Oracle 10*g* – Lösungen für Hochverfügbarkeit

Sicherheit – Integrität – Performance – Verfügbarkeit

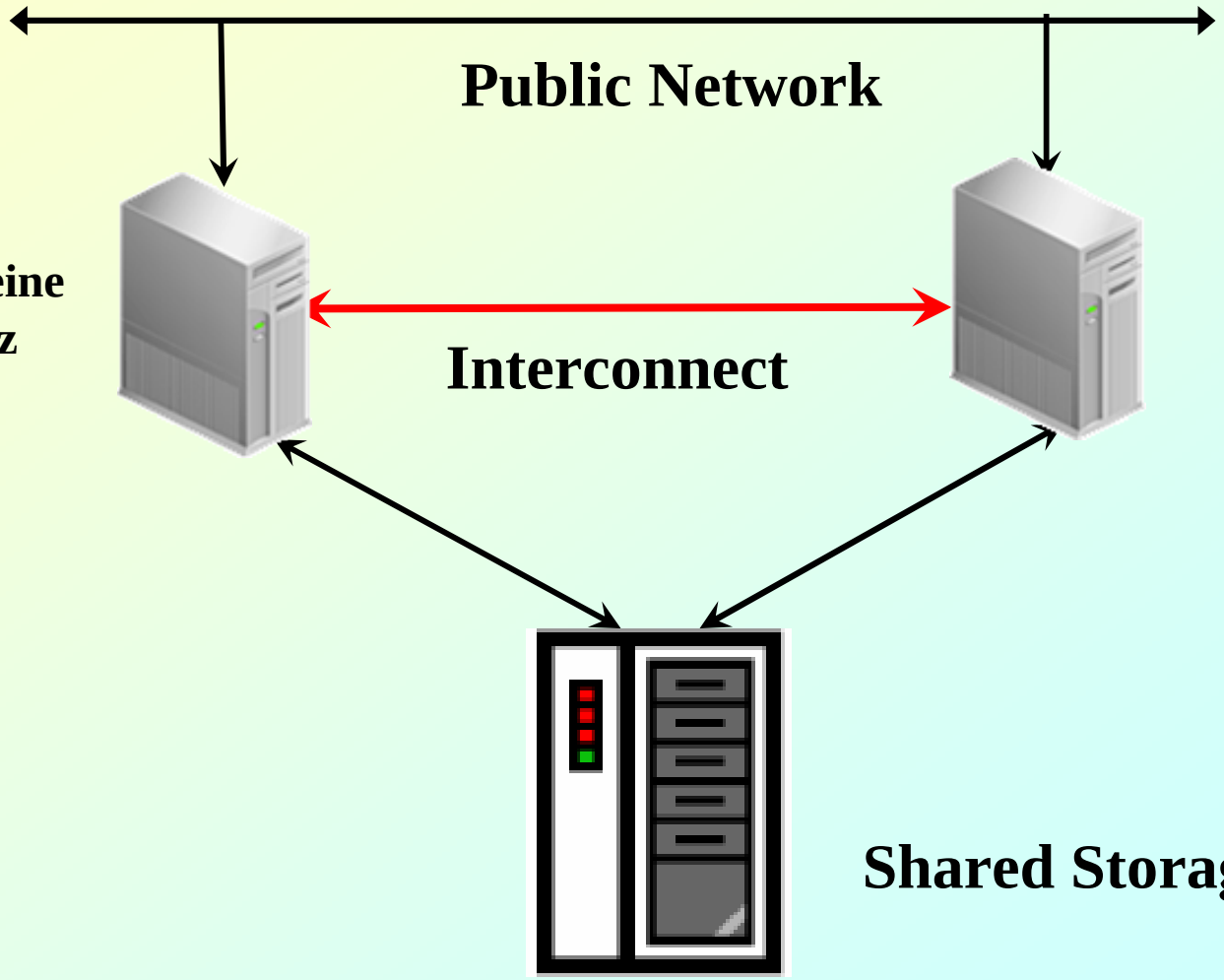
1. Datensicherheit à Daten müssen gegebenenfalls bis zum Zeitpunkt des Crash wiederhergestellt werden können.
2. Datenintegrität à Schutz vor logischen Fehlern und Datenkorruption
3. Hochverfügbarkeit à Minimierung der Ausfallzeit und des Aufwandes für eine eventuelle Wiederherstellung
4. Performance à Nutzeraktionen (Abfragen und DML sollen in einer bestimmeten Zeit ausgeführt werden (Antwortzeit und Durchsatz).

Für diese Forderungen gibt es Oracle Features, die verschiedene Teilbereiche abdecken. Dabei sind Lizenzierungsprobleme zu beachten.
DIE Lösung gibt es nicht.

