Data Guard durch Firewalls? - Kein Problem mit Connection Manager!

Mathias Zarick
Trivadis Delphi GmbH
1200 Wien

Schlüsselworte
Data Guard, Connection Manager, Connectivity, Oracle Net Services

Einleitung
Data Guard hat sich als Hochverfügbarkeitstechnologie für Oracle etabliert. Er spiegelt zuverlässig die Daten. Durch Fast-Start Failover ist es auch möglich die Service-Verfügbarkeit in den Griff zu bekommen. Doch wie ist das mit dem Zugriff der Clients? Hierfür gibt es anerkannte Best Practices. Sie wurden in einem durch Trivadis erfolgreich abgewickelten Projekt mit weiteren Anforderungen kombiniert. Die Clients sind in unterschiedlichen Netzwerksegmenten und haben teilweise Firewalls zwischen den Datenbankservern und sich selbst. Diese Herausforderungen galt es in dem Projekt zu meistern.

Ablöse der Streams Multimaster-Replikation
Der Kunde setzte in einem vorigen Konzept auf Oracle Streams. Mit Streams war es möglich eine Multimaster-Replikation zu betreiben, welche die Daten zwischen verschiedenen Datenbanken asynchron austauscht. Hier setzte der Kunde 5 Datenbanken ein, was zu folgender Architektur führte:

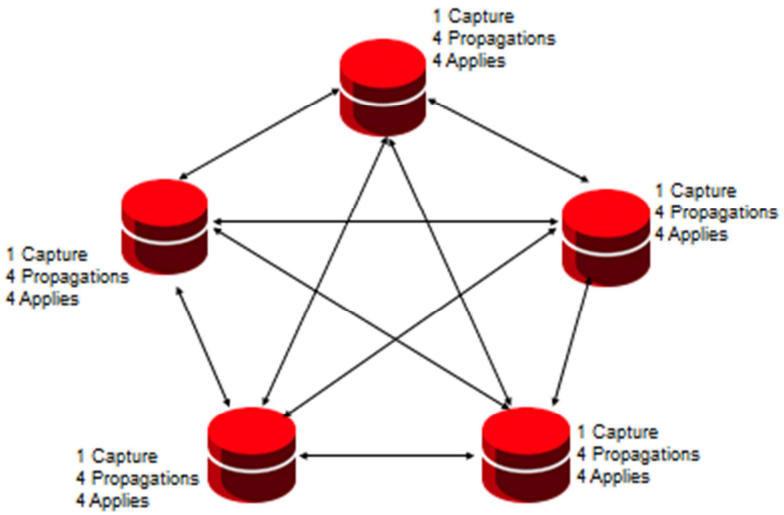

Abb. 1: Streams Setup

Die Abbildung zeigt einen vollständig vermaschten Graphen, d.h. jede DB hat die Änderungen durch den Capture erfasst und an jede andere DB durch die Propagation gesendet. Durch die Apply-Prozesse glichen sich die Daten aus.
Dieses Setup funktionierte in den meisten Fällen zuverlässig, doch zeitweise wurde dem Projekt die Asynchronität der Replikation zum Verhängnis. Da die Applikation auch unabhängig von Streams Advanced Queuing verwendet, gab es eine erhöhte Komplexität, die zunächst auch gemeistert werden konnte. Aber irgendwann kam man zu der Erkenntnis, dass es in seltenen Fällen dazu gekommen war, dass Messages aus dem

Advanced Queuing eintrafen, bevor die dazugehörigen referenzierten Daten angekommen waren. Eine Lösung für dieses Problem wäre aufwendig gewesen, und sehr wahrscheinlich hätte man auch in die Applikation eingreifen müssen.
So wurde schnell klar, dass eine andere Lösung für die Hochverfügbarkeit implementiert werden musste.

