

# **Konsolidierung mit Exadata – Projekterfahrungen und Best Practices**

**Frank Schneede  
Oracle Deutschland B. V. & Co.KG  
Hannover**

## **Schlüsselworte**

Exadata, Database Machine, Konsolidierung, Best Practice

## **Einleitung**

Die Exadata Database Machine wurde anfangs als reine Datawarehouse-Lösung positioniert. Mit den neueren Generationen erweiterte sich das Einsatzspektrum zuerst auf OLTP Anwendungen, mit größerem Hauptspeicher und Flash Technologie konnten auch große Datenbanken (VLDB) und Datenbank Konsolidierungsprojekte abgebildet werden. Mit der neuen Exadata X3 und dem in der neuen Exadata Software unterstützten Exadata WriteBack Flash Cache lassen sich auch OLTP Anwendungen, die sich durch besonders hohe Schreib-Transaktionslasten auszeichnen, problemlos auf Exadata betreiben.

Dieser Vortrag stellt ein Exadata Konsolidierungsprojekt vor und beschreibt die Best Practises, die bei der Nutzung von Exadata als Konsolidierungsplattform eingehalten werden sollten. Die Beweggründe, die zur Entscheidung für Exadata als Konsolidierungslösung geführt haben, werden ebenso dargestellt, wie die Technologien, die auf Exadata als Konsolidierungsplattform eingesetzt werden.

## **Effizienzsteigerung in der IT durch Konsolidierung auf Exadata**

Gewachsene IT Umgebungen haben häufig *Silocharakter*, das heisst, dass jede Anwendung über eine eigene Infrastruktur („*Silo*“) verfügt und separat administriert wird. Jedes dieser *Silos* ist so ausgelegt, dass die maximale Last der Anwendung in Stoßzeiten bearbeitet werden kann. Die Infrastruktur ist relativ unflexibel, da ungenutzte Ressourcen nicht ohne weiteres anderen Anwendungen zugeordnet werden können. Die Einführung neuer Anwendungen ist jedes Mal ein eigenes Projekt, in dem die gesamte Architektur konzipiert, eingerichtet und getestet werden muss. Die Folge sind lange Projektlaufzeiten – von der ersten Planung bis zum Produktionsstart vergehen erfahrungsgemäß oft Monate. Die lange Laufzeit entsteht dadurch, dass die Lieferanten der unterschiedlichen Komponenten sich sowohl untereinander als auch mit den verantwortlichen Bereichen der IT Abteilung abstimmen müssen. Auftretende Probleme führen zu *Finger-Pointing*, also gegenseitigen Schuldzuweisungen der beteiligten Parteien. Die Lösungsfindung verzögert den Projektfortschritt in diesem Fall erneut. Allerdings bietet ein derartiger Architekturansatz auch einen Vorteil. Die Kosten jeder Applikation sind leicht zu ermitteln und im Zuge einer kundeninternen Leistungsverrechnung zuzuordnen.

