

mit
DVD
database

DATEN-MANAGEMENT / ARCHITEKTUREN / BUSINESS / LÖSUNGEN

Januar/Februar 1/10

database pro

€ 14,95

Deutschland € 14,95 Österreich € 16,45 BeNeLux € 16,45 Schweiz sfr 29,90



Open Source Datenbanken

- Marktübersicht – offen für den Business-Einsatz ab S. 68
- db4o – objektorientiertes Datenmanagement ab S. 74
- NoSQL – die Zukunft für Daten im Web 2.0 ab S. 31

Test- und Demoverionen, Auswahl:

JAHRESARCHIV 2009

- Ausgaben 6/08 bis 6/09
- Über 900 Seiten Profiwissen
- Mit Index und schneller Suche

IBM DB2

- IBM DB2 Express-C 9.7
- Quest Toad for DB2 (Trial)
- DB2 Add-in for Visual Studio
- Redbook: DB2 Workload Manager

DATENBANKEN

- MySQL 5.1
- PostgreSQL 8.4.1.1
- Firebird 2.1.3
- MariaDB 5.1.39

ORACLE-IDES

- Toad DBA Suite for Oracle 10.1
- SQL Navigator 6.2.1
- PL/SQL Developer 8.0
- Hora 8.1.8.3
- Oracle SQL Developer 1.5.5

SPECIAL

- combit List & Label 15
- Quest Spotlight on SQL Server Enterpr. 6.0
- KeepTool 9

DT-Control
geprüft:
Nicht jugend-
beeinträchtigt

SQL Azure 1.0

So funktioniert Microsofts
Datenlösung in der Cloud S. 36

Governance

Daten und Informationen
steuern für den Erfolg S. 87

DB2 im Unternehmen

- Transaktionen und Workload Management S. 46
- Natives XML mit pureXML S. 59
- Hochverfügbarkeit mit DB2 pureScale S. 82

Windows 7 – stark mit 64 Bit

Umfassendes Basiswissen für den Betriebssystem-Wechsel S. 20

Tipps, Tests & Trends

Gupta-Datenbankumstieg mit sqlPorter und sqlTranslator S. 42
Dazu: Oracle-IDEs, MySQL unter Windows, XML-Schema,
Datenbankspiegelung, Lotus Notes Ansichten, SQLite-Workshop,
Geschichte der Datenbanken



Datenspiegelung – Ansatz für ein durchgängiges Backup

Aus eins mach zwei

Eine vollständige Datensicherung umfasst mehr als die Spiegelung der Daten in der Datenbank – ein durchgängiges Konzept berücksichtigt auch die Umgebungsdaten. **Lars Albrecht**

Auf einen Blick

Inhalt

Die haus eigenen Mittel der Datenbankhersteller taugen nur zum Spiegeln der in der Datenbank abgelegten Daten. Dateien, die außerhalb der Datenbank vorliegen, und No-logging-Operationen bemerkt das Spiegelsystem nicht. Eine besondere Herausforderung ist die Spiegelung von Multi-Datenbank-Umgebungen.

Plattform

Plattformübergreifend

Autor

Lars Albrecht ist Geschäftsführer der Libelle AG. Davor war er bei IBM Deutschland als IBM Certified Consultant und Business Development Manager beschäftigt.

Um eine Standby-Datenbank aufzubauen, wird eine Kopie der Originaldatenbank auf ein anderes System im lokalen Netz (LAN) oder in ein entferntes Netz (Wide Area Network, WAN) transferiert. Transaktionen der Echtdatenbank werden dann auf das Spiegelsystem übertragen und der Spiegeldatenbank zugeführt. Kommen die Recovery-Dateien zeitversetzt bei der Spiegeldatenbank an – zum Beispiel nach vier Stunden –, bietet diese Lösung einen optimalen Schutz vor logischen Fehlern. Wird beispielsweise um 14:00:00 Uhr die Echtdatenbank durch einen logischen Fehler zerstört, lässt sich die Spiegelseite auf 13:59:59 Uhr wiederherstellen und so die Datenbank mit gültigen Daten wieder nutzen (Bild 1).

Setzt man eine Standby-Datenbank mit den Hausmitteln der Oracle Enterprise Edition auf, sind die Komplexität und das erforderliche Know-how eine erste Hürde. Die größere Herausforderung aber ist, dass konzeptionell und funktionell wichtige Anforderungen nicht berücksichtigt werden. Diese müssen entweder manuell per Zusatzskript geregelt werden oder sie entfallen komplett. Wie sehen diese Anforderungen aus, und wie können sie gelöst werden?

