

# OBIEE und Exadata

-

## Performance Tuning Projekterfahrung

Moritz Werning  
sumIT AG  
Schweiz

### Schlüsselworte

Performance Tuning, Oracle Business Intelligence Suite Enterprise Edition, OBIEE, OBI, Analytics, RPD, Exadata

### Einleitung

Die Kombination aus Oracle Business Intelligence Suite Enterprise Edition 11g (OBIEE) und Oracle Exadata Database Machine bietet eine sehr gute Performance, trotzdem gibt es Möglichkeiten, diese Performance noch weiter zu verbessern.

Dieses Manuskript diskutiert, wie OBIEE so konfiguriert werden kann, dass die Oracle 11gR2 Datenbank Features im Allgemeinen und Oracle Exadata Database Machine Ressourcen im Speziellen bestmöglich genutzt werden können.

Anhand der Erfahrungen, die wir im letzten Dreivierteljahr in einem Migrationsprojekt von einer IBM Hardware auf Oracle Exadata Database Machine gemacht haben, wird dieses Manuskript zeigen, welche verschiedenen Aspekte zu beachten sind, welche Optionen OBIEE bietet und wie diese genutzt werden können, um eine optimale Abfrageperformance zu erreichen. Das Manuskript zeigt, wie OBIEE auf verschiedenen Ebenen (Global, Objekt und Analyse) konfiguriert werden kann, um für die verschiedenen Abfragekategorien die beste Performance zu erreichen und die Kommunikation zwischen OBIEE und der DB zu optimieren. Dabei werden auch die bestehenden Grenzen und möglichen Probleme einer solchen Optimierung aufgezeigt.

Dieses Manuskript gibt BI Entwicklern praktische Tipps, wie Performanceprobleme analysiert werden können und versetzt sie zusätzlich in die Lage, die OBIEE Umgebung bestmöglich zu konfigurieren.

Gleichzeitig werden Vorschläge gemacht, wie in Zusammenarbeit mit den DBAs auch die DB optimal für die Anforderungen des OBIEE angepasst und der DB-Workload am besten verwaltet werden kann, um dem BI Anwender eine überragende Abfrageperformance zu liefern.

### Grundlagen

Um Performance Probleme zu analysieren und nach Möglichkeit auch beheben zu können, ist es wichtig, die beteiligten Komponenten zu kennen. Man sollte wissen, wie diese Komponenten interagieren und welches mögliche Ursachen von Performance Problemen sind.

Nur mit diesem Wissen ist es möglich, an der richtigen Stelle Anpassungen vorzunehmen, um die Performance zu verbessern.

Der OBIEE Anwender interagiert per Webbrowser mit einer Dashboardseite im Weblogic Plugin „Analytics“, welches wiederum auf dem BI Presentation Server basiert. Der BI Presentation Server generiert für Analysen und Prompts Abfragen an den BI Server in logisches SQL.

Der BI Server erstellt daraus physische SQLs je nach Datenquelle. Diese physischen SQLs werden über die im RPD definierte Connection vom BI Server gegen den definierten DB Server ausgeführt.

Der DB Server schickt die Ergebnisse an den BI Server zurück, dieser verarbeitet diese Ergebnisse weiter (Aggregation, Stich Joins). Die vom BI Server verarbeiteten Ergebnisse werden an den BI Presentation Server übergeben, welcher die Ergebnisse graphisch aufbereitet und durch den Webserver an den Webbrowser des Anwenders übermittelt.

Sowohl der BI Presentation Server als auch jede BI Server Instance verfügen über lokale Caches. In einer Cluster Umgebung gibt es auch noch einen globalen Cache. Bei Abfragen wird zunächst versucht, die Abfrage aus dem Cache zu beantworten. Auch die Oracle Datenbank verfügt über einen Result Cache.

