

HA² - RAC trifft Dataguard

oder: HA für unentschlossene

About Me

- Benjamin Kurschies
- Jahrgang 1980
- seit 2007 Oracle DBA
- Seit 2015 Stellv. DOAG Regioleiter Hamburg

Technisch

- Spezialisiert auf HA und Performance Tuning
- aktuell noch Fokus auf Oracle 11.2

ORACLE®

Certified Professional

Oracle Database 11g
Administrator

ORACLE®

Certified Expert

Oracle Real Application
Clusters 11g and
Grid Infrastructure
Administrator

ORACLE®

Certified Expert

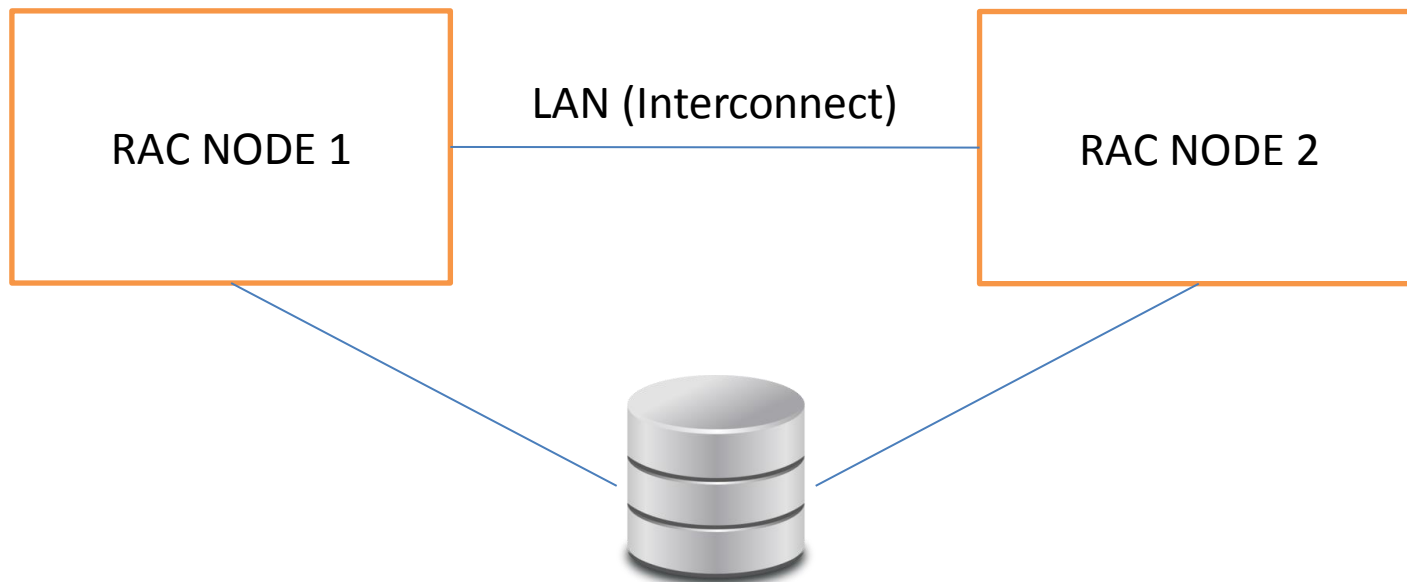
Oracle Database 11g
Performance Tuning

Inhalt

- RAC
- Dataguard
- Warum Kombinieren?
- Das Zusammenspiel
- Fallstricke
- Q&A

Kurz erklärt: RAC

- Active – Active Cluster
- Shared Disk System



Kurz erklärt: RAC – 2

Vorteile:

- Lastverteilung *
- TAF (sofern von App unterstützt)
- einfache Installation

Nachteile

- umfangreiche Vorbedingungen
- kompliziertere Fehlersuche bei Störungen
- Geografische Abhängigkeit
- eingeschränkte Storage Redundanz *
- Datenbank ist „single point of failure“

