

Oracle Database Appliance Projektbericht

Erfolgreiche Migration bei der makeIT

Marco Mischke
Robotron Datenbank Software GmbH
Stuttgarter Str.29
01189 Dresden

Schlüsselworte

Oracle Database Appliance

Einleitung

Oracle liefert mit seinen Appliances komplette Lösungen für das Rechenzentrum. Für kleinere und mittelständige Unternehmen ist die Database Appliance eine interessante Option, da hier hohe Performance zum kleinen Preis erhältlich ist.

Dieser Beitrag erläutert die Erfahrungen während eines erfolgreichen Migrationsprojektes von mehreren einzelnen Windows Servern auf die Database Appliance. Dabei wird der Weg von der initialen Analyse über die benutzten Migrationswege bis hin zum Backup erstrecken. Dabei wird auch Misserfolge während des Projektes eingegangen.

Ausgangslage

Die makeIT GmbH ist ein Hostingdienstleister aus Chemnitz. Derzeit betreuen dort zwei Datenbankadministratoren drei Rechenzentren, von denen zwei produktiv und eins als Backup genutzt werden. Darüber verteilen sich etwa produktive 15 einzelne Server, die durchgängig mit Windows laufen. Auf diesen Servern werden mehr als 30 Datenbanken betrieben von denen drei per DataGuard georedundant abgesichert sind.

Da die vorhandene Hardware sukzessive abgelöst werden sollte, wollte die makeIT die Gelegenheit nutzen um nach möglichen Einsparungen und Verbesserungen für den Betrieb zu suchen. Die bereits implementierten Verfahren wie RMAN Backup, Georedundanz durch DataGuard und vor allem die Vereinheitlichung des Betriebs durch Standards sollten durch eine Redundanz innerhalb der Rechenzentren ergänzt werden. Weiterhin sollte die Anzahl der Server nach Möglichkeit reduziert werden und es sollten nach Möglichkeit keine zusätzlichen laufenden Kosten im Lizenzbereich anfallen. Mit diesem Hintergrund beauftragte die makeIT eine Infrastrukturanalyse um entsprechende Lösungen für die genannten Zielvorgaben zu ermitteln.

Infrastrukturanalyse

Um einen Überblick über die vorhandene Datenbanklandschaft zu gewinnen, wurde eine Infrastrukturanalyse durchgeführt in deren Verlauf diverse Fakten und Kennzahlen aufgenommen wurden. Serverseitig wurden folgende Informationen zusammengetragen:

- Betriebssystem inkl. Architektur und Version
- verfügbarer Hauptspeicher
- Anzahl CPUs und CPU-Cores
- CPU Auslastung, durchschnittlich und maximal

Weiterhin wurden die vorhandenen Datenbanken inventarisiert und deren Konfiguration aufgenommen. Dazu zählen:

- Software Version und Edition
- Storage Bedarf, benutzt im Dateisystem, benutzt innerhalb der Tablespaces

- Memory Konfiguration (SGA, PGA)
- Lizenzierung (Edition, Optionen etc.)
- Status der Datenbank (Produktiv, Test, Entwicklung etc.)

Die gesammelten Daten wurden aggregiert und daraus der Ressourcenbedarf ermittelt. Aus diesen Eckdaten wiederum konnten nun entsprechende Einsparungen und Optimierungen abgeleitet werden um die vorhandene gewachsene Umgebung zu konsolidieren. Die Idee, die Systeme auf einer Oracle Database Appliance zusammenzufassen, wurde durch die Analyseergebnisse untermauert.

Aktueller Bestand	Database Appliance
15 Server	2 Server
64 CPU Cores	72 CPU Cores
10 CPU Cores aktiv (AVG)	
36 CPU Cores aktiv (MAX)	
180GB RAM	256GB RAM
1,7TB Storage, Tendenz steigend	32TB Storage
24 EE Lizenzen, keine RAC Option	Virtualized Platform mit Capacity on Demand

Planung

Mehrere Herausforderungen ergaben sich aus dieser Situation. Zum einen reicht die Zahl der vorhandenen Enterprise Edition Lizenzen nicht aus um die Database Appliance voll auszulasten. Dies kann mit Hilfe der Virtualized Platform gelöst werden. Es werden dabei nur die tatsächlich lizenzierte Anzahl CPU Kerne an die virtuelle Maschine durchgereicht. Die Anzahl kann später nach Bedarf noch erhöht werden, Oracle nennt dies „Capacity on Demand“. Zum anderen sind aktuell keine RAC Lizenzen verfügbar. Daher wurde entschieden, die Database Appliance als Aktiv/Passiv-Cluster zu betreiben. Dies erfordert einige Anpassungen im Betrieb auf die später eingegangen wird.