Die Oracle-Hausmittel sind per Definition datenbankzentrisch und spiegeln deshalb keine

Dateien außerhalb der Datenbank. Mit zunehmender Verbreitung sogenannter „Service Oriented Architectures“ (SOA) belegen Datenbanken in Applikationsumgebungen zwar den größten Bereich, es gibt aber noch andere Teile. Beispiel SAP-Umgebungen: Hier werden viele Dateien außerhalb der Datenbank gehalten. So speichert ein SAP-Portal den gesamten Java-Stack in einfachen Dateien. Werden diese nicht mitgespiegelt, gehen nach dem Umschalten auf das Standby-System wichtige Unternehmensdaten verloren. In der Praxis werden diese Dateien typischerweise manuell per Skript kopiert. Es gibt aber auch Lösungen, die diese Dateien gleich mitspiegeln.

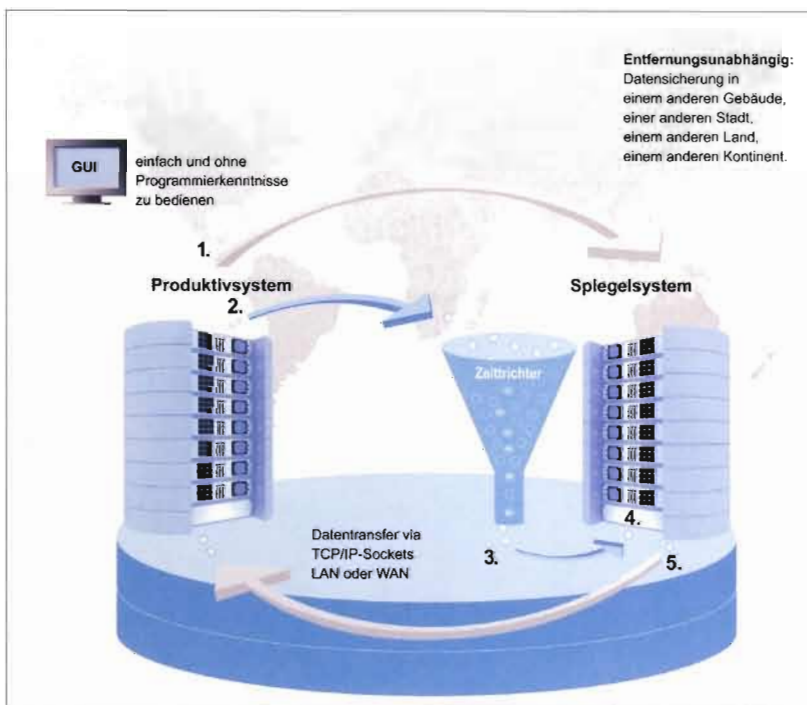
Spiegeln von Nologging-Operationen

Nologging-Operationen sind Datenbankänderungen, die direkt auf Datenfile-Ebene der Datenbank ohne Logdateien durchgeführt werden. Da eine Standby-Datenbank mit Logdateien arbeitet, werden diese Operationen nie auf die Spiegeldatenbank übertragen. Konzeptionell können Nologging-Operationen jedoch automatisch erkannt und auf das Spiegelsystem übertragen werden, sodass keine manuellen Nacharbeiten oder ein kompletter Neuaufbau des Spiegels erforderlich sind. Eine Umsetzung dieses Konzepts findet sich beispielsweise in den Spiegellösungen von Libelle [1]. Alternativ bleibt oft nur die Lösung, die gesamte Applikationsumgebung auf „forced logging“ zu setzen.

Ein konzeptioneller Punkt unabhängig vom verwendeten Spiegelungswerkzeug ist die Nutzung von „Secondary Archive Destinations“. Dabei werden Logdateien in einem primären und einem (optionalen) sekundären Verzeichnis gehalten. Beim Betrieb von Standby-Lösungen wird auf dieselben physikalischen Logdateien zugegriffen, die in der Regel bereits von einer Backup-Software verwaltet werden. Eine Trennung in ein primäres und ein sekundäres Verzeichnis bietet mehr Transparenz in Betrieb und Verwaltung. Listing 1 zeigt die entsprechenden Oracle-Befehle.