Features für die Hochverfügbarkeit

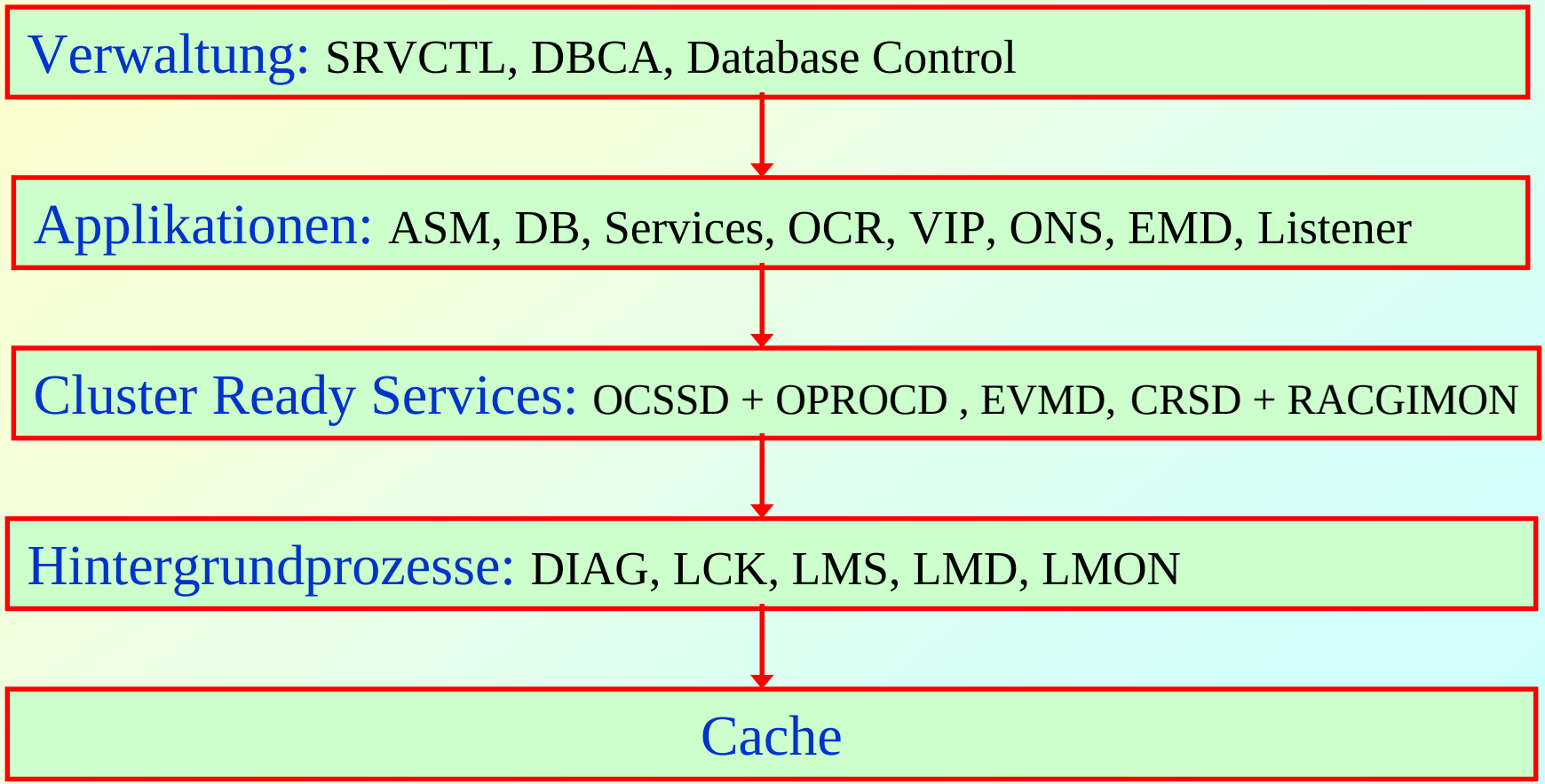
- Fast Start Recovery
- Resumable Space Allocation
- Stillegung der Datenbank (Quiesce Database)
- Online-Wartung
- Flashback
- Data Guard (Standby Database)
- Real Application Cluster (RAC)
- Real Application Clusters Guard (Cold Standby)
- Advanced Replication
- Streams
- Transparent Application Failover, Fast Start Application Notification, Services und Load Balancing

RAC – Prinzip



Auf jedem Knoten läuft eine Oracle-Instanz

Softwarearchitektur



Clusterware – Komponenten

- Ø CSS (Cluster Synchronistion Service)
  - Kommunikation zwischen den Prozessen im Cluster
  - Dynamische Informationen über Knoten und Instanzen
  - Verwaltung der Voting Disk und Überwachung des „heartbeat“
- Ø CRS (Cluster Ready Service)
  - Sichert die Verfügbarkeit der Applikationsressourcen
  - Verwaltet die Cluster Registry (OCR)
- Ø EVM (Event Manager): publiziert Ereignisse
- Ø ONS (Oracle Notification Service): dient dem FAN
- Ø OPROCD (Process Monitor Daemon)
- Ø RACG: dient dem FAN

Clusterware – Prozeßarchitektur

- DIAG (Diagnosability Daemon): Sammelt Diagnoseinformationen über die Oracle-Prozesse im RAC
- LCK (Lock-Prozeß): Verarbeitet Lock-Anfragen, die nicht mit der Cache Fusion zusammenhängen, z.B. Row Cache
- LMD (Lock Manager Daemon): Regelt clusterweite Sperren, die Deadlock Detection und die Lock Conversion
- LMS*n* (Lock Manager Process): Sendet Blöcke über den Interconnect zur Erfüllung von Cache Fusion Requests.
- LMON (Lock Monitor): Rekonfiguration von Sperren beim Hinzufügen und Entfernen von Instanzen, sogenanntes *Dynamic Ressource Remastering*

Clusterware – globale Ressourcen

- GRD (Global Resource Directory): Enthält Informationen über die gemeinsam genutzten Ressourcen des Global Cache, ermöglicht die Lokalisierung aktueller Block Images.
- GCS (Global Cache Service): Blocktransfer über den Interconnect, LMSn und LMD arbeiten für den GCS
- GES (Global Enqueue Service): Verwaltet die Non-Cache Fusion Ressourcen (z.B. Data Dictionary Locks)
- GSD (Global Services Daemon): Koordiniert die Interaktion mit Tools wie DBCA, EM, SRVCTL

Ressourcen Remastering

- Die Verwaltung der Ressourcen ist über die Knoten verteilt.
- Das dynamische Remastering ermöglicht, daß Ressourcen-Master geändert werden können, ohne daß eine vollständige Neukonfiguration erforderlich ist.
- Während der Verarbeitung verschiebt verzögertes dynamisches Remastering Ressourcen zur aktivsten Instance.
- Wenn eine Instanz die Gruppe verläßt, weisen die Hintergrundprozesse nur den Ressourcen der ausgeschiedenen Instanz neue Master zu.
- Analog werden, wenn eine Instanz neu zu einer Gruppe hinzukommt, die Ressourcen allmählich neuen Mastern zugewiesen, je nach Rechnerlast des Clusters.

Ressourcenmodi

- Ressourcen des Global Cache Service haben drei verschiedene Modi:
  - NULL (N)
  - Shared (S)
  - Exclusive (X)
- Der Global Cache Service bearbeitet Ressourcenanforderungen nach dem FIFO-Prinzip (First in, first out).
- Der Modus einer Ressource in einem Block kann sich unabhängig von Transaktionsstatus und Sperren auf Zeilenebene, die mit diesem Block verknüpft sind, ändern.