Kurz erklärt: RAC - 3

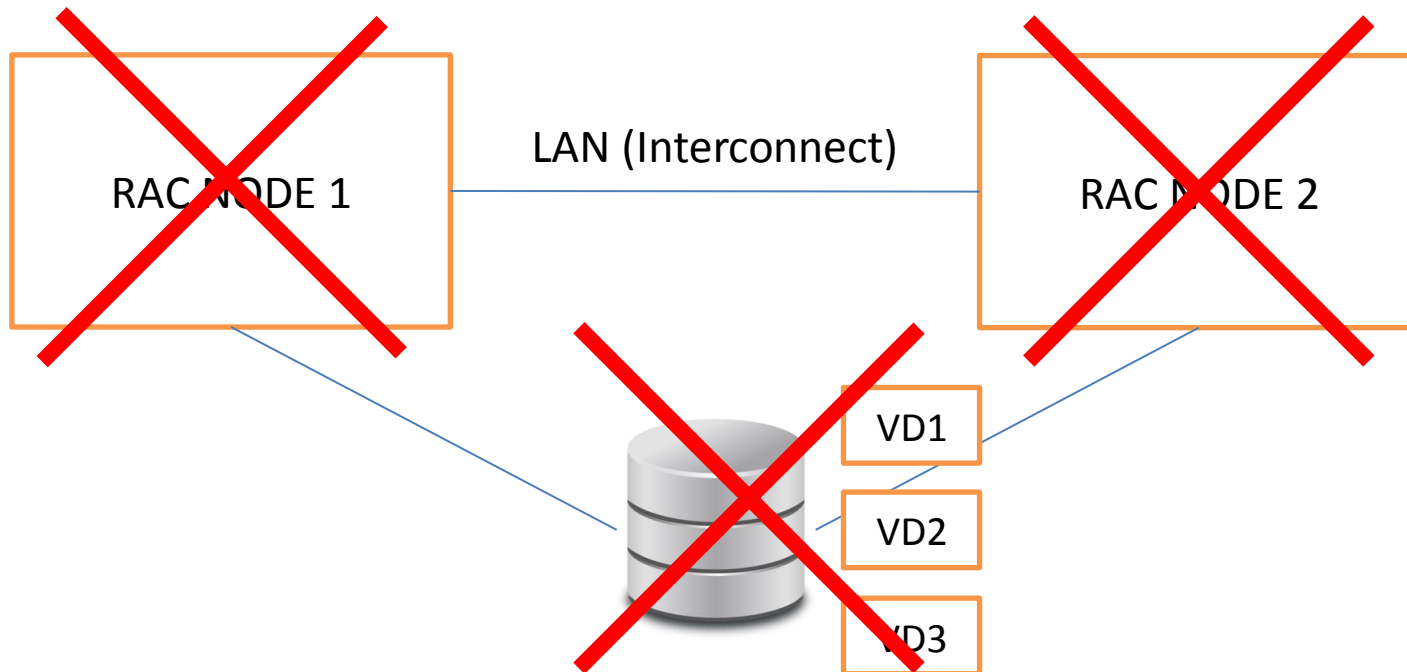
Technik:

- vor Installation der DB Software wird die Oracle Clusterware installiert
- Es wird ein Shared-Disk system benötigt, z.B.
 - SAN (FC/iSCSI) mit gescharten LUN's & ASM
 - NFS Share
 - Hinweis: bei Oracle SE muss ASM genutzt werden
- Die Installation des DB Homes muss nur auf einem Knoten erfolgen
 - über das GUI werden die anderen zu installierenden Knoten ausgewählt
- Das anlegen der Datenbanken (dbca) muss nur auf einem Knoten erfolgen

