler und DBAs ein lizenzfreies und quelloffenes Produkt entstehen und die Lücke im bisher ausschließlich kommerziell besetzten Segment schließen.

**Fazit**

Die hier vorgestellte Variante einer Physical Standby Database liefert trotz ihrer Einfachheit eine für das Disaster Recovery geeignete Lösung und dürfte vor allem für KMUs, die wenig finanzielle Mittel für hoch verfügbare IT-Services aufbringen möchten oder können, von Interesse sein. Sollte die Idee

von der Community aufgegriffen werden, könnte ein Produkt entstehen, das denen der kommerziellen Anbieter um nichts nachsteht, aber weiterhin quelloffen und lizenzfrei bleibt.

**Kontaktadresse:**
Kai Combüchen
Bereich IT
BTN Versandhandel GmbH
An der Waage 1
D-38527 Meine

Telefon: +49 (0) 5304-906 044
Fax: +49 (0) 5304-906 235
E-Mail kai.combuechen@btn-muenzen.de
Internet: www.btn-muenzen.de