Ressourcenrollen

- Ressourcen verwenden Rollen, um Cache-Fusion zu unterstützen. Sie können in einer von zwei Rollen gehalten werden:
  - Lokal: Die mit der Ressource verknüpften Block-Images sind unabhängig von anderen Instanzen und vom Global Cache Service.
  - Global: Die von der Ressource belegten Blöcke sind in mehr als einer Instanz "dirty" und können nicht unabhängig manipuliert werden.
- Ein Past Image ist eine Kopie eines "dirty" Blocks, die an eine andere Instanz übergeben wurde.
- Ein Past Image wird aufrechterhalten, bis es oder ein neueres Image des Blocks auf die Platte geschrieben wurde.

Cache Fusion – Beispiel

Instanz RAC2
Master

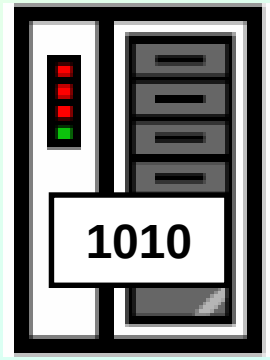
NL0

Instanz RAC2 hält den Master-Datensatz des Global Cache Service für den Block

Instanz RAC1

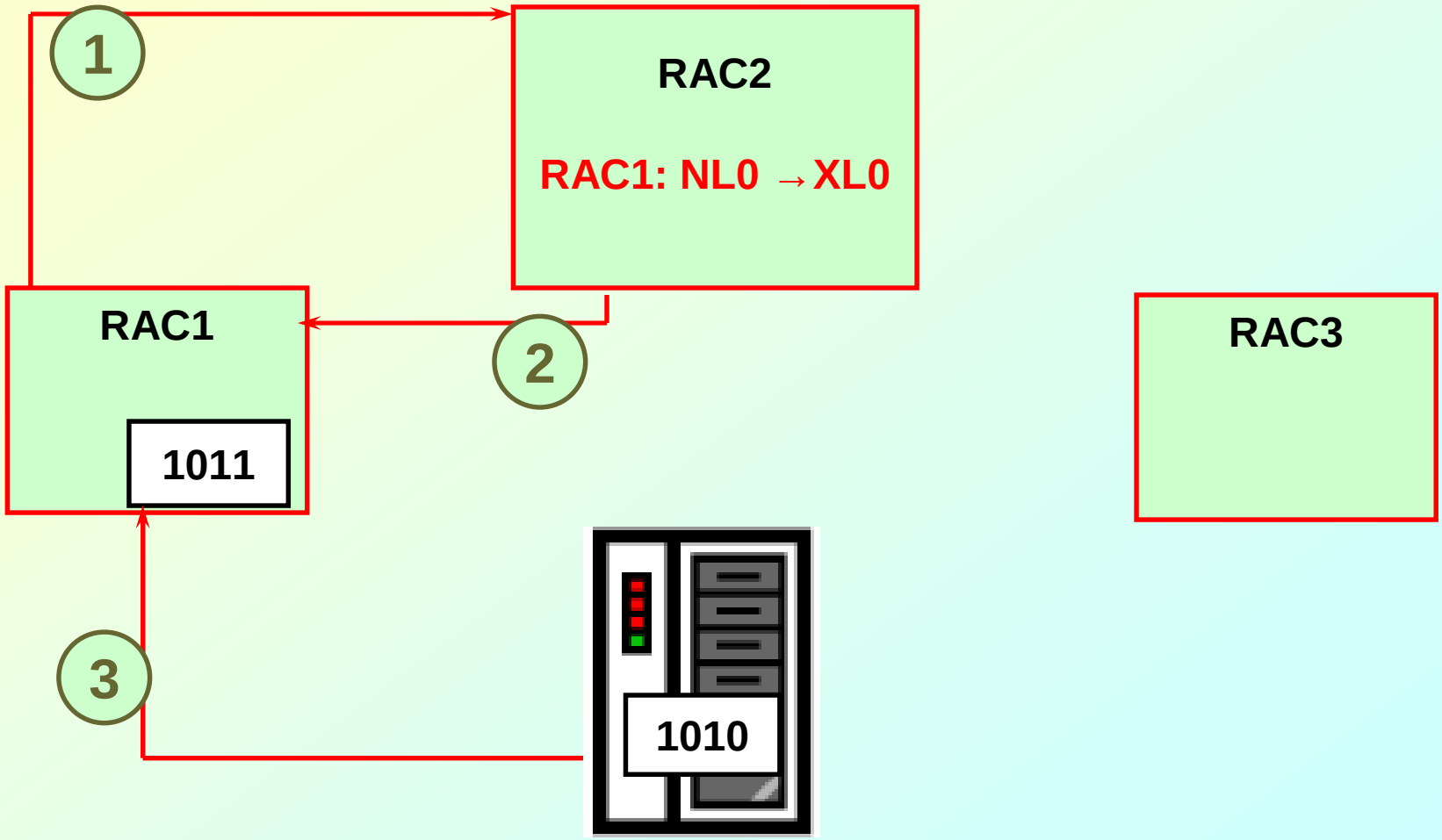
Instanz RAC3

Die Instanzen RAC1 und RAC3 verwenden den Block

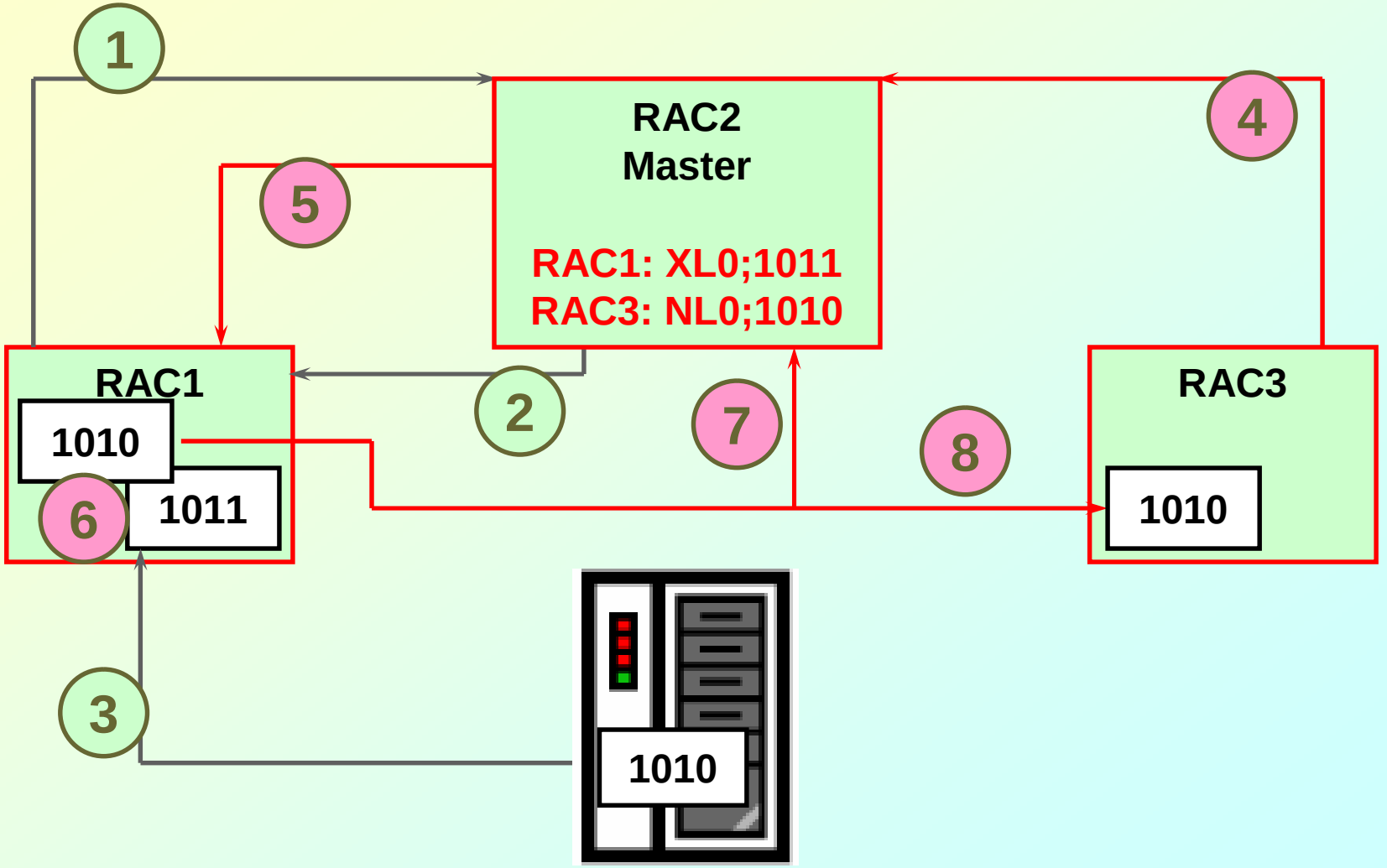


Block befindet zu Beginn auf der Platte mit der SCN 1010

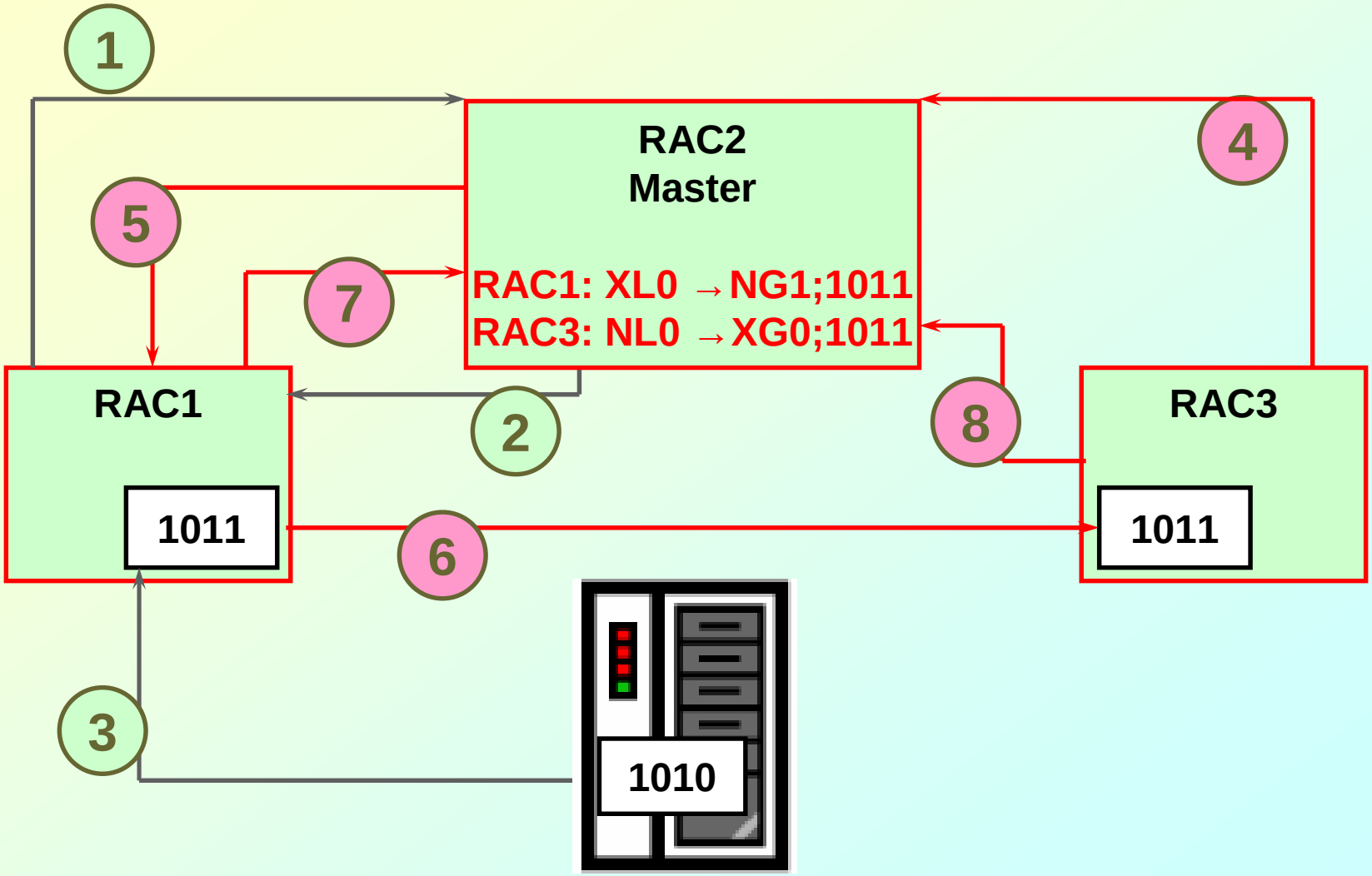
Cache Fusion – Lokal



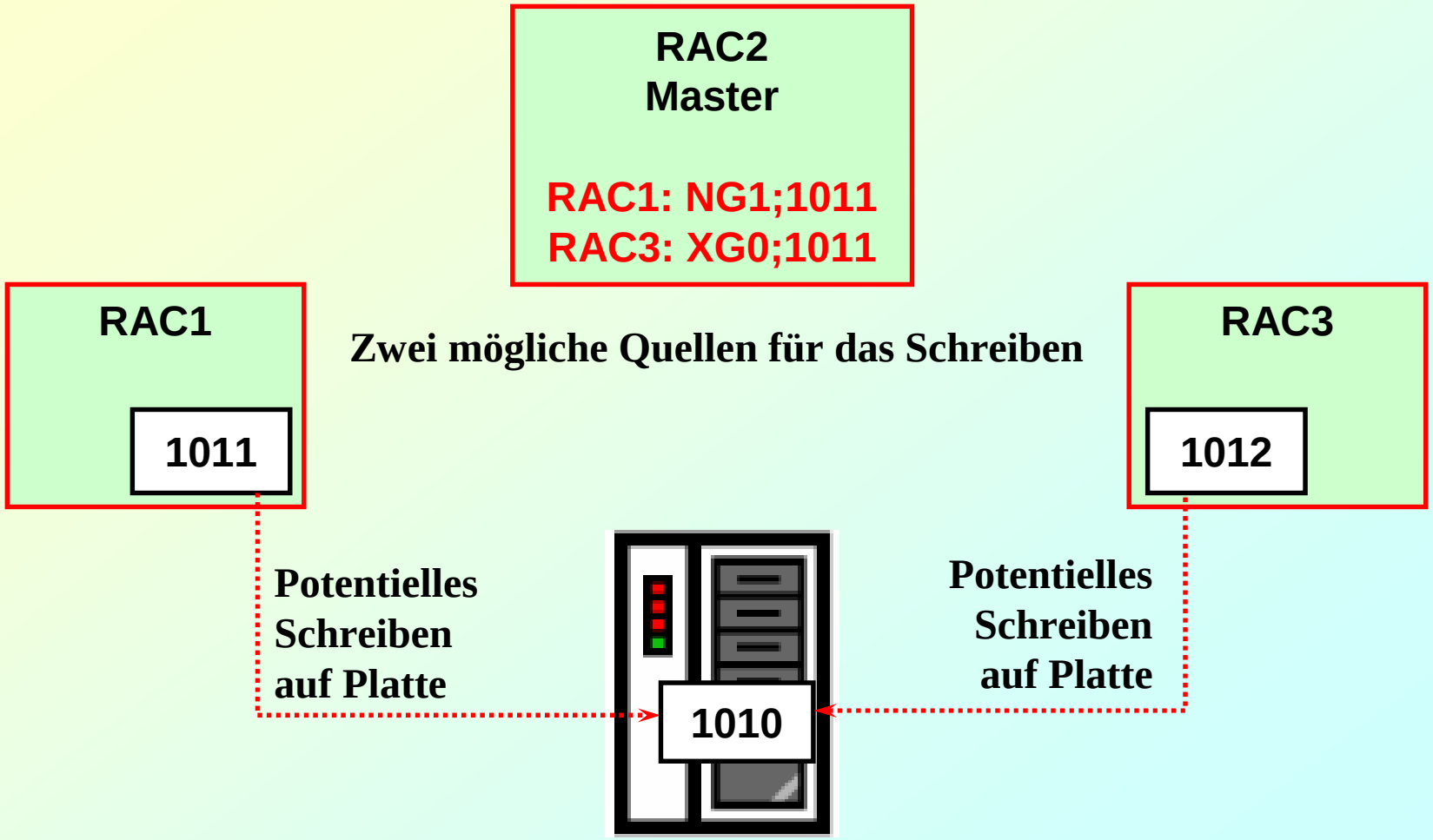
Cache Fusion – Lesekonsistenz



Cache Fusion – Blockübetragung



Cache Fusion – Schreiben



Speicherung der Software

- Oracle Binaries und Clusterware in eigene Homes speichern
- Oracle Home und CRS Home können jeweils in das Shared Storage oder lokal (für jede Instanz ein getrenntes Home) gespeichert werden
- Es muß sichergestellt werden, daß die Software keinen Single Point of Failure bildet (Patches, Upgrades)
- Lokal: Nur eine Instanz fällt gegebenenfalls aus.
- Shared Storage: Unbedingt redundante Homes anlegen.