Die Anwender Abfrageperformance, also die Zeit, die ein Anwender warten muss, bis er die Ergebnisse einer Dashboardseite bzw. die Ergebnisse der darauf enthaltenen Analysen und Prompts in seinem Webbrowser sieht, wird durch sehr viele Faktoren beeinflusst:

- Aufbau der Dashboardseite (Kombination von Analysen und Prompts)
- Aufbau der darin enthaltenen Analysen
- RPD Datenmodell und Dimension Level Anzahl an Elementen
- Physische Datenbankstrukturen
- Objekt Statistiken
- Caches
- Systemlast
- Konfiguration des BI Servers und des BI Presentation Services
- Konfiguration der Fusion Middleware
- Betriebssystem Konfiguration und Hardwareausstattung vom BI Server Host
- DB System und Netzwerk

### **Ausgangssituation beim Kunden**

#### **Infrastruktur vor der Migration auf Exadata**

Die Situation beim Kunden vor der Migration auf Exadata war folgende:

- Oracle Database 11gR1 (ohne OLAP und Essbase)
- Kern Applikation und DWH auf der gleichen IBM Hardware
- IBM Host am Limit
- Hardware nicht auf DWH Anforderungen ausgelegt
- I/O Durchsatz < 1 GB/s
- Wachsende Anforderungen von allen Systemen
- Performance von ETL und Abfragen nicht zufriedenstellend

#### **Infrastruktur nach Migration auf Exadata**

Der Kunde erwarb bei Oracle zwei Exadata *half racks* für eine dedizierte DWH Umgebung. Diese wurden an zwei verschiedenen, entfernten Standorten des Kunden installiert und eine Disaster Recovery Strategie geplant. Im Zuge der Migration auf Exadata wurde auch auf Oracle 11gR2 migriert, welches, wie auch die Exadata DB Maschine, viele neue Features bietet, die eine höhere Performance liefern.

Die Tests auf der neuen Infrastruktur waren sehr vielversprechend. Es konnte ein I/O Durchsatz (7 storage cells) von 10.8 GB/s gemessen werden. Insgesamt konnte die Dauer der täglichen ETL Verarbeitung deutlich verkürzt werden und zusätzlich wurde die Abfrageperformance bis zu einem

Faktor von 50 schneller. Aber es gab auch Ausnahmen, etwa SQLs, welche sehr CPU intensiv waren und von dem massiv schnelleren I/O Durchsatz nicht direkt profitieren konnten und zunächst an die neue Infrastruktur angepasst werden mussten. Hinzu kam der Umstieg von einer Single Instance DB in eine RAC Umgebung mit den damit verbundenen Anpassungsmöglichkeiten. Zusätzlich gab es weitere Optimierungsmöglichkeiten durch den Einsatz neuer Oracle 11gR2 und Exadata Features.

Die OBIEE Infrastruktur besteht OBIEE 11.1.1.5 Installationen auf IBM Hosts unter Solaris. Eine Migration auf OBIEE 11.1.1.6. ist geplant.

### **Physische Datenquellen**

Die physische Quelle für das OBIEE Datenmodell ist ein relationaler Datamart/Access Layer in einem Schema der Oracle DB. OLAP und Essbase sind vom Kunden nicht lizenziert und werden nicht eingesetzt. Der Datamart Layer besteht aus vielen verschiedenen Marts mit *shared* Dimensionen. Bei dem Aufbau des Datamart Layer und der Modellierung wurde entsprechend der Oracle Bestpractice für Datamarts vorgegangen.

Das Datenmodell basiert auf einem Star Schema. Für die Abfragen und die OBIEE Modellierung Abfragen bieten Star Schema große Vorteile. Das BI Modell und die Modellierung im Administration Tool ist deutlich einfacher als etwa die Modellierung auf einem 3NF Schema. Abfragen benötigen deutlich weniger Joins und es können die speziellen DB Performance Features wie Star Transformation und Query Rewrite genutzt werden.

Die Faktentabellen sind partitioniert, in den meisten Fällen Range (Intervall) nach Zeit. Teilweise wurde auch noch eine Hash-Subpartitionierung eingesetzt. Die Tabellen verfügen über Bitmap Indices und Fremdschlüssel zu den Primärschlüsseln der referenzierten Dimensionen. Die Informationen der Dimensionstabellen werden durch die Metadaten über Beziehungen und Hierarchien in den zugehörigen Dimensionen ergänzt, die dem Optimizer hilfreiche Informationen für ein mögliches Query Rewrite liefern.

