The beast - Umstieg von einer E20K nach Oracle Exadata V2

Oliver Gehlert
Metafinanz-Informationssysteme GmbH
München

Schlüsselworte:

Oracle Exadata, Data Warehousing, 11gR2, Migration

Einleitung

In den letzten beiden Jahren sind die Anforderung an das Datawarehouse deutlich gestiegen. Durch deutliche Erweiterungen in der Funktionalität und im Datenumfang kamen sowohl der Datenbankserver als auch die Storage an ihre Grenzen. Eine Aufrüstung des Hauptspeichers brachte für 12 Monate Linderung. Hauptproblem war aber die nicht mehr zeitgemäße I/O-Leistung von Server und Storage, zusätzlich stand kein zusätzlicher Plattenplatz mehr zur Verfügung. CPU-seitig wäre eine Aufrüstung noch möglich gewesen, dies hätte aber die I/O-Problematik nur verstärkt.
Mit I/O-Leistung als einer zentralen Anforderung, kam die Oracle Database Machine in die engere Wahl und wurde als kommende Plattform ausgewählt.
Bei der Migration der Architektur und des Datenbankreleases gab es zahlreiche Punkte die man beachten sollte. Ein paar davon werden im folgeden dargestellt.

Die Herausforderung

Die SUN Maschine war komplett ausgebaut und in 6 Domains unterteilt. Die Resourcenzuweisung ist im unteren Bild zu sehen:

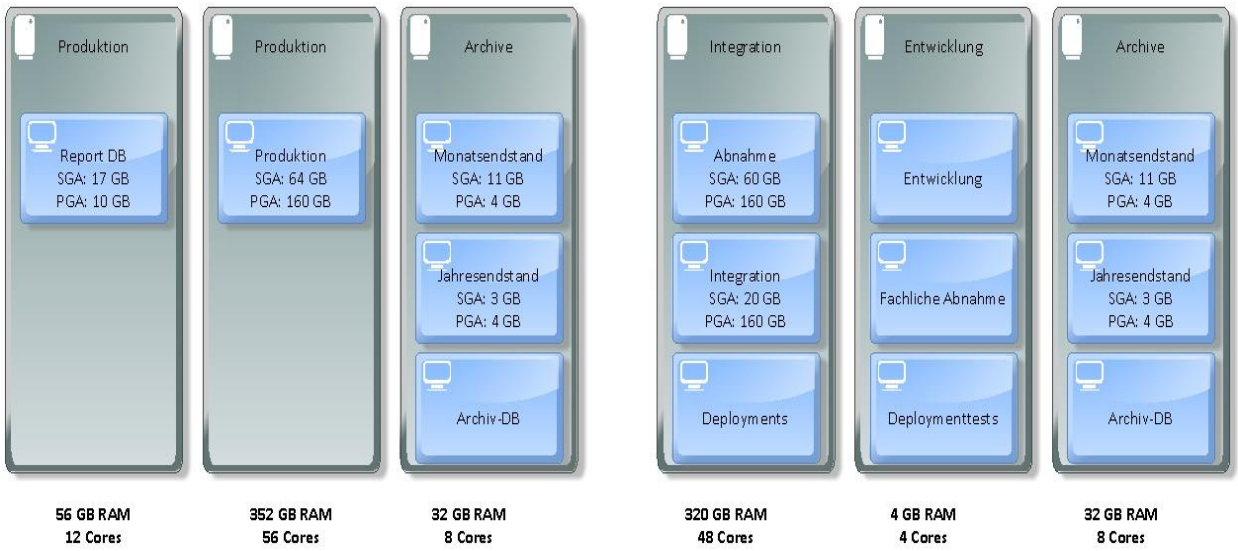

Abbildung 1 Domainaufteilung

Die gesamten Datenbanken mussten nun auf zwei Exadata Half Racks konsolidiert werden. Ein Halfrack für Produktion, das andere Halfrack für Entwicklung und Integration

Downsizing

Für die Produktionsdatenbanken standen 34 CPUs mit 76 Cores zur Verfügung, ein Halfrack hat insgesamt 8 CPUs mit 32 Cores und statt 440 GB RAM stehen nur noch 4 * 72 GB RAM zur Verfügung. Vergleicht man diese Kennzahlen, so stellt sich die Frage, wie man die Ressourcen auf die einzelnen Datenbanken verteilt und ob man die Domainen auf der Exadata Maschine nachbilden kann.