Speicherung der Dateien

Dateien **müssen** in das Shared Storage

- Cluster Registry (OCR) und Voting Disk
- Online Redologs aller Instanzen
- Archivelogs aller Instanzen
- Dateien der Undo-Tablespaces aller Instanzen
- Daten- und Temp-Dateien der Datenbank
- Kontrolldateien
- Flash Recovery Area
- Change Tracking File des RMAN

Dateien **können** in das Shared Storage

- Parameter- und Paßwortdatei

Speicheroptionen – Vergleich

- Ø **ASM** (Advanced Storage Management)
- Keine Zusatzkosten, Verwaltung mit SQL
- Einfache Spiegelung
- Automatisches Rebalancing
- Kein Zugriff mit Filesystem-Befehlen
- Zusätzliche ASM-Instanz

- Ø **CFS** (Cluster File System)
- Zugriff mit Filesystem-Befehlen
- Schlechtere Performance

- Ø **Raw Devices**
- Gute Performance
- Ungewohnte Verwaltung
- Kein Zugriff mit Filesystem-Befehlen

Shared Storage - Möglichkeiten

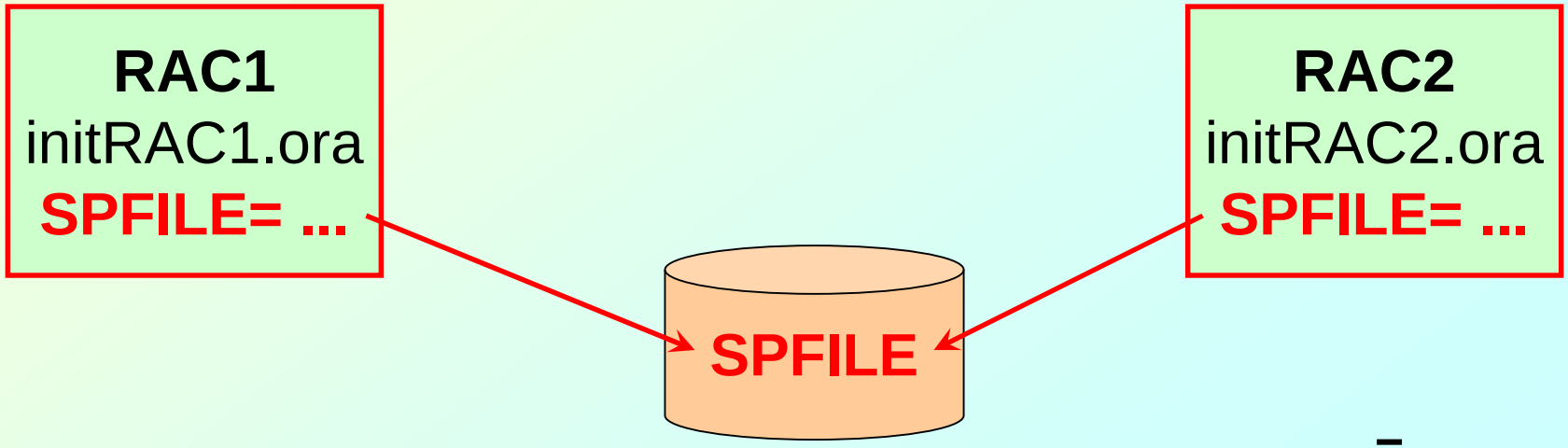
	CFS	ASM	RAW
Oracle Home	ja*	nein	ja
CRS-Home	ja*	nein	ja
OCR	ja	nein	ja
Voting Disk	ja	nein	ja
Datendateien	ja	ja	ja
Redologs	ja	ja	nein
Archivelogs	ja	ja	nein
Spfile*	ja	ja	ja

Verwaltung des SPFILE

- Parameter können mit `ALTER SYSTEM SET` von jeder Instanz geändert werden.

```
ALTER SYSTEM SET parameter SCOPE=MEMORY sid='<sid|*>';
```

- SPFILE-Einträge
  - `*.parameter` beziehen sich auf alle Instanzen
  - `sid.parameter` bezieht sich nur auf *sid*
  - *`sid`*`.parameter` geht vor `*.parameter`



Parameter mit identischen Werten auf allen Instanzen

- **ACTIVE_INSTANCE_COUNT**
- **ARCHIVE_LAG_TARGET**
- **CLUSTER_DATABASE**
- **CONTROL_FILES**
- **DB_BLOCK_SIZE**
- **DB_DOMAIN**
- **DB_FILES**
- **DB_NAME**
- **DB_RECOVERY_FILE_DEST**
- **DB_RECOVERY_FILE_DEST_SIZE**
- **DB_UNIQUE_NAME**
- **TRACE_ENABLED**
- **UNDO_MANAGEMENT**

Ablauf der Installation

- Voraussetzungen prüfen (Betriebssystem, Storage)
- Netzwerkkonfiguration (privates und öffentliches Netz)
- Nutzer- und Verzeichnisäquivalenz herstellen
- Knoten konfigurieren
- CFS oder ASM installieren
- Oracle Clusterware installieren
- Oracle Software installieren
- Virtuelle IP-Adressen einrichten (VIPCA)
- Datenbank erstellen (manuell oder DBCA)
- Failoverkonfiguration

Starten und Stoppen von Instanzen

- `start/stop` mit SVRCTL (Syntax):

```
srvctl start|stop instance -d <db_name> -i <inst_name_list>
[-o open|mount|nomount|normal|transactional|immediate|abort>]
[-c <connect_str> | -q]
```

```
srvctl start|stop database -d <db_name>
[-o open|mount|nomount|normal|transactional|immediate|abort>]
[-c <connect_str> | -q]
```

- Beispiele:

```
srvctl start instance -d RACDB -i RACDB1,RACDB2
```

```
srvctl stop instance -d RACDB -i RACDB1,RACDB2
```

```
srvctl start database -d RACDB -o open
```

Netzwerkkonfiguration für Hochverfügbarkeit

Namensauflösung und **Listener** – Lastverteilung und Failover konfigurieren

- Connect Time Failover: Verbindung zu einem anderen Listener (Knoten), falls der ursprünglich kontaktierte nicht verfügbar ist.
  `SOURCE_ROUTE=ON`
- Transparent Application Failover: Automatisches Reconnect zur Datenbank, falls der Knoten nicht mehr verfügbar ist. Aktive SELECT-Anweisungen können übernommen werden.
  `FAILOVER=ON`
- Client Load Balancing: Verteilung der Connects über alle zur Verfügung stehenden Listener-Adressen (Methode round robin)
  `LOAD_BALANCE=ON`
- Connection Load Balancing: Connects werden je nach Workload gleichmäßig über die Knoten, Instanzen und, sofern Shared Server verwendet wird, Dispatcher verteilt.
  Initialisierungsparameter REMOTE_LISTENER setzen