Die DB Objekt Statistiken werden nach jeder ETL Verarbeitung inkrementell und mittels *AUTO\_SAMPLE\_SIZE* gerechnet.

### **Eingesetzte Systemüberwachung**

Die Systemüberwachung ist ein wichtiger Teil des Performance Tunings. Sie gibt Hinweise auf allgemeine Performance Probleme und Probleme in einzelnen Systemkomponenten.

In diesem Projekt werden von uns verschiedene Techniken zur Systemüberwachung und zur Analyse der Auswirkungen von Performance Tunings eingesetzt.

Innerhalb von Analytics selbst kann von BI Administratoren auf das Session Management zugegriffen werden. Dort gibt es eine Übersicht über offene BI Presentation Service Sessions, deren Laufzeiten sowie bei entsprechendem Loglevel auch Zugriff auf das NQSQuery.log. Aus Performancegründen ist der Loglevel für Standardanwender aber 0, so dass im Normalfall keine Informationen aus dem NQSQuery.log sichtbar sind.

Bewährt haben sich auch der Einsatz des BI Usage Trackings und die Auswertung der Usage Daten. Dabei könnten z.B. Alerts eingerichtet werden, welche den BI Entwickler informieren, wenn es zu Performanceproblemen kommt.

Die Fusion Middleware Control und die Administration Console bieten eine ganze Reihe von

Metriken, welche Aufschluss über die einzelnen Systeme und deren Auslastung und Performance geben.

Beim aktiven Test wird das in EM DB Console und *Enterprise Manager Grid Control* enthaltene SQL Monitoring eingesetzt.

Eine sinnvolle Ergänzung, welche beim Kunden derzeit noch nicht eingesetzt wird, ist das *Oracle Business Intelligence Management Pack*, welches den *Enterprise Manager Grid Control* auf OBIEE ausweitet.

## **Testzyklus**

Beim Performance Tuning und Performance Tests hat es sich bewährt, im Vorhinein ein klares Vorgehen zu definieren. Zunächst sollte das Problem analysiert und dann das Ziel des Tests definiert werden. Dafür sollte zunächst das Problem eingegrenzt werden und eine Aufnahme des IST-Zustandes erfolgen. Das heißt, die Antwortzeiten und die Systemlast sollten gemessen und die Anwendererwartungen erfasst werden. Welche Antwortzeiten sind schnell und ab wann empfindet der Anwender die Antwortzeiten als zu langsam? Hilfreich ist es, Anwenderbeschwerden zu hinterfragen; nicht immer entsprechen die geschilderten Erfahrungen und Interpretationen den Tatsachen.

Nach dieser ersten Analyse sollte ein klares Ziel definiert werden, z.B.: Die Antwortzeit von Dashboardseite S ist zwei Minuten und soll nicht länger als zehn Sekunden sein. Zunächst muss also die Ursache für die lange Antwortzeit gefunden werden. Diese Ursache kann in allen die Antwortzeit beeinflussenden Faktoren (siehe Grundlage) liegen.

Aus Erfahrungswerten können hier meist schon gewisse Annahmen getroffen werden. Wenn es darum geht, Kundenbeschwerden möglichst schnell zu beheben, ist es meist am schnellsten, Top Down vorzugehen und zu versuchen, die Ursache zu isolieren. Dabei können alle Informationen aus der Systemüberwachung genutzt werden. Eine andere Möglichkeit ist das Bottom Up Vorgehen, welches aber meist aufwändiger ist.

Wichtig ist es, möglichst schnell zu identifizieren, ob das Problem auf Datenbankebenen zu finden ist oder auf BI Server Ebene.

Sobald z.B. eine Analyse und daraus resultierende physische SQL als Ursache für die langsamen Antwortzeiten ausgemacht wurde, kann weiter getestet werden.

- Ist BI Datenmodell optimal?
- Ist das physische SQL so, wie es sein sollte?
- Ist der Execution Plan des physischen SQL optimal?
- Wie kann der Execution Plan optimiert werden?

Dabei sollte das Ziel und der Test immer wieder an die neuen Erkenntnisse angepasst werden und die Performance Tuning Schritte eingeleitet werden.