Speicheraufteilung

Beim Sizing von PGA und SGA auf der neuen Umgebung haben wir zwei unterschiedliche Verfahren verwendet. Bei der Produktionsmaschine liegt die Summe von SGA und PGA über alle Instanzen hinweg unter der Gesamtsumme des verfügbaren Hauptspeichers, bei der Integrationsmaschine haben wir uns für eine Überprovisionierung entschieden, da dort immer nur auf einer Integrationsumgebung größere Tests gestartet werden.
Den neuen Parameter „MEMORY_TARGET“ haben wir nicht verwendet, um die Verteilung auf PGA und SGA selber steuern zu können.

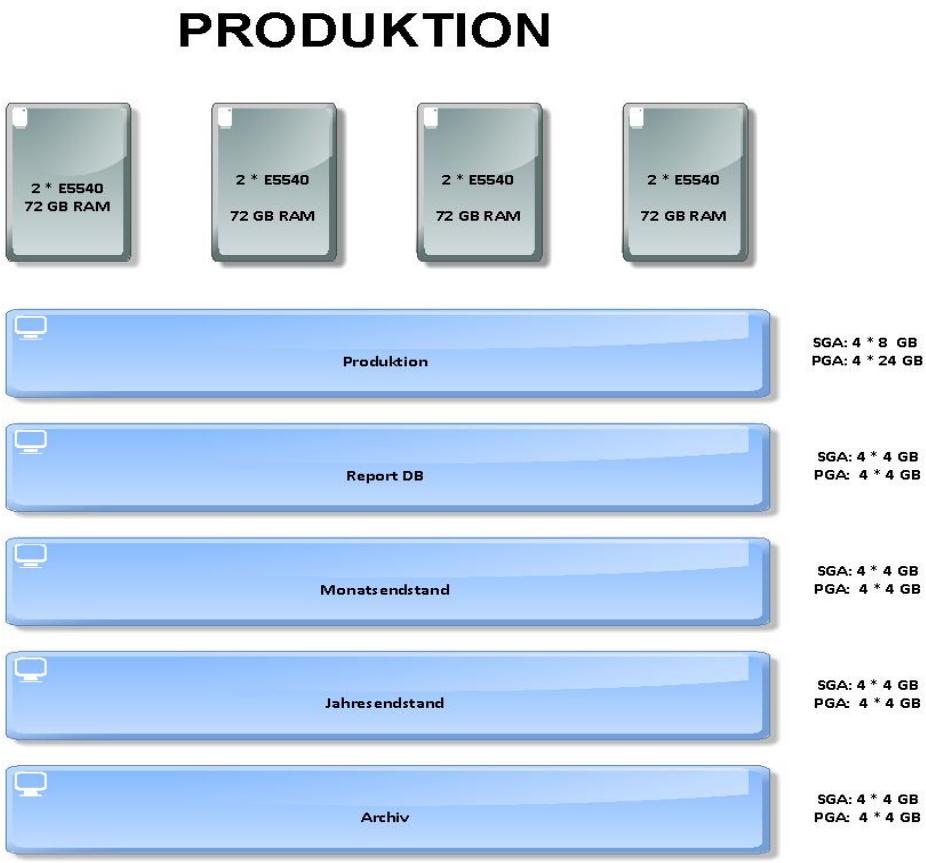

Abbildung 2 Aufteilung Produktivdatenbanken

Resourcenmanager

Da bei der Exadata keine Domains zur Verfügung stehen, musste ein anderer Weg gefunden werden, die Datenbanken voneinander zu trennen.
Bei der Version 11.2 gibt es als neues Feature das sogenannte „Instance caging“, bei dem Resoucemanagerpläne und die Anpassung des Parameters cpu_count für eine Beschränkung der Instanz auf die vorgegebene Anzahl an CPU sorgt. Der Resourcemanagerplan muss dazu CPU-Direktiven enthalten.
Wir haben uns gegen eine direkte Umstzung dieses Konzepts entschieden, da sich bei uns die Steuerung der Benutzergruppen über den erlaubten Parallelitätsgrad bewährt hat. Aufgrund der hohen Anzahl an Reportbenutzern werden diese auf Parallel 1 begrenzt, ETL-Prozesse auf Parallel 96.
In Produktion entspricht die Summe des Parameters cpu_count über alle Instanzen der Anzahl der verfügbaren Cores, in der Integrationsumgebung kommt es wieder zu einer Überprovisionierung.