Parameter von Transparent Application Failover

Konfiguration des Failover mit dem Subparameter FAILOVER_MODE

- TYPE:
  `SESSION` wird auf einen erreichbaren Knoten umgeleitet
  `SELECT`-Anweisungen werden gehen auf die neue Instanz über
  `NONE` Verbindungen gehen verloren (Standard)
- METHOD:
  `BASIC` Im Failoverfall muß die Verbindung zur neuen Instanz erst aufgebaut werden.
  `PRECONNECT` Bei jedem Connect wird auch eine Verbindung zu einer zweiten (Failover-)Instanz hergestellt.
- RETRIES: Anzahl der neuen Connect-Versuche
- DELAY: Wartezeit zwischen den Reconnect-Versuchen
- BACKUP: Alternativer TNS-Name

Fast Application Notification

FAN ist neu in Oracle 10*g* Release 2

- RAC-spezifisches Feature zur Hochverfügbarkeit
- Basis: Versendung von HA-Events im RAC über ONS
- Vermeidung von Ausfallzeiten wegen z.B. TCP-Timeouts
- Applikationen, die auf einen TCP-Timeout warten würden, können mit FAN gezielt benachrichtigt werden, und das Failover durchführen.
- Applikationen werden benachrichtigt, wenn sie versuchen, sich zu einem Service zu verbinden, der nicht verfügbar ist.
- Applikationen werden benachrichtigt, wenn der Service wieder zur Verfügung steht, damit sie sich verbinden können.
- Auf die Ereignisse kann reagiert werden durch:
  - Starten und stoppen serverseitiger Applikationen
  - Verlagerung von Ressourcen
  - Versendung von Informationen
- FAN-Unterstützung: OCI, JDBC, ODP.NET (jeweils 10g Release 2)

Services

- Anwendungen, die Unternehmensbereichen entsprechen und ähnliche Workloadanforderungen haben.
- n:m Beziehung zwischen Instanz und Service
- Technik: Weiterentwicklung der Net-Service-Namen (Auflösung der Services in der `tnsnames.ora`)
- Ziele:
  - Automatisches Workloadmanagement in Verbindung mit dem Ressource Manager
  - Automatisches Connection Load Balancing
  - Zugriff auf spezifische Connections
- Ermöglicht serverseitiges TAF
- Geht (noch) nicht über Database Control (EM)

Methodenvergleich 1

1. RAID
   - Ø Spiegelung und Striping auf Hardwarebene
   - Ø Oracle empfiehlt RAID 1+0 für OLTP und RAID 5 für DSS (überwiegend lesender Zugriff)
   - Ø Konflikt von Performance, Sicherheit und Kosten
2. Spiegelung
   - Ø Empfohlen für Online Redo Logs, Kontrolldateien und Archivierte Redo Logs
   - Ø Macht nur Sinn bei Spiegelung auf verschiedene Geräte
3. Archivierung
   - Ø Wesentlich für die Option einer crashnahen Wiederherstellung
   - Ø Möglich auf lokale und Remote-Ziele
   - Ø Zwingend erforderlich für andere Methoden (Data Guard, Flashback)
4. Export (logisches Backup, in 10*g* erweitert zu *Data Pump*)
   - Ø Kein Ersatz für physikalisches Backup, bietet zusätzliche Sicherheit
   - Ø Fixierung eines *status quo* der Datenbank (Schutz vor Benutzerfehlern und Datenkorruption)
   - Ø Export von Strukturen (Metadaten)
   - Ø Inkrementelle Exporte möglich

Methodenvergleich 2

5. Backup (physikalisches Backup)
   - Ø In unterschiedlichen Modi möglich:
     - Offline (Imagekopie) ⇔ Online (Backupmodus)
     - Konsistent (nur Offline) ⇔ Inkonsistent (Offline oder Online)
     - Komplett (alle Datendateien) ⇔ Partiell (einzelne Tablespaces)
     - Voll (alle Blöcke) ⇔ Inkrementell (nur geänderte Blöcke)
   - Ø Kann realisiert werden mit Betriebssystemmitteln oder dem Recovery Manager (RMAN)
6. Oracle Flashback (Benutzerfehler und schnelles Recovery)
   - Ø Schutz vor Benutzerfehlern
   - Ø Schnelles Point-in-Time-Recovery
   - Ø Auf der Basis des UNDO Management
     - Flashback Query (Historische Daten abfragen – einzige Möglichkeit in 9i)
     - Flashback Table (Tabelle zurücksetzen)
   - Ø Auf der Basis des Recycle Bin
     - Flashback Drop (Schemaobjekte wiederherstellen)
   - Ø Auf der Basis von Flashback Logs
     - Flashback Database (Datenbank zurücksetzen)

Methodenvergleich 3

7. Oracle Data Guard - Physical Standby Database

- Hochverfügbarkeitslösung (Anwendung der Redo Logs auf sekundäre DB) Failover nach Crash des Produktionssystems oder Switchover für Pflegemaßnahmen
- Alternative Quelle für Backup (Entlastung des Produktionssystems)
- Zeitversetztes Nachführen, d.h. Schutz vor Fehlern
- Öffnen für Reporting möglich, für die Verfügbarkeit aber nicht sinnvoll

8. Oracle Data Guard - Logical Standby Database

- Hochverfügbarkeitslösung (Anwendung der SQL auf sekundäre DB),
- Ist immer offen, kann zusätzliche Objekte enthalten (Indizes, MVs)
- Failover oder Switchover möglich, sinnvoll aber eher für Reporting (Lastverteilung)
- Nicht alle Datentypen und Objekte möglich (Log Miner-Technologie)