Weiterhin wurde von der makeIT eine Georedundanz gewünscht. Diese wird mit einer zweiten baugleichen Database Appliance in einem anderen Rechenzentrum realisiert. Für ausgewählte Datenbanken mit entsprechend hohen Service Leveln laufen auf der zweiten Database Appliance Standby Datenbanken die per DataGuard gemanaged werden. Weiterhin sollen auf der zweiten Database Appliance Test- und Entwicklungsdatenbanken laufen um die Ressourcen effizient zu nutzen.

Migrationswege

Da mit dem Wechsel auf die Database Appliance auch ein Wechsel des Betriebssystems von bisher Windows auf nun Oracle Linux erfolgt, mussten die möglichen Migrationswege evaluiert werden. Die My Oracle Support Note 733205.1 gibt hier eine Übersicht über die möglichen Wege. Für die geplante Umstellung kamen RMAN Duplicate sowie Export/Import in Frage.

Da laut MOS Note 1079563.1 zum Thema Mixed Platform Support für RMAN Duplicate der Weg zwischen Linux und Windows seit der Datenbank Version 11.1 unterstützt wird, war dies der erste

Ansatz. Bedingung hierbei ist, dass die zu migrierende Datenbank nicht upgegraded wurde. Die betreffenden Datenbanken wurden alle in 11.2 angelegt, das sollte daher kein Problem darstellen. Das Kommando zum Duplicate ist sehr simpel:

```
duplicate target database to <ziel>
from active database
spfile [...]
noresume;
```

Leider endete das sehr schnell in einem Fehler „ORA-00600 [4502]“. Ursache dafür ist offenbar das durchgeführte Update von 11.2.0.1 auf 11.2.0.4. Dieser Weg ist also ausgeschlossen. Die nächste Variante ist am Ende nur eine geringe Abwandlung, diesmal wird die Datenbank per Standby übertragen. Das Vorgehen und die unterstützten Versionen und Plattformen finden sich in MOS Note 413484.1. Dieser Weg führte zum Ziel:

```
duplicate target database for standby to <ziel>
from active database
spfile [...]
noresume;
```

Damit konnte eine 1:1 Kopie der originalen Datenbank auf der Database Appliance erzeugt werden. Außerdem bot diese Variante die Möglichkeit, die Kopie gleich per ArchiveLog Transfer und Managed Recovery auf dem aktuellen Stand zu halten. Dadurch konnte das Zeitfenster für die Umstellung und damit die Ausfallzeit extrem verkürzt werden.

Zu guter Letzt wurden noch verschiedene Wege per Data Pump Export/Import getestet. Dies ist insbesondere für kleinere Datenbanken mit geringen Anforderungen an die Verfügbarkeit praktikabel. Hierbei kommt einmal der direkte Import in die Zieldatenbank per Network Link in Frage. Sollte dies nicht fehlerfrei möglich sein, ist der Transfer klassisch mittels eines Dumpfiles immer noch offen. Bekannte Probleme zum Thema Data Pump Import findet man im My Oracle Support in Note 1947791.1.

Migration

Die eigentliche Migration der Datenbanken auf die Database Appliance besteht nun im Grunde aus folgenden Schritten:

1. Datenbank Software installieren
2. Storage für die Datenbank einrichten
3. SQL*Net vorbereiten
4. Datenbank duplizieren
5. temporären Betrieb als Standby Datenbank einrichten
6. Aktivieren der Standby Datenbank auf der Database Appliance

Das zentrale Tool zur Verwaltung der Database Appliance ist „oakcli“, mit diesem wird zuerst das Datenbank Home erstellt.

```
[root@oda-base-01 ~]# oakcli create dbhome -version 11.2.0.4.6
```

```
[root@oda-base-01 ~]# oakcli show dbhomes
```