Die Anforderungsanalyse
Die Applikation benutzt Standard Oracle Clients, die über Oracle Net auf die DB zugreifen. Hochverfügbare Oracle Connect Deskriptoren sind hier also zu verwenden. Zu Streams Zeiten hatte jeder Client mindestens 2 DBs in seinem Netzwerksegment, so war das Problem gelöst. Mehrere DBs sollten in der neuen Lösung aber wegen dem Asynchronitätsproblem vermieden werden.
Die Daten sind über zwei verschiedene Data Center zu spiegeln. Die dafür zu nutzenden Server sind in unterschiedlichen Netzwerksegmenten, die durch eine Firewall voneinander getrennt sind.
Schnell wurde klar, dass man mittels Data Guard eine Hochverfügbarkeit für Service und Daten zur Verfügung stellen kann. Auch ein RAC wäre möglich gewesen, dieser wurde aber aus Gründen der Komplexität für Administration und Storage-Architektur sowie den erhöhten Lizenzkosten ausgeschlossen.
So wurde das Data Guard Konzept weiterverfolgt und verfeinert. Ein automatischer Failover in Ausfallszenarien der Primary wurde schnell beschlossen, da manuelle Eingriffe bei Störungen in dem System nicht tolerierbar sind. Es wurde also festgelegt, Fast-Start Failover zu implementieren. In dem Projekt kam folgende Software zum Einsatz, welche vor allem durch den Kunden und den Applikationshersteller beeinflusst waren:

- Windows Server 2008 R2
- Oracle Database 11.2.0.3
- das letzte damals verfügbare Oracle Windows Patch Bundle 13 für 11.2.0.3 (Nov. 2012)

Als Lizenzen standen Oracle Enterprise Edition Lizenzen ohne weitere Optionen oder Packs zur Verfügung.
Die zahlreichen Clients der Datenbankapplikation sind in den 2 verschiedenen Netzwerksegmenten lokalisiert. Sie haben keinen Zugriff auf das jeweils andere Netzwerk. Eine Administration von unzähligen Firewallregeln für alle diese Clients war ausgeschlossen.

Überblick der konzipierten Architektur
In jedem Netzwerk muss es 2 Rechner geben, über die eine Verbindung zur Datenbank aufgebaut werden kann, so dass bei einem Ausfall immer noch ein Rechner vorhanden ist. So kam der Connection Manager ins Spiel. Durch den Connection Manager wird die folgende Architektur möglich:

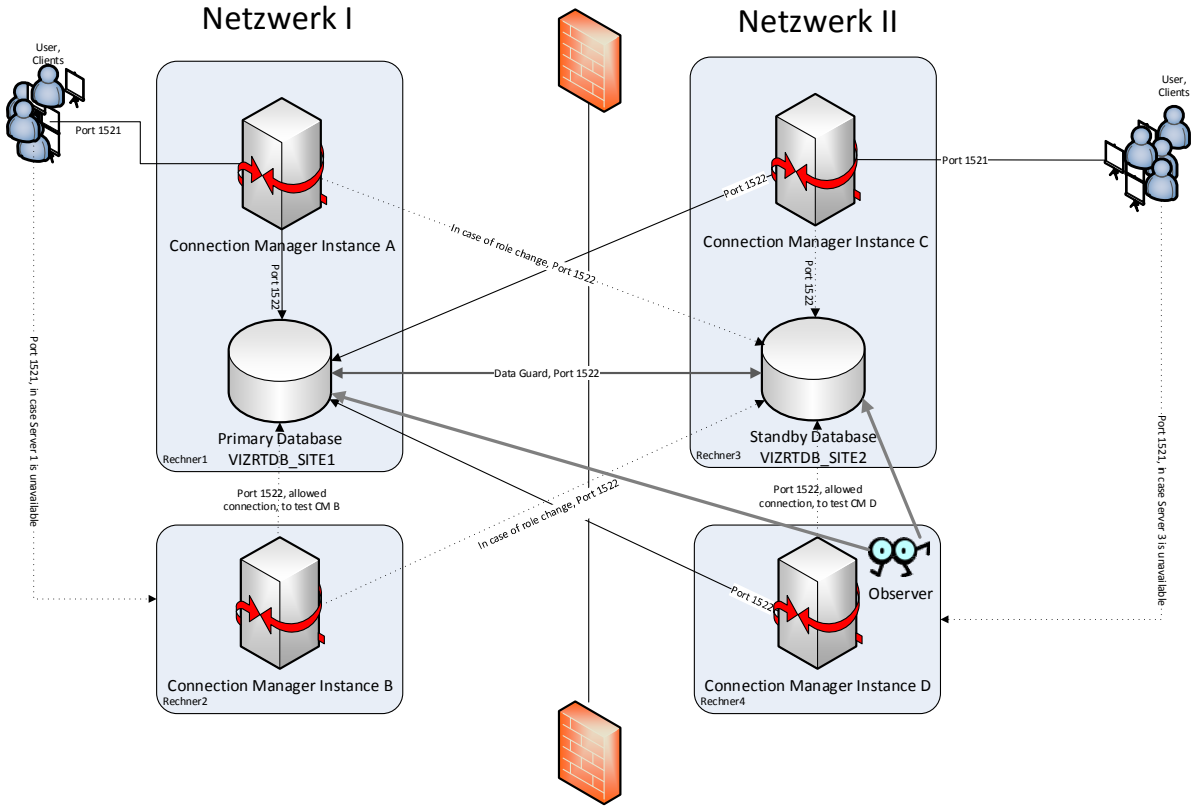

Abb. 2: Architektur mit Data Guard und Connection Manager

Die zentrale Komponente dieser Architektur, welche für die Hochverfügbarkeit des Gesamtsystems zuständig ist, ist Oracle Data Guard. Dieses Feature erlaubt das Betreiben einer synchronen Kopie der Oracle Datenbank auf einem zweiten Rechner, einer Standby Database. Es wird eine physical standby database betrieben, bei der alle Inhalte wirklich blockident sind. Der Betrieb dieser Umgebung ist wesentlich einfacher als die des Vorgängers mit einer Streams Multimaster Replikation. Außerdem gibt es für die Oracle Sessions, welche alle auf der Primary DB angemeldet sind keine Asynchronität oder Divergenz von Daten, alle Clients sehen zu einer Zeit immer denselben Datenbestand.
Oracle Data Guard ist in der Enterprise Edition der Oracle Database enthalten. Eine zusätzliche Option Active Data Guard, welche es ermöglicht die Standby DB read only zu öffnen und trotzdem aktuell zu halten (Real Time Query), wäre extra zu lizensieren gewesen, wurde in diesem Projekt aber nicht benötigt und nicht benutzt.

Connection Manager
Da der Kunde strikte Richtlinien in Bezug auf mögliche Connectivity hat, ist eine Firewall zwischen Primary und Standby DB installiert. Um die Oracle Clients, die sich sowohl im Netzwerk der Primärdatenbank als auch im Netzwerk der Standby Datenbank befinden, auf die Datenbank verbinden zu lassen und die Firewall nicht mit zu vielen Ausnahmen zu konfigurieren, wurde Oracle Connection Manager (CM) verwendet. CM ist eine Software von Oracle, die einen mehrstufigen Verbindungsaufbau erlaubt. CM selbst baut die Verbindung zum DB Service auf und leitet den Oracle Net Verkehr wie ein Proxy weiter (Oracle Net Proxy). Es wurden Connection Manager Instanzen auf dem Rechner der Primär und der Standby Datenbank installiert. In jedem Netzwerk wurden je 1 weiterer Connection Manager installiert, um Rechnerausfälle im jeweiligen Netzwerksegment überstehen zu können.