Parallelität

In der alten Umgebung wurden einige Workarounds eingesetzt um die Parallelität von Statements zu steuern. Der Parameter cpu_count wurde auf 20 gesetzt, obwohl 56 Cores verfügbar waren und der Parallelitätsgrad von Tabellen wurde zu Beginn des ETL-Prozesses gesetzt. Hierbei wurde die Liefermenge berücksichtigt.

PARALLEL_DEGREE_POLICY

Um das manuelle Setzen der Parallelitätsgrade zu vermeiden, erschien der Parameter „parallel_degree_policy“ vielversprechend. Erste Tests mit einzelnen Statements zeigten eine sehr gute Performance, wenngleich sehr viele Prozesse gestartet wurden. bei parallelem Aufruf mehrer Statements brav die Performance drastisch ein, Sessions schienen zu hängen.
Bei Überwachung mit Grid Control 11g konnte man auch den Grund erkenn. Das erste Statement wurde mit 192 Parallelprozessen angestartet, die weiteren Statements wurden gequeued aber nicht mehr aus der Queue abgerufen. Behoben wurde dieses Problem mit Bundlepatch 5.
Um die Anzahl der Parallelprozesse für ein Statement zu begrenzen haben wir noch ein paar zusätzliche Anpassungen vornehmen müssen:

- Keine Parallel-Hints mit default, default
- Anpassung der Parameter
    - parallel_threads_per_cpu
    - parallel_degree_limit
    - parallel_servers_target

Zusätzlich wurde der Parallelitätsgrad über den Resourcenmanager beschränkt, damit ein Queueing vermieden wird.
Bei einigen Statements kam es zu Problemen, wenn der Parameter „parallel_degree_limit“ auf AUTO gesetzt war. Diese Statements wurden mit nur wenigen Prozessen gestartet, die dann maximal 700 MB PGA belegt haben, aber den Temptablespace gefüllt haben. Setzte man den Parameter auf MANUAL, so liefen die Statements mit mehr Prozessen und hoher PGA-Belastung in wenigen Sekunden durch. Die Ursache für das Verhalten ist noch nicht geklärt, aber das Setzen des Parameters auf MANUAL im Pre-Mappingprozess löst das Problem auf Sessionebene. Aufgrund des hohen PGA-Verbrauchs müssen die Statements von der Jobsteuerung besonders eingeplant werden.

Das Setzen des Parameters „parallel_degree_policy“ auf den Wert AUTO hat noch zwei Nebenwirkungen. Der Optimizer verwendet standardmäßig dynamic sampling um den Executionplan zu bestimmen und die Entscheidung, ob und wie stark parallelisiert wird, hängt von den erwarteten Kardinalitäten ab. Damit sind korrekte Statistiken unter 11.2 noch wichtiger als in früheren Releases.

Jobsteuerung

Bei der Umstellung auf ein Cluster waren im Bereich Jobsteuerung einige Punkten zu beachten. Alle Skripte, Logfiles und Lieferdateien mussten im Cluster erreichbar sein und der Connect konnte nicht mehr mittels „connect /“ durchgeführt werden. Das waren zumindestens die Punkte, die einem sofort eingefallen sind. Erst später kam die Frage, ob die UC4-Agenten für Oracle Enterprise Linux zertifiziert sind. Das sind sie eigentlich nicht, aber für Redhat Enterprise Linux. Auf Nachfrage haben wir dann die Freigabe für die entsprechende Oracle Enterprise Linux Version bekommen.
ACFS kann auf der Exadata aus technischen Gründen leider nicht als Clusterfilesystem verwendet werden, daher mussten wir dbfs verwenden. Die Performance von dbfs ist leider nicht besonders hoch, aber ausreichend. Möchte man ausführbare Skripte auf ein dbfs-Volume legen, so muss dies ohne die Option „direct_io“ gemountet werden, sonst bekommt man die Fehlermeldung

```
/bin/bash: bad interpreter: Bad address
```

Die Authentifizierung wurde auf ein Wallet umgestellt, das Wallet muss aber auf alle Knoten kopiert werden, da das ORACLE_HOME auf der Exadata nicht gemeinsam verwendet wird. Der Connect musste in allen Skripten von „connect /“ auf „connect /@service“ umgestellt werden.