Oracle Home Name	Oracle Home version	Home Location
OraDb11204_home1	11.2.0.4.6 (20299013,20420937)	

```
/u01/[...]/11.2.0.4/dbhome_1
OraDb12102_home1 12.1.0.2.3 (20299023,20299022)
/u01/[...]/12.1.0.2/dbhome_1
```

Nun wird das Storage für die zu migrierende Datenbank vorbereitet. Die Database Appliance benutzt seit Version 12.1.2.3.0 per Standard ACFS für die Datenbankdateien. Der entsprechende Snapshot sowie die Bereiche für Fast Recovery Area und Redologs werden wie folgt angelegt:

```
[root@oda-base-01 ~]# oakcli create dbstorage -db <DB Unique Name>
[...]
INFO: 2015-05-21 08:59:46: DATA:
/u02/app/oracle/oradata/datastore/.ACFS/snaps/<DB Name>
INFO: 2015-05-21 08:59:46: REDO: /u01/app/oracle/oradata/datastore/<DB
Name>
INFO: 2015-05-21 08:59:46: RECO:
/u01/app/oracle/fast_recovery_area/datastore/<DB Name>
```

Die ausgegebenen Pfade werden dann im weiteren Verlauf verwendet und müssen daher dokumentiert werden. Zuvor muss noch der lokale Listener angepasst werden damit das RMAN Duplicate funktioniert. Dazu muss eine SID_LIST erstellt werden um die Anmeldung an der Datenbankkopie über den Listener auch im heruntergefahrenen Zustand zu ermöglichen:

```
[grid@oda-base-01 ~]$ vi $ORACLE_HOME/network/admin/listener.ora

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = <Global DB Name>)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0.4/dbhome_1)
      (SID_NAME = <Global DB Name>)
    )
  )

[grid@oda-base-01 ~]$ lsnrctl reload
```

Passend dazu werden dann noch Einträge in der tnsnames.ora auf Quell- und Ziel-Server benötigt. Nun muss für die Datenbankkopie noch eine Passwortdatei erzeugt werden:

```
[oracle@oda-base-01 ~]$ orapwd \
> file=/u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/orapw<DB Name> \
> password=change_on_install ignorecase=y force=y
```

Damit sind alle nötigen Vorbereitungen abgeschlossen und die Datenbank kann mittels RMAN Duplicate auf die Database Appliance kopiert werden. Zu beachten sind dabei besonders die Parameter für die Speicherung der Datenbankdateien sowie die zur Namensgebung der Datenbank.

```
[oracle@oda-base-01 ~]$ export ORACLE_SID=<DB Name>
[oracle@oda-base-01 ~]$ export ORACLE_HOME=<Home Path>
[oracle@oda-base-01 ~]$ rman
RMAN> connect target sys@<Quell-DB>
RMAN> connect auxiliary sys@<Ziel-DB>
RMAN> startup clone nomount
RMAN> duplicate target database for standby
```

```

from active database
spfile
  set compatible '11.2.0.4.0',
  set db_name '<DB Name>',
  set db_unique_name '<DB Unique Name>',
  set db_create_file_dest
'/u02/app/oracle/oradata/datastore/.ACFS/snaps/<Unique Name>',
  set db_create_online_log_dest_1
'/u01/app/oracle/oradata/datastore/<Unique Name>',
  set db_recovery_file_dest
'/u01/app/oracle/fast_recovery_area/datastore/<Unq. Name>',
  set control_files '',
  set diagnostic_dest '/u01/app/oracle',
  set audit_file_dest '/u01/app/oracle/admin/<Unique Name>/adump',
  set filesystemio_options 'ASYNCH',
  reset [...]
  reset memory_target
  set sga_target '16G',
  set pga_aggregate_target '4G',
noresume
dorecover;

```

Um die Kopie nun laufen aktuell zu halten, wird diese als Archive Destination in der Quell-Datenbank eingerichtet:

```

SQL> alter system set log_archive_dest_2='SERVICE=<Ziel-DB> [...]';
SQL> alter system set log_archive_dest_state_2=ENABLE;

```

Die Ziel-Datenbank wird anschließend per managed Recovery auf dem aktuellen Stand gehalten.

```

SQL> alter database recover managed standby database
2> disconnect from session;

```

Die eigentliche Umschaltung kann dann zu einem beliebigen Zeitpunkt erfolgen. Dazu wird die Quell-Datenbank gezielt gestoppt und das letzte Archivelog übertragen.

```

SQL> shutdown immediate
SQL> startup mount
SQL> alter system switch logfile;