## **Performance Tuning**

Performance Tuning kann man auf verschiedenen Ebenen durchführen. Manche Tuning Maßnahmen können vom BI Entwickler selbst durchgeführt werden, für andere ist die Zusammenarbeit mit dem DBA oder den Systemadministratoren sinnvoll oder notwendig. Es hat sich gezeigt, dass eine gute Zusammenarbeit von Datenbankentwicklern und BI Entwicklern sehr wichtig ist.

## **Tuning auf DB Ebene**

Auf Datenbankebenen können verschiedene Maßnahmen durchgeführt werden, die dazu führen, dass physische SQLs schneller ausgeführt werden und der DB-Workload auf der Datenbank besser verwaltet wird.

### **DB Workload Management – DB Services**

In Zusammenarbeit mit den DBA sollte ein Resource Plan Management eingerichtet werden oder bestehende Ressourcenpläne um Regeln für den BI Server erweitert werden. Der BI Server sollte sich über einen (oder mehrere) dedizierte DB Services verbinden, um die Einteilung und Auswertung möglichst einfach zu machen. Die DB Services bieten auch Möglichkeiten, den BI Server DB-Workload auf einige der Exadata RAC Nodes zu beschränken. Im Ressourcenplan sollte definiert werden, wie viel von den Ressourcen die Gruppen oder die Gruppe des BI Servers verwenden dürfen.

Über die *Ressourcenzuweisung* ist es sinnvoll, „Max. Auslastungsgrenzwert“ sowie prozentuale Werte für die verschiedenen priorisierten Ressourcen Gruppen einzurichten.

Im Abschnitt *Anweisungswerte* bietet es sich an, Dinge wie den „Max. Parallelitätsgrad“, „Max. Prozentsatz von Parallel Server-Ziel“ usw. einzuschränken.

Im Abschnitt *Schwellenwerte* hat es sich sehr bewährt, Grenzwerte für die Ausführungszeit (s) zu definieren. Denn gerade bei der Report Entwicklung durch unerfahrene Reportersteller werden oft physische SQL auf der DB ausgeführt, die zu sehr großen Laufzeiten führen und gar nicht beabsichtigt waren.

### **Degree of Parallelism (DOP)**

Parallelität ist eine sehr einfache Möglichkeit, die Ausführungszeit von SQLs zu beschleunigen. Es ist aber Vorsicht geboten. Je grösser der DOP einer Abfrage, desto weniger Parallel Server stehen für andere Abfragen zur Verfügung und desto höher ist die Systemlast. Das Abschätzen des DOP ist ein schwieriges Problem, da sich die Datenmengen je Abfrage und über die Zeit ändern und die Last auf dem System variabel ist.

Zum Glück gibt es mit Oracle 11gR2 das Feature Auto DOP, welche zusammen mit *parallel statement queuing* und *in memory parallel processing* große Vorteile bietet.

Es bietet sich an, dieses Feature einzusetzen, jedoch sind ausführliche Tests in Zusammenarbeit mit dem DBA sinnvoll, um sicher zu stellen, dass Auto DOP auch funktioniert.

### **Compression**

Table Compression, vor allem die Oracle Exadata HCC (Hybrid Columnar Compression) bietet im DWH Umfeld große Vorteile. Die Datenmengen, welche für Abfragen gelesen werden müssen, werden deutlich kleiner und die Ausführungszeiten kleiner.

## **Tuning auf OBIEE Ebene**

Auf OBIEE Ebene können verschiedenste Massnahmen durchgeführt werden, welche die Antwortzeiten der Anwender verbessern. Die Massnahmen reichen von globalen Konfigurationsanpassungen im Webserver, über das Setzen von Hints auf Physical Tables im RPD bis hin zur Anpassung von Dashboardseiten.

## **OBIEE Caches**

Der OBIEE verfügt mit dem BI Presentation Server Cache und dem BI Server Cache über zwei lokale Caches, welche Datei basiert sind. Diese Abfragecaches verkürzen die Antwortzeiten für die meisten Abfragen massiv. Es ist in jedem Fall angeraten, die Caches zu aktivieren und zu konfigurieren. Eine RAM-Disk als Ablage für die Cache Dateien ist sinnvoll.