Ein einmaliges Definieren von Log Archive erfordert einen Neustart der Datenbank. Anschließend kann die Anzahl der Destinations ohne Neustart erweitert werden. Übertragene Datei-

Grundprinzip der Datensicherung durch Spiegelung auf einem anderen System (Bild 1)



Listing 1: Oracle-Befehle zur Spiegelung

```

SQL> -- Ist die Datenbank im Archive Mode?
SQL> archive log list

SQL> -- Welche Archive-Methode wird verwendet?
SQL> show parameter log_archive_dest

SQL> -- Wenn log_archive_dest_x aktiviert ist, kann zweite Destination so aktiviert werden:
SQL> alter system set log_archive_dest_2='LOCATION=<2nd destination> optional reopen=60' scope=both;
SQL> --
SQL> -- OPTIONAL bedeutet, die Log-Destination ist optional und vermeidet Archive Stuck.
SQL> -- MANDATORY bedeutet, die LOG-Destination ist zwingend und kann zu Archive Stuck führen.
SQL> -- Da der Default bei den LOG-Destinations OPTIONAL ist, ist zu überlegen, ob eine
SQL> -- LOG-Destination für Tapebackup nicht auf MANDATORY gesetzt werden muss.
SQL> -- REOPEN=60 Alle 60 Sekunden wird versucht, ob die Destination wieder beschreibbar ist.

SQL> -- Zum Deaktivieren der LOG-Destination: (erfordert Neustart der Datenbank)
SQL> alter system reset log_archive_dest_2 scope=spfile sid='*';

SQL> -- Wird noch die alte LOG_ARCHIVE_DEST eingesetzt, ist Folgendes zu tun:
SQL> alter system reset log_archive_dest scope=spfile sid='*';
SQL> alter system set log_archive_dest_1='LOCATION=<1. Pfad zu Log-Dateien> mandatory' scope=spfile;
SQL> alter system set log_archive_dest_2='LOCATION=<2. Pfad zu Log-Dateien> optional' scope=spfile;
SQL> -- Durchstarten der DB. Beim Einsatz des init-File muss dieses ebenfalls geändert werden.

```

en aus dem zweiten Verzeichnis lassen sich dann entweder per Skript oder durch Tools automatisiert löschen.

Der Standby-Betrieb ist vor allem in WAN-Spiegelungen gefährdet, wenn keine ausreichende Bandbreite zur Verfügung steht oder die Netzwerke instabil sind. Generelle Basisfunktionen in Oracle 11g enthalten zwar einen rudimentären Kompressionsalgorithmus, das reicht im Praxisbetrieb bei großen Datenmengen aber oft nicht aus. Zusätzlich zu einer erweiterten, parametrisierbaren Kompression empfiehlt es sich, das IP-Protokoll so zu erweitern, dass mit größeren IP-Paketen gearbeitet wird, die parallel übers Netz geschickt werden. Libelle hat sehr gute Erfahrungen in Projekten gemacht, wenn der Datentransfer über die Standard-Komprimierung hinaus optimiert wurde. Speziell dafür wurde ein Verfahren entwickelt, das einzelne Datenpakete in kleinere Pakete aufteilt und diese parallel an den Zielrechner schickt.

Multi-Datenbank-Umgebungen spiegeln

Eine weitere Herausforderung ist das Spiegeln von Umgebungen mit mehreren Datenbanken. Bezüglich der SOA von SAP-Applikationen bedeutet dies, dass ein und derselbe Geschäftsprozess über verschiedene SAP-Applikationen und damit in mehreren Datenbanken gehalten und verändert wird. Das Absichern mit Standby-Lösungen erfordert, konzeptionell und in der Umsetzung per Skript oder per Software-Werkzeug,

sogenannte Recovery Consistency Points [2] über das Dateisystem und gegebenenfalls über verschiedene Standby-Systeme zu verwalten.

Per Knopfdruck auf das Spiegelsystem umschalten

Als letzter Schritt im Standby-Betrieb müssen beim Umschalten der Datenbank die Nutzer auf das Spiegelsystem umgelenkt werden. SAP-Lösungen erwarten zum Beispiel, dass der Spiegel mit dem identischen Hostnamen startet, damit das SAP-System überhaupt aktiv werden kann. In einem Netzwerk wird idealerweise auch die gleiche IP-Adresse verwendet. Entsprechend erweiterte Spiegelungslösungen wie beispielsweise Libelle BusinessShadow beinhalten genau diese Funktionalität. Dadurch ist es möglich, nicht nur die Datenbank, sondern auch die gesamte Applikation auf Knopfdruck umzuschalten.

Das Verfahren der Standby-Datenbank deckt die kritischen Anforderungen an Datenbankspiegelungen in weiten Teilen ab. Die Oracle-Hausmittel sind aber nicht ausreichend, um die genannten Anforderungen komplett zu erfüllen. Um eine durchgängige Spiegelung zu erreichen, empfiehlt es sich deshalb – falls diese Bereiche nicht durch eine durchgehende automatisierte Spiegellösung abgedeckt werden –, die Lücken zumindest konzeptionell zu schließen. [bl]

[1] www.libelle.com
 [2] http://en.wikipedia.org/wiki/Recovery_Consistency_Objective