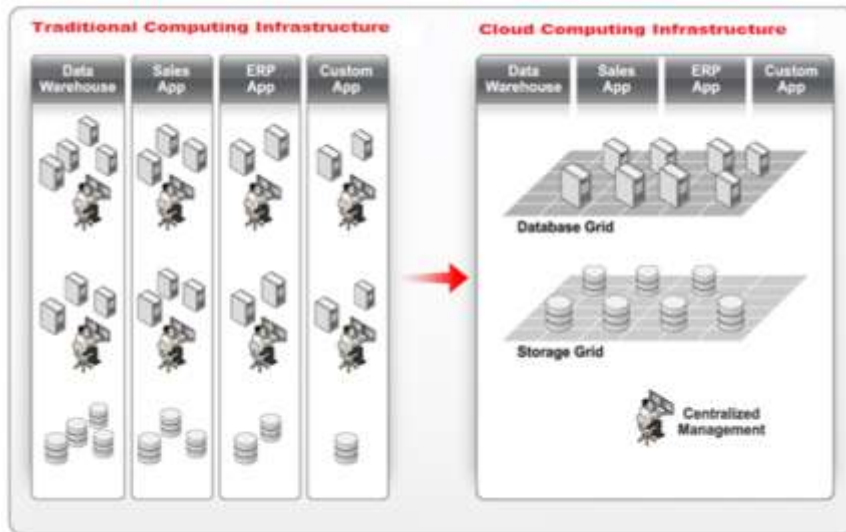


Abb. 1: Herkömmliche „Silo“-IT Struktur vs. Exadata

Eine Konsolidierungsplattform wie Exadata bietet hingegen durch die für Datenbank und Storage konsequent umgesetzte Grid Architektur den Vorteil, dass Ressourcen zu jeder Zeit bedarfsgerecht verteilt werden können. Die Komplexität der IT Struktur ist durch konsequente Standardisierung der eingesetzten Hardware und Software deutlich reduziert und dem Endkunden können bessere Service Level angeboten werden. Eine Verrechnung der Leistungen an die Nutzer einer auf der Exadata Database Machine konsolidierten IT gestaltet sich allerdings etwas schwieriger, da die Kosten der Konsolidierungsplattform nach unterschiedlichen Nutzungskriterien verursachungsgerecht erfolgen sollte.

Eine Umstellung einer herkömmlichen IT Architektur auf Exadata stößt besonders in großen Unternehmen auf Widerstände, da die IT aus mehreren, historisch gewachsenen Bereichen besteht. So gliedert sich die IT beispielsweise auf in

- DB Competence Center / DB Betrieb
- Netzwerksteuerung
- Storageverwaltung
- Systemverwaltung / OS Betrieb
- Anwendungsunterstützung
- 

In derart aufgesplitteten IT Bereichen erfordert die Konsolidierung auf Exadata zwar nicht zwingend eine Umordnung von Aufgaben, allerdings bestehen seitens der betroffenen Bereiche zum Teil erhebliche Vorbehalte gegenüber einer vertikal integrierten Lösung wie Exadata. Unternehmen, die die Komplexität herkömmlicher IT Architekturen und die daraus resultierende Unflexibilität erkannt und daher Exadata als Konsolidierungsplattform eingeführt haben, berichten von extrem kurzen Projektlaufzeiten und einer erheblichen Effizienzsteigerung der IT. Dieses bezieht sich sowohl auf die Einführung und Migration als auch auf den Betrieb der Exadata. (siehe hierzu auch folgende aktuelle Untersuchung der FactGroup: <http://www.oracle.com/us/products/database/exadata-vs-ibm-1870172.pdf>)

In kleineren, mittelständischen IT Abteilungen stellt sich die Situation allerdings oft anders dar. Die IT ist häufig eine Art „One-Man-Show“, in der mehrere Systeme von einer sehr kleinen Mannschaft betreut werden. Das führt zu der unerwünschten Situation, dass die IT Systeme nur noch notdürftig am

Leben gehalten und nicht weiterentwickelt werden können, da ein ausgeprägter Mangel an Ressourcen vorliegt. In dem hier vorgestellten Projekt eines deutschen Kunden war der Wunsch nach einer einheitlichen Plattform für insgesamt 5 Produktions und 4 Testdatenbanken ausschlaggebend für die Entscheidung, Exadata als Konsolidierungslösung zu wählen.

### Sorgfältige Planung als Basis

Um eine problemlosen Umstieg auf Exadata sicherzustellen, wurde die Migration durch Oracle Consulting vorbereitet. Bevor die eigentliche Migration erfolgen kann, sind die Konfigurationsparameter der Exadata (Namen, IP-Adressen, ...) in Zusammenarbeit zwischen Oracle ACS und dem Kunden festgelegt worden. Zentrales Werkzeug war dabei bislang das Configuration Worksheet, das Bestandteil der Dokumentation war. Abbildung 2 zeigt einen Teil des Worksheets.