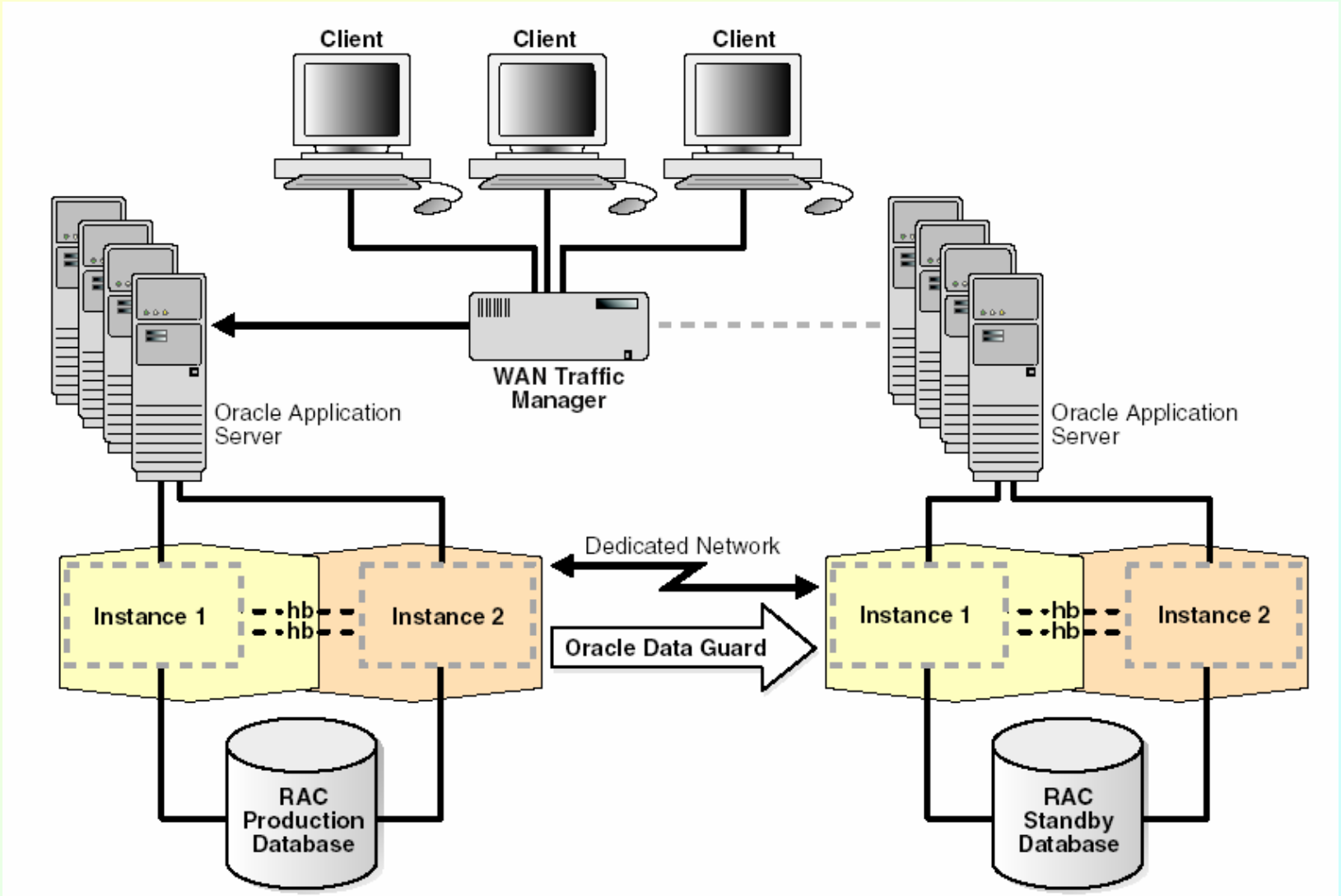
9. Duplikate

- Mit RMAN (zwingend erforderlich) aus dem Produktionssystem abgeleitet
- Eigenständige DB (für Test und Reporting, weniger Hochverfügbarkeit)
- Kann vom RMAN regelmäßig mit Produktionssystem synchronisiert werden

Methodenvergleich 4

10. Real Application Clusters (RAC)
    - Ø DIE **lokale** Hochverfügbarkeitslösung von Oracle
    - Ø Transparentes Failover (Minimaler Transaktionsverlust)
    - Ø Mehrere Instanzen greifen auf eine Datenbank zu
    - Ø Hauptproblem Kosten
11. Real Application Clusters Guard (Cold Standby unter Windows)
    - Ø Transparentes Failover
    - Ø Redundante Hardware im Normalbetrieb nicht nutzbar
12. Oracle Advanced Replication
    - Ø Replikation von Objekten oder Gruppen von Objekten in verteilten Systemen
    - Ø Aktualisierung durch DB-Links (synchron oder asynchron) und transparente Verwaltung konkurrierender DML
    - Ø Möglich als Master Replication (Peer-to-Peer Replication) oder Materialized View Replication (Snapshots)
    - Ø Nicht alle Objekte können repliziert werden (z.B. Sequenzen)
13. Oracle Streams
    - Ø Infrastruktur für gemeinsame Datennutzung und -integration
    - Ø Basiert auf dem Log Miner
    - Ø Gegenüber Advanced Replication eingeschränkte Funktionalität

Oracle Maximum Availability Architecture



Literatur

- A. Held: Oracle 10g. Hochverfügbarkeit. Addison-Wesley, München 2005
- M. Hart, S. Jesse: Oracle Database 10g High Availability with RAC, Flashback & Data Guard. Osborne/McGraw-Hill, New York 2004.
- SIG Database 9.9.2005 (Beiträge von M. Michalewicz, Orcacle Deutschland) http://www.doag.org/public//sig/database
- http://metalink.oracle.com, http://puschitz.com, http://otn.oracle.com
- Oracle10*g* Database Online Documentation, Release 2, vor allem folgende Teile:
  - Oracle Database High Availability Overview
  - Oracle Data Guard Concepts and Administration
  - Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide
  - Oracle Database Advanced Replication
  - Oracle Streams Concepts and Administration Guide

Dr. Frank Haney
info@it-haney.de
Tel.: 03641-210224

ORACLE | CERTIFIED PROFESSIONAL