```

Ist dieses letzte Archivelog auf der Ziel-Datenbank eingespielt worden, kann die Datenbank aktiviert werden.

```

SQL> alter database recover managed standby database cancel;
SQL> alter database activate standby database;

```

Damit ist die Umstellung erfolgreich vollzogen. Um den Betrieb der Database Appliance zu gewährleisten, muss die Datenbank nun noch in der Clusterware registriert werden. Dies ist notwendig um ein späteres automatisches Patchen zu gewährleisten. Auch hier ist wieder auf die verschiedenen Namen Acht zu geben.

```

srvctl add database -d <DB Unique Name> -n <DB Name> -i <Instance Name>
\
-o /u01/app/oracle/product/11.2.0.4/dbhome_1

```

```

-c SINGLE -x <ODA Base Server Name>
-p /u02/app/oracle/oradata/datastore/.ACFS/snaps/<Unique Name>/<Instance
Name>/spfile<Instance Name>.ora
-j
/u01/app/oracle/oradata/datastore,/u02/app/oracle/oradata/datastore,/u01
/app/oracle/fast_recovery_area/datastore,/u02/app/oracle/oradata/flashda
ta

```

Am Ende wird diese Ressource deaktiviert und eine eigene Ressource zur Implementierung als Aktiv/Passiv-Cluster zu implementieren. Zur Einspielung von Patches wird dies entsprechend wieder zurück getauscht.

Backup

Die Sicherung der Datenbanken auf der Database Appliance sollen auf ein zentrales Storage erfolgen. Dieses stellt einen DFS Share für die Backups bereit. In der Grundkonfiguration kann das Oracle Linux auf der Database Appliance lediglich einfache SMB Shares mounten. Für einen erfolgreichen DFS Mount werden weitere Pakete benötigt. Diese dürfen jedoch nicht ohne weiteres auf der Database Appliance installiert werden. Hierfür stellt Oracle in MOS Note 1461798.1 ein spezielles Tool bereit, das auf der Database Appliance bereitgestellt werden muss.

```

[root@oda-base-01]$ export http_proxy=192.168.1.1
[root@oda-base-01]$ ./yumsetup
[root@oda-base-01]$ ./getrpm keyutils
[root@oda-base-01]$ rpm -ivh /opt/oracle/oak/pkgrepos/rpms/keyutils-1.2-
1.el5.x86_64.rpm

```

Nachdem die Datei /etc/request-key.conf angepasst wie folgt wurde

```

create cifs.spnego * * /usr/sbin/cifs.upcall -c %k
create dns_resolver * * /usr/sbin/cifs.upcall %k

```

ist auch der DFS Mount erfolgreich.

```

[grid@oda-base-01 ~]$ mount -t cifs //<Domain>/dfs /mnt \
> -o username=xxxx, domain=<Domain>

```

Dies soll im Sinne der einfacheren Handhabung und besserer Performance mittelfristig durch einen NFS Mount ersetzt werden.

Fazit

Aus betrieblicher und monetärer Sicht ist die Konsolidierung auf zwei Oracle Database Appliances ein voller Erfolg. Die nun eingesetzte Hardware wird wesentlich effizienter ausgelastet, bietet eine enorme Performance und genügend Potential für ein weiteres Wachstum. Es sind ausreichend CPU Kerne verfügbar und Optionen wie RAC zur besseren Skalierung sind ebenfalls noch möglich. Die Redundanz innerhalb eines Rechenzentrums war zuvor gar nicht vorhanden und wird nun innerhalb der Database Appliance realisiert. Die Georedundanz war mit der alten Infrastruktur nur mit viel Aufwand bei Hardware und Konfiguration möglich und ist nun Teil des Standard Portfolios.

Die Verwaltung der beiden Database Appliances erfolgt zu großen Teilen über Cloud Control, das ein spezielles Plugin dafür bereitstellt und die Administration damit sehr einfach macht.

Kontaktadresse:

Marco Mischke
Robotron Datenbank Software GmbH
Stuttgarter Str. 29
D-01189 Dresden

Telefon: +49 (0) 351-25859 2884
Fax: +49 (0) 351-25859 3696
E-Mail marco.mischke@robotron.de
Internet: www.robotron.de
Twitter: @MarcoDBA