Table 3 General Configuration Worksheet

Item	Entry	Description and Example
Operating system	Linux	The operating system for the database servers. Valid value: Linux or Solaris
Oracle Exadata Database Machine name	DM01	This name is the basis for all host names generated during installation. Select a value of 4 characters or fewer. Example: dm01
Type of system	X2-2 Quarter Rack	Type of Oracle Exadata Database Machine being installed at your site. Valid values: X2-2 Full Rack, X2-2 Half Rack, X2-2 Quarter Rack, or X2-8 Full Rack
Disk type	High Performance	Type of hard disks in Oracle Exadata Database Machine. Valid values: High Performance or High Capacity
Country	Germany	Country where Oracle Exadata Database Machine will be installed. Example: United States

Abb.2: Exadata Konfiguration im Configuration Worksheet

**Die Vorbereitung der Exadata Installation hat sich kürzlich geändert!** Seit Mitte Oktober 2012 wird die Konfiguration der Exadata Database Machine mit dem **Oracle Exadata Deployment Assistant** durchgeführt. Der Deployment Assistant ist im aktuellen OneCommand Utility (Patch 14617927) enthalten. Die aktualisierte Dokumentation (Patch 10386736) beschreibt den Umgang mit dem Tool.

Der Oracle Exadata Deployment Assistant ist ein Java-Programm, das unabhängig von der Exadata installiert und zum Beispiel auf Windows Plattform genutzt werden kann. Abbildung 3 zeigt einen Screenshot der Anwendung. Der Assistant kann für alle Exadata X2 und X3 Modelle, Expansion Racks und SuperCluster verwendet werden. Es werden nach Abschluss der Konfiguration die

notwendigen XML-Konfigurationsdateien generiert. Die Konfigurationsdateien sind Basis der intialen Exadata Konfiguration