Hybrid gehört die Zukunft

Nicht nur bei Automobilen sind Hybridmodelle im kommen. Oracle Exadata enthält bereits Hybrid Columnar Compression zur Komprimierung von Tabelleninhalten. Verwendet man die neuen Komprimierungsvarianten, so benötigt man keine zusätzliche Advanced Compresion Option. Die Verwendung bisherigen Codes, der nur das Schlüsselwort „Compress“ bei der Definition von Tabellen bzw. Tablespaces verwendet aber die Advanced Compression Option. Dieser Code muss umgeschrieben werden, um nicht zusätzliche Lizenzen zu benötigen.
Im Zuge der Migration sollte die Compression deutlich stärker verwendet werden als in der alten Umgebung, wenn dies keine Auswirkung auf die Laufzeit des ETL-Prozesses hat.
HCC bietet insgesamt vier verschieden Kompressionslevel, die auf unterschiedliche Ziele hin optimiert sind:

- Compress for query low
- Compress for query high
- Compress for archive low
- Compress for archive high

Die beiden Level compress for archive sind stärker auf Komprimierung, als auf Performance ausgerichtet, während die beiden anderen Modi die Performance nur wenig beeinflussen sollen.
Da in unserem Datawarehouse viele große Tabellen per „Truncate/Insert“ Strategie befüllt werden, sind diese für Hybrid Columnar Compression gut geeignet. Bei Performancetests ergab sich folgendes Ergebnis:

	Archive High	archive low	Query High	Query low	compress
Blöcke	67373	71107	74014	153112	226753
Erstellung	00:06:59.41	00:01:26.46	00:01:19.20	00:00:52.04	00:01:00.43
Statistiken	00:05:11.06	00:03:08.48	00:03:20.96	00:05:05.42	00:02:03.10
Count(*)	7.899.714	7.899.714	7.899.714	7.899.714	7.899.714
Num Rows laut dba_tables	21.211.480	14.021.320	14.227.820	13.768.060	7.977.240

Abbildung 3 Vergleich der unterschiedlichen Kompressionsarten

Die Performance beim erstellen einer ganzen Partition hängt vom Komprimierungsverfahren ab, besonders stark ist die Abweichung bei „compress for archive high“. Die Ursache dafür haben wir nicht bestimmen können, aus Performancegründen haben wir die Variante „compres for query high“ ausgewählt. Hiermit benötigt man nur noch 30% des Platzbedarfes der bisherigen Lösung.
Bestimmt man die Statistiken auf komprimierte Objekte, so muss man die verwendeten Optionen sorgfältig prüfen. Bestimmt man die Statistiken mit der Option „Block_Sample => TRUE“, so erhält man fehlerhafte Zeilenanzahlen bei einer Sample_Size < 100 %. In der obigen Tabelle sieht man, dass nur die Werte für „compress“ ausreichend genau sind. An den Einstellungen für die Statistikbestimmung ist noch Tuning notwendig, da die Laufzeiten in allen Fällen über den bisherigen Laufzeiten liegen, und sich im Fall von „compress for query low“ sogar verdoppeln. Die Ursache dafür ist noch nicht genau bestimmt.

Environment friendly?

Über die Einbindung der Oracle Exadata in das bestehende Rechenzentrum könnte man einen eigenen Vortrag halten. Insbesondere die Backupkonfiguration war hier ein großes Thema, da die Oracle Exadata nur eine Infiniband oder Gigabit Ehernet-Schnittstelle zur Verfügung stellt. Das Backup eines Datawarehouses über Gigabit-Ethernet ist nicht sonderlich performant, Infiniband ist aber in Rechenzentren nicht so verbreitet. Über einen Infiniband nach 10 G-Ethernet-Umsetzer konnten dann sehr gute Backup- und Recoveryzeiten erreicht werden.

Fazit

Die Performance der Oracle Exadata war von Anfang an sehr hoch. Bereits mit minimalen Anpassungen am ETL-Prozess konnten deutliche Laufzeiteinsparungen erzielt werden. In den nächsten Releases werden dann auch Neuerungen aus dem Bereich SQL umgesetzt, so dass noch weitere Einsparungen zu erwarten sind. Damit ist auch wieder die Möglichkeit gegeben Erweiterungen zu implementieren.
Die monatlichen Verrechnungskosten der Oracle Exadata sind deutlich geringer als die der bisherigen Lösung. Dies liegt nicht zuletzt an den eingesparten SAN-Kosten. Insgesamt kann man nur sagen, so macht Downsizing Controllern, Anwendern und den Entwicklern Spaß!

Kontaktadresse:

Oliver Gehlert
Metafinanz-Informationssysteme GmbH
Leopoldstraße 146
D-80804 München

Telefon:	+49 (0) 89 360531-0
Fax:	+49 (0) 89 360531-5015
E-Mail	oliver.gehlert@metafinanz.de
Internet:	www.metafinanz.de