Kurz erklärt: RAC - 4

Problem: Storage Redundanz

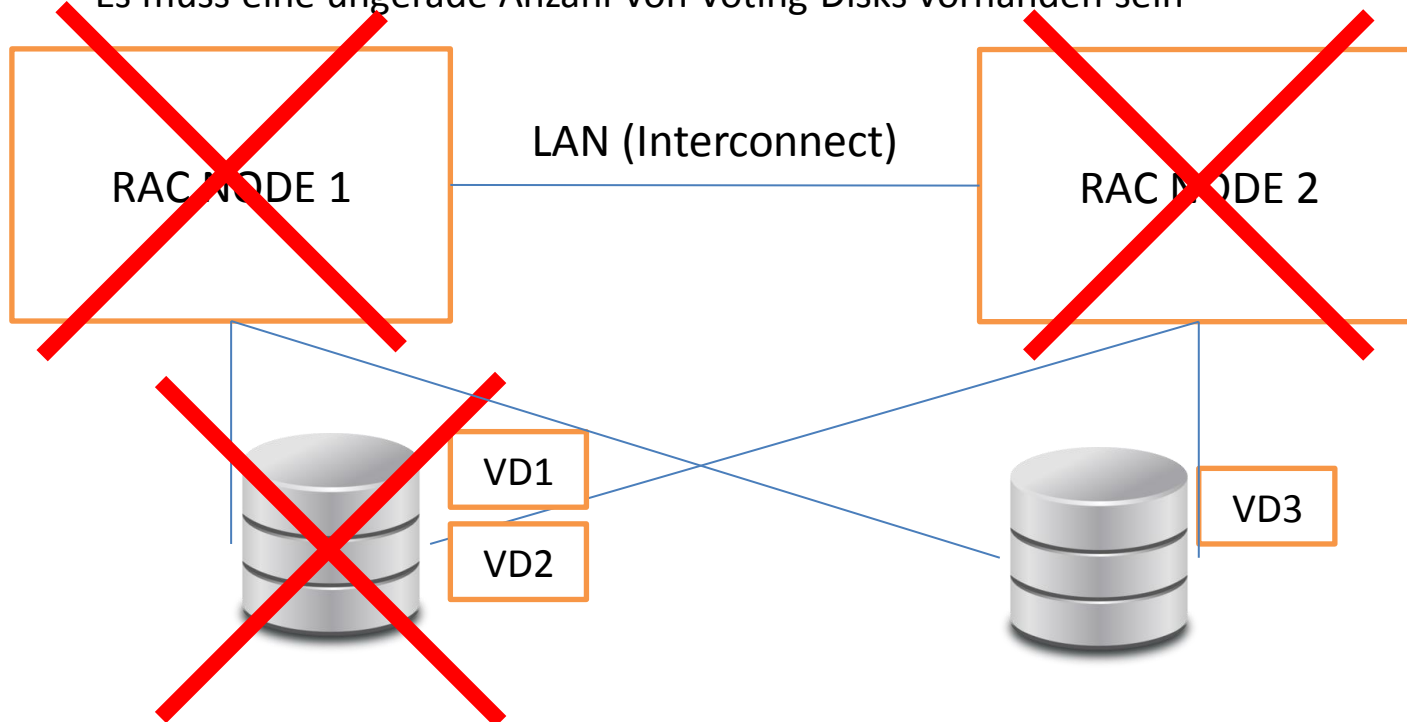
- Wenn nur ein Storage vorhanden ist, sind bei einem Ausfall alle DB's down