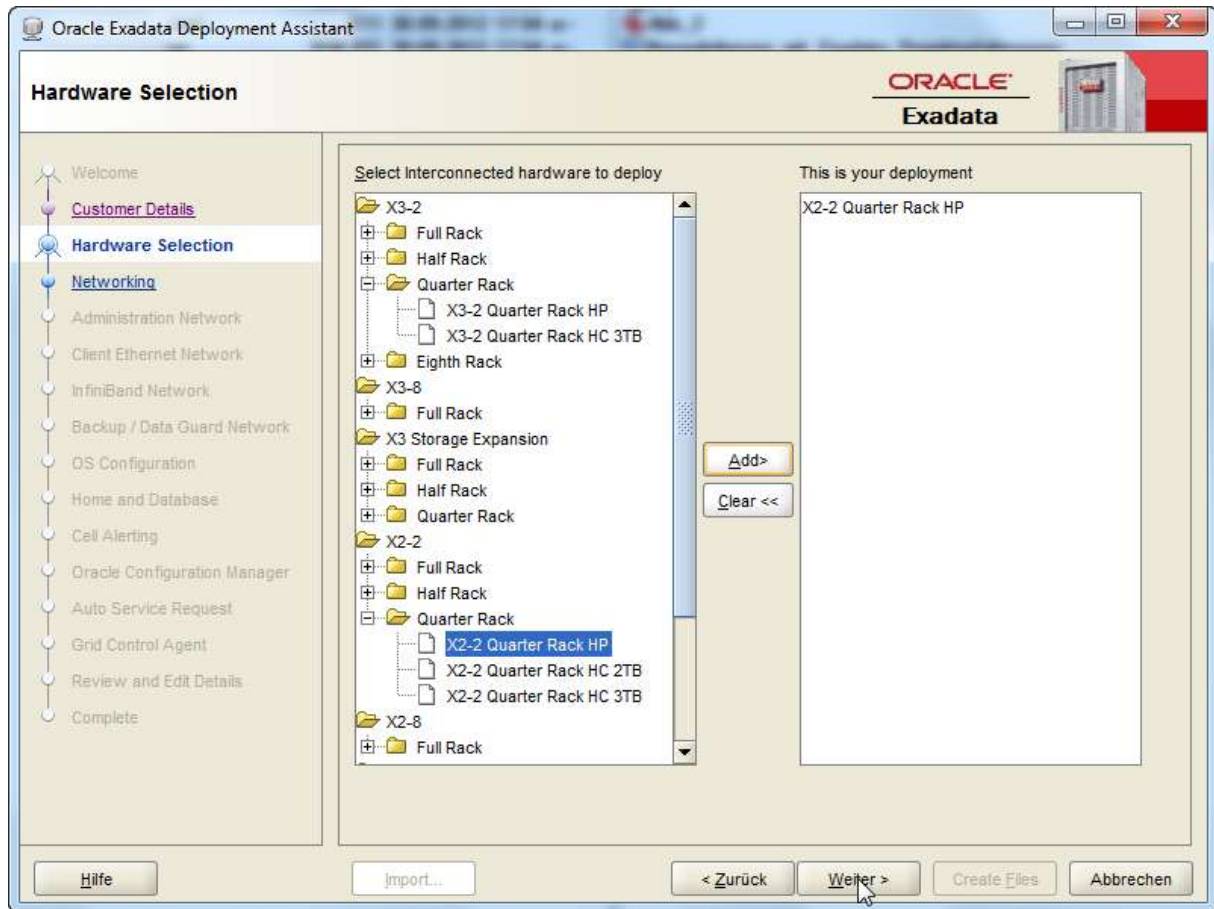


Abb.3: Oracle Exadata Deployment Assistant

Das Konsolidierungsprojekt lief in mehreren Phasen ab:

1. Aufnahme der Ist-Umgebungen
2. Installation der Datenbanken auf Exadata-Zielsystem
  - a. Clusterware Konfiguration
  - b. Verifikation der Umgebung, Komponenten- und Ausfalltests im Clusterumfeld (ASM, Clusterware, Datenbank)
  - c. Swingbench Lasttests
3. Migration
  - a. Migrationskonzept
  - b. Durchführung von Probemigrationen
  - c. Migration mit DataPump, ggf via Database Links
4. Funktions- und Lasttests durch die Applikationen
5. Inbetriebnahme durch Migration des Produktivdatenbestandes

### **Aufnahme der Ist-Umgebung**

Um die Ist-Umgebung aufzunehmen, wurden Reports mit dem RDA (Remote Diagnostic Agent) erstellt. Hierzu sollte die aktuelle Version aus My Oracle Support geladen werden. Die wichtigsten Informationen, die aus den Reports entnommen wurden, sind:

- Zeichensatz
- Installierte Komponenten (DBA\_REGISTRY)
- Parameter, die ggf. nach 11g zu „vererben“ sind
- Größe und Anzahl der Tablespaces, Schemata mit ihren Eigenschaften - nur informell, denn diese Informationen werden aus dem Quellsystem generiert und auf dem Zielsystem dann angewendet.

Nach den RDA-Reports ist es sinnvoll, einen RDA-Healthcheck auf dem Quellsystem Server durchzuführen. Siehe hierzu [MOS Note 250262.1: RDA 4 – Health Check / Validation Engine Guide](#) . Dieser Check dient dazu, Konfigurationsparameter zu ermitteln, die gegebenenfalls die Systemstabilität negativ beeinflussen können.

Ein weiteres Skript gibt Aufschluss über den Zustand der Datenbanken. Siehe hierzu [MOS Note 136697.1: „hcheck.sql“ script to check for known problems in Oracle8i, Oracle 9i, Oracle 10g, Oracle 11g](#). Das Skript prüft das Data Dictionary und die Anwendungsschemata nach Inkonsistenzen und gibt konkrete Handlungsanweisungen zu deren Beseitigung. Die Beachtung und Durchführung der Handlungsanweisungen ist wesentlich für den Erfolg der späteren Migration!

Zum Abschluss wurden AWR Reports von signifikanten Zeiträumen gezogen, um später den eine Vorher/Nachher Analyse vornehmen und so den Erfolg der Migration bewerten zu können.

### **Installation der Datenbanken**

Nach der Übergabe der Exadata Database Machine an den Kunden wurden die Datenbanken aufgesetzt. Das geschah mit dem Database Creation Assistant (DBCA). Aus der ersten Datenbank-Installation wurde ein Template erzeugt, das als Vorlage für die weiteren aufzusetzenden Datenbanken genutzt wurde. Dadurch konnte eine Datenbank Standardkonfiguration für alle 9 Datenbanken implementiert werden. Kleinere applikationsspezifische Anpassungen der Datenbanken wurden im Nachgang durchgeführt.