Installationsdetails und Konfigurationen
Das Data Guard System (Redo Log Transport und Broker Kommunikation) wird auf Port 1522 betrieben, sämtliche Kommunikation zwischen den DBs und von dem Observer basiert auf Port 1522. Oracle Administrationszugriffe passieren ebenfalls über Port 1522. Der Port 1522 ist jedoch für Zugriffe der

Applikation und von End-Usern nicht erlaubt. Es soll ausschließlich über den Connection Manager zugegriffen werden, um sicherstellen zu können, alle Connectivity in Problem- und Desasterfällen überblicken und administrieren zu können.
Die Connection Manager werden über Port 1521 erreicht. Sie selbst greifen dann auf die DB über Port 1522 zu.
Durch einen Eintrag in der sqlnet.ora des DB Servers wird sichergestellt, dass nur die erlaubten Server auf Port 1522 zugreifen können:

```
SQLNET.EXPIRE_TIME = 5   # Dead Connection Detection, Keep Alive Packets
TCP.VALIDNODE_CHECKING=yes
TCP.INVITED_NODES=(Rechner1,Rechner2,Rechner3,Rechner4)
```

Wird ein direkter Zugriff auf die Datenbank über den Standard Listener auf Port 1522 von einem anderen Rechner versucht, der nicht in der Liste ist, so wird dort der Fehler „ORA-12537 TNS:connection closed“ geworfen.
Die Datenbank betreibt folgende DB Services, welche dynamisch auf der jeweiligen Instanz gestarten werden:

- VIZRTDB — Der „normale“ Service für den read write Zugriff, welcher nur auf der Primary DB gestartet ist
- VIZRTDB_RO — Falls die Standby DB einmal read only geöffnet werden sollte, um entsprechende Recherchen zu machen
- VIZRTDB_SNAP — Falls die Standby DB einmal read write geöffnet werden sollte, um entsprechende Recherchen zu machen (Snapshot Standby)

Die Services werden dynamisch an dem DB Listener auf Port 1522 registriert, sobald die DB Instance in dem entsprechenden Modus geöffnet ist. Ebenso werden sie an allen Connection Managern registriert.
Dafür werden auf den DB Instanzen folgende init.ora parameter gesetzt:

local_listener Rechner1, VIZRTDB_SITE1	(ADDRESS=(PROTOCOL=TCP)(HOST=Rechner1)(PORT=1522))
local_listener Rechner3, VIZRTDB_SITE2	(ADDRESS=(PROTOCOL=TCP)(HOST=Rechner2)(PORT=1522))
remote_listener (zeigen auf alle CMs, für beide DB Instanzen gleich)	(ADDRESS_LIST= (ADDRESS=(HOST=SWI19GRVP)(PROTOCOL=TCP)(PORT=1521)) (ADDRESS=(HOST=SWI19GRCM2)(PROTOCOL=TCP)(PORT=1521)) (ADDRESS=(HOST=SWI19RPVP)(PROTOCOL=TCP)(PORT=1521)) (ADDRESS=(HOST=SWI19RPCM2)(PROTOCOL=TCP)(PORT=1521)))

Um den Bedarf des lizenzpflichtigen Real Time Query Features (Active Data Guard Option) zu vermeiden, darf die Standby DB nicht read only geöffnet werden, während das Recovery eingeschaltet ist. Das wird durch einen speziellen After Startup on Database Trigger sichergestellt. Dieser Trigger kümmert sich auch um das korrekte Starten der DB Services.
Die Clients verwenden folgende Connect Deskriptoren, die als TNS Aliases in der tnsnames.ora konfiguriert werden. Sie verbinden sich über die Oracle CM über Port 1521. Es wird immer über den primären CM verbunden. Falls die Verbindung nicht innerhalb von 3 Sekunden gelingt, wird über den sekundären CM verbunden. Der Service Name VIZRTDB läuft immer nur auf der Primär DB. Die CM Instanzen wissen, auf welchen Listener auf Port 1522 sie verbinden müssen, um die Oracle Net Anfrage zustande kommen zu lassen. Eine Verbindung ist immer 2-stufig. Der Client hält eine TCP/IP Verbindung über Port 1521 zum CM, der CM verbindet diese mit einer Verbindung auf den DB Server der Primär DB auf Port 1522.

```
# Network 1
VIZRTDB =
  (DESCRIPTION =
    (ADDRESS_LIST=
```

```
      (LOAD_BALANCE=OFF)
      (CONNECT_TIMEOUT=3)
      (ADDRESS = (PROTOCOL = TCP)(HOST = Rechner1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = Rechner2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = VIZRTDB)
    )
  )


# Network 2
VIZRTDB =
  (DESCRIPTION =
    (ADDRESS_LIST=
      (LOAD_BALANCE=OFF)
      (CONNECT_TIMEOUT=3)
      (ADDRESS = (PROTOCOL = TCP)(HOST = Rechner3)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = Rechner4)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = VIZRTDB)
    )
  )
```