Kurz erklärt: RAC - 5

Problem: Storage Redundanz

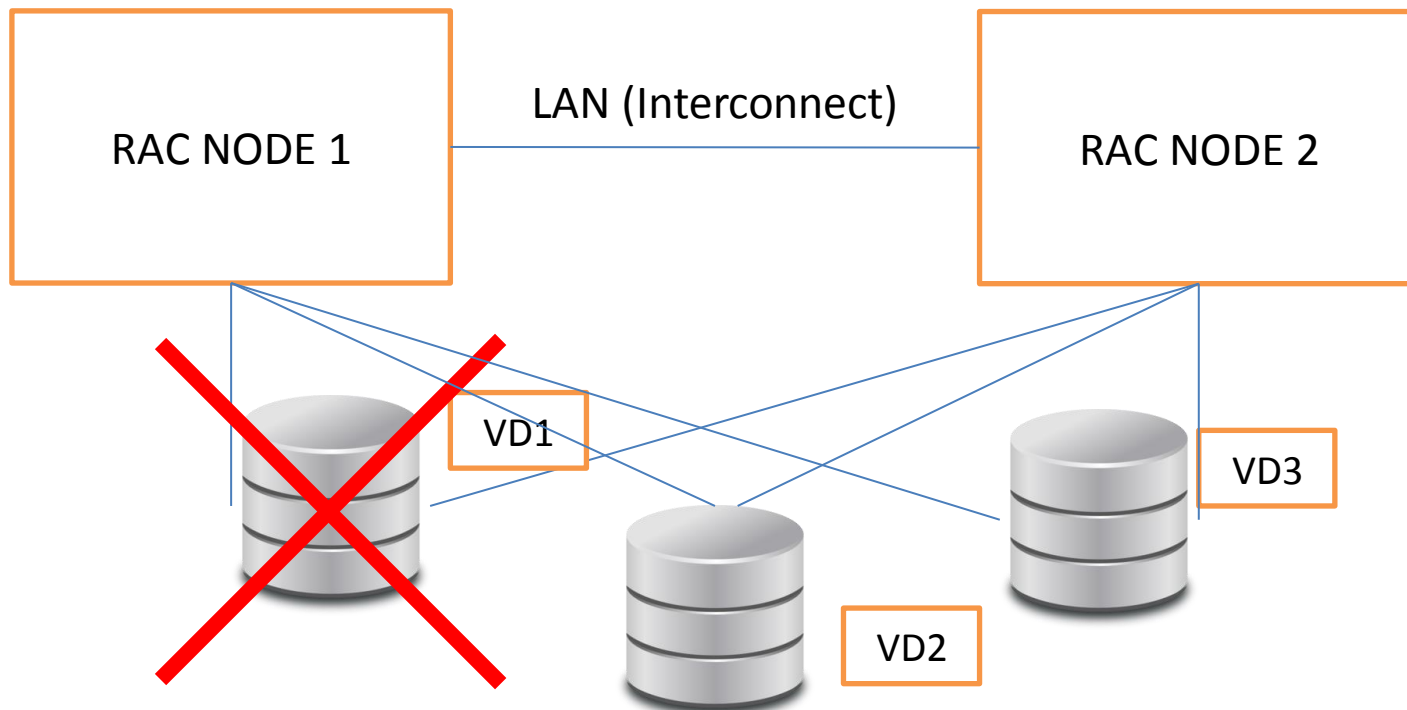
- bei mehr als einem Storage...
 - Um ein „Split-Brain“ bei Ausfall des Intrconnect zu verhindern, werden voting disks verwendet.
 - Es muss eine ungerade Anzahl von Voting Disks vorhanden sein



Kurz erklärt: RAC - 6

Problem: Storage Redundanz

- erst ab 3 unabhängigen Storage Systemen ist eine „echte“ storage Redundanz gegeben
 - Hinweis: 3. Storage System muss nicht schnell sein