### **Migration auf Exadata**

Für die Migration wurde DataPump verwendet, da die Datenmengen dies zuließen und der Kunde mit der Technologie vertraut war. Zudem bietet dieses Vorgehen die Möglichkeit, Aufräumarbeiten durchzuführen. Folgende Arbeitsschritte waren notwendig:

1. Anlegen der Tablespaces wie im Quellsystem
2. QS im Quellsystem: Rekompilieren aller Objekte, entfernen überflüssiger Objekte
3. DataPump via Database Link auf Schemaebene
4. QS auf dem Zielsystem und Behandlung der Fehlermeldungen im DataPump

Für die Durchführung der Migration ist es sinnvoll, die einzelnen Schritte so weit wie möglich zu automatisieren und vor allem ausführlich zu testen. Die Automatisierung lässt sich sehr gut über Script-Generatoren erreichen. Das folgende Skript `genTablespaces.sql` wurde für jede Datenbank ausgeführt und manuell angepasst.

```

set serveroutput on
set lines 200
set feedback off

declare
mydbname varchar2(30);
cursor c1 is
select a.tablespace_name mytbs,
       CEIL(sum(bytes)/(1024*1024)) mysize
  from dba_data_files a, dba_tablespaces b
 where a.tablespace_name = b.tablespace_name
       and a.tablespace_name not in ('SYSTEM', 'SYSAUX')
       and b.contents NOT IN ('UNDO','TEMPORARY','PERFSTAT')
       and b.contents NOT IN ('RBS','TB_RBS')
 group by a.tablespace_name;

begin
select name into mydbname from v$database;
dbms_output.put_line ('-- database ' || mydbname);
dbms_output.put_line ('connect sys/Oracle2012@' ||
                      mydbname || ' as sysdba');

for i in c1 loop
  dbms_output.put_line ('create tablespace ' ||
                        i.mytbs || ' datafile ''+DATA'' size ' ||
                        i.mysize || 'M AUTOEXTEND ON MAXSIZE UNLIMITED;');
end loop;
end;
/

```

Da die Umgebung des Kunden überschaubar war, wurden die generierten Skripte manuell überarbeitet. Eine vollständige Automatisierung wäre aufwändig zu erstellen und brächte in der Situation keinen Vorteil. Anpassungen der generierten Skripte erfolgten primär im Bereich der Storage Klauseln, der Generator summiert die Tablespacegröße einfach auf, was bei der verwendeten Blockgröße von 8k zum Teil die Grenze von 32GB für ein Datafile überschritt. Entsprechend wurden hier manuell mehrere Datafiles angelegt.

Die QS Maßnahmen, die auf dem Quellsystem durchgeführt werden mussten, dienten dazu, dass DataPump fehlerfrei ausgeführt werden konnte. Ungültige Objekte wurden rekompiliert und – falls das nicht erfolgreich möglich ist – schriftlich festgehalten. Ein kleines Skript half dabei:

```

SQL> sta ?/rdbms/admin/utlrp -- recompile

spool $HOME/scripts/objects.lst
select owner, object_type, count(*)
  from dba_objects
 group by owner.object_type
 order by 1,2;

select owner, object_type, object_name
  from dba_objects
 where status = 'INVALID'
 order by 1,2,3;

```

Zusätzlich wurden sämtliche Datenbank Schemata überprüft, ob der Default-Tablespace auf dem Quellsystem vorhanden ist, da ansonsten DataPump das Schema nicht exportieren kann.

Für die Datenübernahme wurde DataPump gewählt. Da der Platz für die zu übertragenden Dumpfiles nicht ausreichte, wurde der Import über Database Links durchgeführt. Zur Nutzung von DataPump als Migrationstool ist die [MOS Note 1264715.1: Master Note for Data Pump](#) hilfreich. Folgende Themen mussten bei der Migration über Data Pump aufbereitet werden:

- Database Links der Applikationen
- Materialized Views
- Jobs in der Datenbank
- Dateien im Filesystem der XML DB
- Tabellen mit Datentypen LONG, LONG RAW und ANYTYPE
- DB Public Objekte (Synonyme, DB Links)
- VIEW Kompilierungsfehler

Der Import wird schließlich in drei Schritten ausgeführt:

1. Migration der Schemata mit Tabellen und Daten, Dokumentation der Ausnahmen (EXCLUDE....) und Fehler.
2. Migration von Tabellen mit LONG / LONG RAW Datentypen
3. Migration von REF Constraints

Im ersten Schritt wurden nur alle Daten übertragen, lediglich Berechtigungen, Statistiken, Kommentare, Database Links und REF Constraints wurden ausgeschlossen. Die Berechtigungen wurden nachgezogen, nachdem alle Daten importiert worden sind. Zum Abschluss wurden alle Objekte kompiliert. Die genannten Schritte erfolgten automatisiert, das Ergebnis wurde in Logfiles dokumentiert.