Zusätzlich sollte eine Intelligente Cache Strategie entworfen und umgesetzt werden. Dabei sollte sinnvolles Cache Seeding nach wichtigen ETLs z.B. MEVs durchgeführt werden. Die wichtigsten Dashboardsseiten sollten sich für die wichtigsten Prompt Einstellungen im Cache befinden. Das logical SQL zu den Dashboardsseiten kann etwa aus dem Usage Tracking extrahiert werden und z.B. per NQCMD in den Cache gefüllt werden.

## **Aggregat Tabellen auch mit Exadata**

In diesem Projekt hat sich gezeigt, dass selbst mit dem großartigen I/O Durchsatz der Exadata Aggregat Tabellen sinnvoll und notwendig sind, um sinnvolle Antwortzeiten zu erreichen.

Dabei hat sich gezeigt, dass bei den teilweise komplexen physischen SQL, die der BI Server generiert, Query Rewrite auf Materialized Views nicht zuverlässig funktioniert. Aus diesem Grund bietet sich die Nutzung *Aggregate Persistence Wizards* oder die Einbindung manuell erstellter und gewarteter Tabellen oder MViews in das BI Datenmodell an.

## **Fusion Middleware Konfiguration und OBIEE Konfiguration**

Eine grosse Anzahl von Performance kritischen Konfiguration Einstellungen lassen sich in der Fusion Middleware Control und dem BI Server NQSConfig.INI und dem BI Presentation Server Instanceconfig.xml vornehmen, diese Konfigurationen werden aber an dieser Stelle nicht alle aufgeführt.

## **OBIEE Connection Pool Konfiguration**

Die eingestellte Anzahl an Verbindungen, welche der BI Server zur Datenbank aufbauen darf, sollte zunächst auf Basis der maximalen Anzahl gleichzeitiger User und der Anzahl Analysen je Dashboardseite eingestellt werden. Dabei hat sich die folgende bekannte Formel bewährt.

Maximum Connections = 20% der Anzahl gleichzeitiger Anwender \* Anzahl Analysen je Dashboardseite (besser Anzahl physischer SQL Abfragen)

Es ist jedoch zu beachten, dass diese Einstellungen, wie andere auch, gut getestet werden sollten. Wenn die Anzahl der Verbindungen zu klein eingestellt wurde, müssen die Anwender unnötig warten. Ist die Anzahl zu groß, kann der Arbeitsspeicher auf dem BI Server Host und DB Ebene knapp werden.

In den Admin Tool Connection Pool Einstellungen können allgemeine Einstellungen je DB Session mittels ALTER-statements eingerichtet werden. Dies erweist sich gerade zum Testen als sehr hilfreich.

- ENABLE/DISABLE Star Query Transformation
- ENABLE/DISABLE Parallel Query
- Set PARALLEL\_DEGREE\_POLICY
- Parallel Min-Time-Threshold
- Force Parallel Local
- Dynamic Sampling Level

Um über die Grid Control oder EM DB Console SQL-Überwachung direkt Informationen über die von der Datenbank ausgeführten BI Server erhalten zu können, ist es sinnvoll, benötigte Informationen vom BI Server an die Datenbank weiter zu geben. Auch hierfür können die Connection Scripts verwendet werden. Mittels der PL/SQL Packages *dbms\_application\_info* und *dbms\_session*, die per Connection Script, je Session oder je nach Information auch vor jeder Query Ausführung aufgerufen werden können, lassen sich Informationen wie Anwendername und sogar die Bezeichnung der Dashboardseiten und der Analysen setzen. Dies vereinfacht die Analyse von langlaufenden SQLs direkt aus der SQL Überwachung heraus sehr.

## **DB Hints in OBIEE**

Mit Hilfe des Admin Tools können innerhalb des RPDs auf verschiedenen Ebenen Hints gesetzt werden. Generell sollte dies aber der letzte Ausweg sein. Zunächst sollte versucht werden, das Problem auf anderem Wege zu lösen.