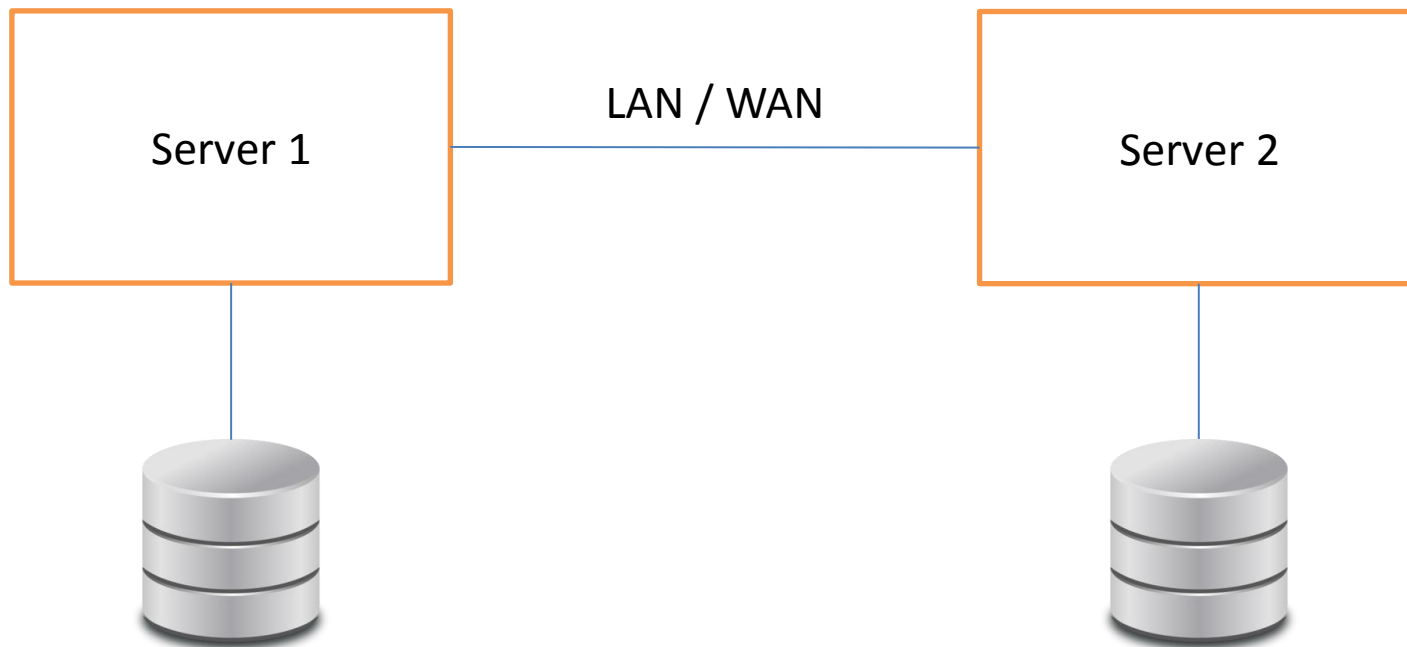
Kurz erklärt: RAC - 7

Fazit:

- RAC stellt den „Service“ bei Ausfall einer Instanz sehr schnell wieder her – Vollautomatisch
- Sofern von der Applikation unterstützt komplett transparent für den Enduser
- Storage ist den meisten Fällen ein SPOF
- die Datenbank selbst ist ein SPOF

Kurz erklärt: Dataguard

- Active – Passive „Cluster“
- Log Shipping Mechanismus



Kurz erklärt: Dataguard - 2

Vorteile

- Replikation auch über große Entfernungen möglich
- Komplette unabhängig von der Infrastruktur

Nachteile

- Standby kann nicht produktiv genutzt werden
 - (außer aktive Standby)
- umfangreiche / „komplizierte“ Einrichtung
- Oracle EE

Kurz erklärt: Dataguard - 3

Technik:

- Vorarbeiten

- Listener anpassen
- „orapw“ Datei erstellen / kopieren
- auf spfile umstellen (!) / spfile für Standby erstellen
- tns einträge hinzufügen
- force_logging einschalten
- standby_redo_logs anlegen

- Kopieren der DB

- z.B. via rman duplicate:

```
rman target sys@primary auxiliary sys@standby_dgmgrl  
duplicate target database for standby from active database;
```

Kurz erklärt: Dataguard - 4

Technik (fortsetzung):

- DG Configuration erstellen

- dg_broker starten (`alter system set dg_broker_start=true`)
dgmgrl /
create configuration DB as primary database is PRIMARY ...
add database STANDBY as connect identifier is
enable configuration

- Kontrolle (!)