Der zweite Schritt diente der Übertragung von Tabellen, die LONG / LONG RAW Datentypen beinhalten. Die betreffenden Tabellen wurden manuell kopiert. Abschließend wurden noch die REF Constraints erstellt. Die Überwachung und Qualitätssicherung erfolgte über Logfiles bzw. manuelle Überwachung der DataPump Jobs.

Damit war die eigentliche Migration abgeschlossen. Mehrfache Testläufe vor Produktionsbeginn halfen, eventuell vorhandene Probleme auszumerzen und so die Qualität für die Produktionsmigration zu sichern.

### **Nachgelagerte Schritte**

Das vorgestellte Konsolidierungsprojekt ist relativ klein und überschaubar. Es erfolgte eine 1:1 Migration auf die neue Exadata Plattform. Es wurden in den Anwendungen und Datenstrukturen keine Exadata spezifischen Optimierungen durchgeführt. Erste Analysen haben gezeigt, dass im ersten Schritt keine weiteren Optimierungsschritte notwendig sind, da die Exadata für das prognostizierte Transaktionsvolumen mehr als ausreichend ist. Sollten später weitere Datenbanken auf die Exadata gebracht werden, so stehen Features wie

- I/O Resource Manager
- Instance Caging
- Quality of Service Monitoring

zur Verfügung, um auftretende Engpässe nachhaltig zu beheben.

### **Zusammenfassung**

Das vorgestellte Projekt zeigt die gelungene Konsolidierung einer gesamten IT Umgebung auf einer Exadata Database Machine. Das gewählte Vorgehen wurde unter Berücksichtigung der [MOS Note 757552.1: Oracle Exadata Best Practices](#) und dem [Oracle Whitepaper \(November 2011\): Best Practises for Database Consolidation On Exadata Database Machine](#) den Erfordernissen des Projektes angepasst. Nach der 1:1 Migration der Datenbanken werden weitere Optimierungsmaßnahmen wie die Nutzung von Exadata spezifischen Funktionen und die K-Fall Absicherung mittels Data Guard evaluiert und bedarfsorientiert umgesetzt.

Die Exadata Database Machine ist, insbesondere durch die Modifikationen der aktuellen Modellgeneration, als Konsolidierungsplattform besonders gut geeignet. Während reine Konsolidierungsthemen im Umfeld deutscher Kunden noch relativ rar sind, gibt es international durchaus Erfahrungen mit zum Teil sehr großen Konsolidierungsprojekten auf der Exadata Database Machine. Auf der Oracle Open World 2012 wurde in diesem Zusammenhang das Projekt des Navigationsgeräteherstellers Garmin vorgestellt. Details hierzu sind in den Weiterführenden Informationen zu finden.

### **Weiterführende Informationen:**

<http://www.oracle.com/technetwork/database/focus-areas/performance/instance-caging-wp-166854.pdf>

<http://www.oracle.com/technetwork/database/focus-areas/performance/resource-manager-twp-133705.pdf>

<http://www.oracle.com/technetwork/database/features/availability/exadata-consolidation-522500.pdf>

<http://www.oracle.com/technetwork/database/features/availability/xmigration-11-133466.pdf>

<http://www.oracle.com/technetwork/database/availability/garmin-1667151.pdf>

<http://www.oracle.com/us/corporate/customers/customersearch/garmininternational-1-exadata-ss-1561598.html>

### **Kontaktadresse:**

Frank Schneede  
Oracle Deutschland B. V. & Co. KG  
Thurnithstr. 2  
D-30519 Hannover

Telefon: +49 (0) 511 95787 - 250  
Fax: +49 (0) 511 95787 - 100  
E-Mail: frank.schneede@oracle.com  
Internet: www.oracle.com