Dabei kann man Hints für den allgemeinen Tabellenzugriff auf den Physical Tables setzen wie z.B. die Folgenden:

```
PARALLEL(AUTO)
NOPARALLEL(T99999)
STAR_TRANSFORMATION
NO_STAR_TRANSFORMATION
USE_HASH(T99999)
...
```

Eine andere Möglichkeit ist das Setzen der Hints auf Physical Joins (Physical Complex Join, Physical foreign key), mit diesen Hints hat man die Möglichkeit, den Join zwischen zwei Tabellen und den Execution Plan in der DB feiner zu beeinflussen. Wird der Join nicht ausgeführt, ist auch der Hint nicht im physischen SQL enthalten.

```
STAR_TRANSFORMATION
USE_HASH(T19999,T99991)
NO_STAR_TRANSFORMATION
...
```

## **OBIEE und Star Transformation, wann und wie benutzen?**

Star Transformation ist eine der Performance Features, die große Vorteile bei Abfragen auf große Faktentabellen mit starken Einschränkungen auf sehr viele Dimensionen bietet, welche nur eine kleine Anzahl Zeilen zurückliefern. Abfragen mit dieser Selektivität führen ohne Star Transformation oft zu langen Ausführungszeiten, deren Ursache oft in *MERGE JOIN CARTESIAN* oder *NESTED LOOP* mit Full Table Access liegt. Mit Star Transformation wird ein Full Table Access verhindert.

Aufgrund der teilweise komplexen physischen SQLs des BI Servers gibt es jedoch leider auch Analysen, welche bei der Aktivierung von STAR Transformation für die ganze Session oder DB deutlich längere Ausführungszeiten aufweisen und sich leider durch *NO\_STAR\_TRANSFORMATION* Hints auch nicht beeinflussen lassen.

Eine weitere Problematik sind Analysen, welche auf große Faktentabellen abfragen, die keine Aggregat Tabellen besitzen und sowohl für Detail als auch für Übersichtsabfragen benutzt werden. Hier führt die STAR TRANSFORMATION auch zu längerer Ausführungszeit für die Übersichtsabfragen.

Aus diesem Grund setzen wir derzeit *STAR\_TRANSFORMATION* Hints nur in Joins mit Detail Dimension ein.

Um den DB Optimizer bei komplexen Übersichtsabfragen zu unterstützen, haben sich die USE\_HASH Hints als einfache Lösung bewährt.

### **Zusammenfassung & Ausblick**

Die Kombination aus Oracle Business Intelligence Suite Enterprise Edition 11g (OBIEE) und Oracle Exadata Database Machine bietet eine sehr gute Performance. Trotzdem behalten bekannte Performance Tuning Maßnahmen auch mit einer Exadata ihre Gültigkeit.

OBIEE ist dann am schnellsten, wenn die Abfrage ohne physische SQL und die Exadata DB beantwortet werden können.

In diesem Manuskript wurden verschiedene Tuning Massnahmen besprochen, welche dem BI Entwickler einen Überblick über die Möglichkeiten des Tunings geben.

- Hinterfrage Anwenderbeschwerden, nicht immer entsprechen die geschilderten Erfahrungen und Interpretationen den Tatsachen.
- Vertraue nicht auf die massige IO Performance der Oracle Exadata.
- Aggregate sind immer sinnvoll, auch mit Exadata.
- Caching ist wichtig. Ohne Caching kann man auch eine Exadata in die Knie zwingen.
- Vorsicht vor Dashboardseiten, die sehr viele einzelne oder sehr große physische SQLs geniieren.
- Vorsicht mit dem DOP, weniger ist manchmal mehr. Auto DOP kann das Leben einfacher machen.
- Vorsicht mit Star Query Transformation, nicht immer liefert sie bessere Abfrageperformance (gerade bei komplexem SQL).

Eine Oracle Exalytics In-Memory Machine im Zusammenspiel mit einer Oracle Exadata Maschine scheint eine optimale Erweiterung zu sein.

### **Kontaktadresse:**

Täferenstrasse 28  
CH-5405 Baden-Dättwil  
Schweiz

Telefon: +41 (0) 56 470 25 00  
Fax: +41 (0) 56 470 25 05  
E-Mail: [moritz.werning@sumit.ch](mailto:moritz.werning@sumit.ch)  
Internet: [www.sumIT.ch](http://www.sumIT.ch)