- dgmgrl> show configuration
- standby – alertlog: logs müssen „in transit“ sein

Kurz erklärt: Dataguard - 5

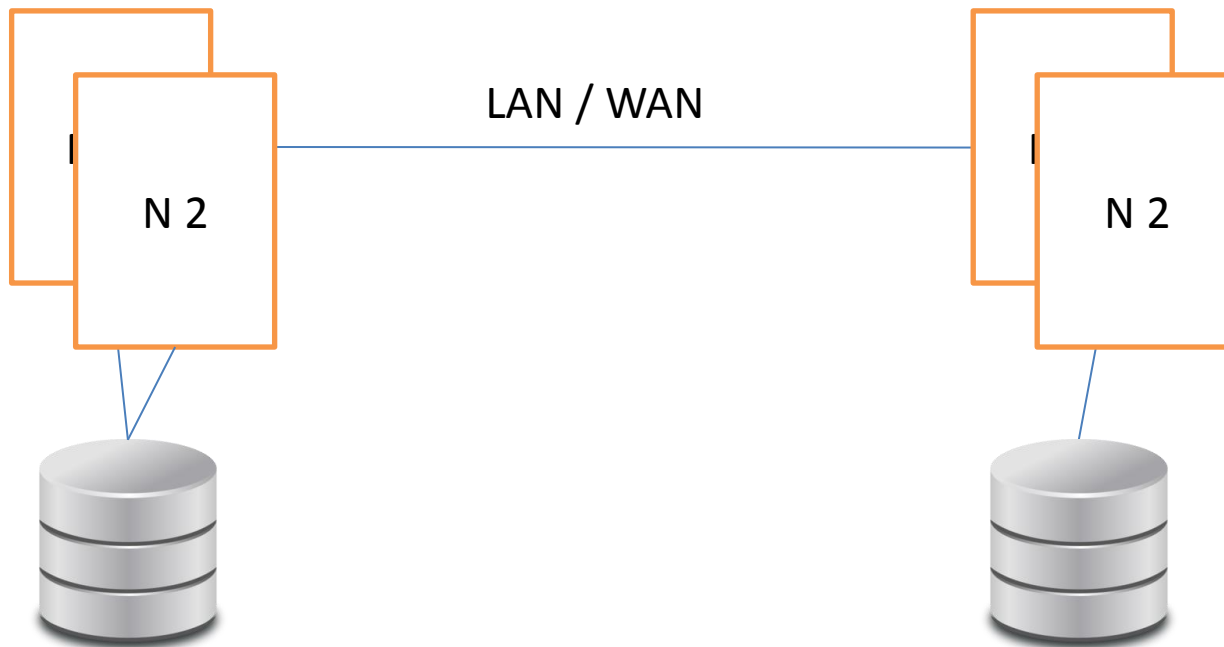
Technik (fortsetzung):

- Best-Practise:

- db_create_file_dest setzen
- standby_file_management auf „AUTO“ setzen
- Wenn „log_archive_dest_n“ gesetzt ist, manuell applied archive logs auf Stdb. löschen
- Wenn DB im ASYNC mode betrieben wird, Flashback aktivieren
 - * reduziert die Zeit zum Re-Sync nach einem Failover

Das Zusammenspiel

- Prinzipiell können beide HA Technologien beliebig miteinander kombiniert werden



Warum Kombinieren?

Folgende Situation hat uns dazu bewogen, beide Technologien miteinander zu kombinieren:

1. Bei Ausfall eines Servers muss die DB innerhalb von Sekunden wieder bereitstehen
2. Die Applikation arbeitet Seriell und benötigt extrem kurze Reaktionszeiten, Synchrones DG ist zu langsam
3. es muss K-Fall abgesichert sein (Ausfall eines kompletten RZ)

Das Zusammenspiel

- Folgende „kleinigkeiten“ müssen beachtet werden:
 - es müssen (standby) redo logs für jeden thread angelegt werden
 - * Wenn Primary 2 Knoten und Standby 3 Knoten hat, benötigt Primary 3 Threads (!)
 - es müssen „dgmgri“ einträge in der TNSNAMES.ora für jede Instanz angelegt werden
 - es muss auf jedem Knoten ein statischer Listener Eintrag erstellt werden.
 - bei RAC muss das spfile auf shared Storage liegen
 - bei RAC müssen die DG_BROKER_CONFIG_FILEn auf shared Storage sein.

Fallstricke

- Undo Tablespace (Admin Managed Databases)
 - jede Instanz im RAC benötigt seinen eigenen Undo Tablespace
 - Wenn die Standby-DB mehr Instanzen hat als das Primär-System müssen auf der Primär-Site entsprechend zusätzliche Undo Tablespaces angelegt werden (!)
 - beim erstellen des spfile auf der Standby müssen die Instanzen einem undo tablespace zugewiesen werden
- Datenbank in Clusterware registrieren
 - via „srvctl“ die Datenbank im richtigem Modus (z.B. role=physical_standby, startup=mount) registrieren
 - bei einem Schwenk (switchover / failover) die Einstellungen anpassen (srvctl modify database ...)

Fallstricke - 2

- Performance
 - nur eine Standby-Instanz kann recovern
 - Bei Datenbanken mit hoher Last kommt die Standby „nicht hinterher“
 - Siehe MAA „Redo Apply Best Practices“
 - z.B. DB_BLOCK_CHECKING und DB_BLOCK_CHECKSUM abschalten
 - Achtung: nur auf der Standby – nach Switchover / Failover anpassen!

Q & A

Benjamin Kurschies

benjamin@kurschies.de

www.xing.com/profile/Benjamin_Kurschies