Um den Oracle CM zu betreiben, wurde eine Oracle Client Installation durchgeführt, auf den DB Servern wurde das in einem zweiten gesonderten Oracle Home installiert. Die Installation des Client Oracle Home erfolgt als Custom Installation, gewählt werden „Oracle Connection Manager“, „Oracle Net Listener“ und „Oracle Database Utilities“. Aus diesem Oracle Home wurde außerdem von Server 4 ein Observer gestartet, welcher für das Fast-Start Failover – also das automatische Failover auf die Standby im Falle eines Ausfalles – benötigt wird. Auf Server 2 wurde dieser nur als startbar konfiguriert, aber nicht gestartet (Backup Observer).

Das Konfigurationsfile des Oracle CM:

```
cman_Rechner1.domain =
(configuration=
  (address=(protocol=tcp)(host=Rechner1)(port=1521))
  (parameter_list =
    (connection_statistics=yes)
    (max_connections=256)
    (outbound_connect_timeout=3)
    (min_gateway_processes=2)
    (max_gateway_processes=16)
    (max_cmctl_sessions=4)
    (event_group=init_and_term,memory_ops)
  )
  (rule_list=
    (rule=(src=*)(dst=*)(srv=cmon)(act=accept))   # rule for CMCTL connects
    (rule=(src=*)(dst=Rechner1)(srv=VIZRTDB)(act=accept))
    (rule=(src=*)(dst=Rechner3)(srv=VIZRTDB)(act=accept))
    (rule=(src=*)(dst=Rechner1)(srv=VIZRTDB_RO)(act=accept))
    (rule=(src=*)(dst=Rechner3)(srv=VIZRTDB_RO)(act=accept))
    (rule=(src=*)(dst=Rechner1)(srv=VIZRTDB_SNAP)(act=accept))
    (rule=(src=*)(dst=Rechner3)(srv=VIZRTDB_SNAP)(act=accept))
  )
)
```

Es werden nur Connections auf bekannte Server und Services erlaubt.

Fazit

In dem erfolgreich absolvierten Projekt wurden schwierige Herausforderungen gemeistert: Hochverfügbarkeit für eine Oracle Applikation trotz Verteilung der Clients und Server innerhalb 2 durch Firewalls getrennter Netzwerke. Die in der implementierten Architektur wesentlichen Komponenten sind Oracle Data Guard und Connection Manager. Ein durchdachtes Service Konzept, strikte und validierte Zugriffsregeln sorgen dann auch aus Sicht eines jeden Clients für Hochverfügbarkeit. Die Features, die eine Lizenz des Active Data Guard nötig machen würden, wurden durch einen Schutzmechanismus „verriegelt“. Einen Observer auf Windows zu implementieren war zu guter Letzt eine kleinere Herausforderung, welche leicht mittels Trivadis Toolbox und Microsoft Windows Resource Kit gelöst werden konnte.

Kontaktadresse:

Mathias Zarick
Trivadis Delphi GmbH
Millennium Tower
Handelskai 94-96
A-1200 Wien

Telefon: +43 1 332 35 31 32
Fax: +43 1 332 35 34
E-Mail mathias.zarick@trivadis.com
Internet: www.trivadis.com