Datenbanktuning am Beispiel eines Smart-Meter-Systems

Thomas Lehmann
Robotron Datenbank-Software GmbH
Dresden

Schlüsselworte:

Datenbank, Tuning, Smart Meter, Exadata, Database Control, AWR, Statspack, Warteereignisse, Reverse Key Index, DBMS_LOCK, Sequence, Smart Scan, Flash Cache, Hybrid Columnar Compression

Einleitung

Im Rahmen eines Skalierungstest auf Oracle Exadata X2-2 wurden Performanceprobleme ausfindig gemacht und beseitigt. Der Vortrag stellt Initial die Grundlagen und Herangehensweisen von Tuning vor. Weiterhin werden einige Beispiel aus dem Projekt näher vorgestellt. Zum Abschluss werden diese Erkenntnisse im RAC-Umfeld beleuchtet und es soll beschrieben werden, welche Exadataspezifischen Features helfen um Performanceprobleme abzumildern.

Vorstellung Smart-Metering und EXADATA

Smart-Meter oder sogenannte „Intelligente Stromzähler“ ermöglichen die Ermittlung des tatsächlichen Energieverbrauchs in Abhängigkeit der Nutzungszeit. Damit ist eine bessere Steuerbarkeit sowie Vergleichbarkeit möglich. Zudem wird durch diese Art der Erfassung ein umfangreiches Bild der Energiekosten möglich.

Oracle bietet mit der EXADATA eine sehr mächtige Hardwarekonfiguration im Datenbankumfeld. Je nach Anwendungsfall kann der Server unterschiedlich ausgestattet und konfiguriert werden. Eine spätere Aufstockung ist ebenfalls möglich (Capacity On Demand).
In der Regel laufen Datenbank in der EXADATA im Real Application Cluster.

Tuningkonzepte und Tuningwerkzeuge

Im Performancetuning unterscheidet man prinzipiell zwischen der aktiven und proaktiven Herangehensweise. Proaktives Monitoring wird in regelmäßigen Abständen wiederholt. Die Ergebnisse fußen auf Vergleichswerten (z.B. Performancewerten aus der Vorwoche). Der DBA kennt sein System und sieht damit sehr schnell ob ein Performanceproblem vorliegt.
Das aktiven Monitoring wird meinst durch einen Störfall ausgelöst. In der Regel besteht hierbei eine Überbeanspruchung einer Ressource – entweder temporär Aufgrund eines geänderten Nutzer- oder Datenverhaltens oder permanent (zum Beispiel aufgrund einer Änderung im Programmablauf).

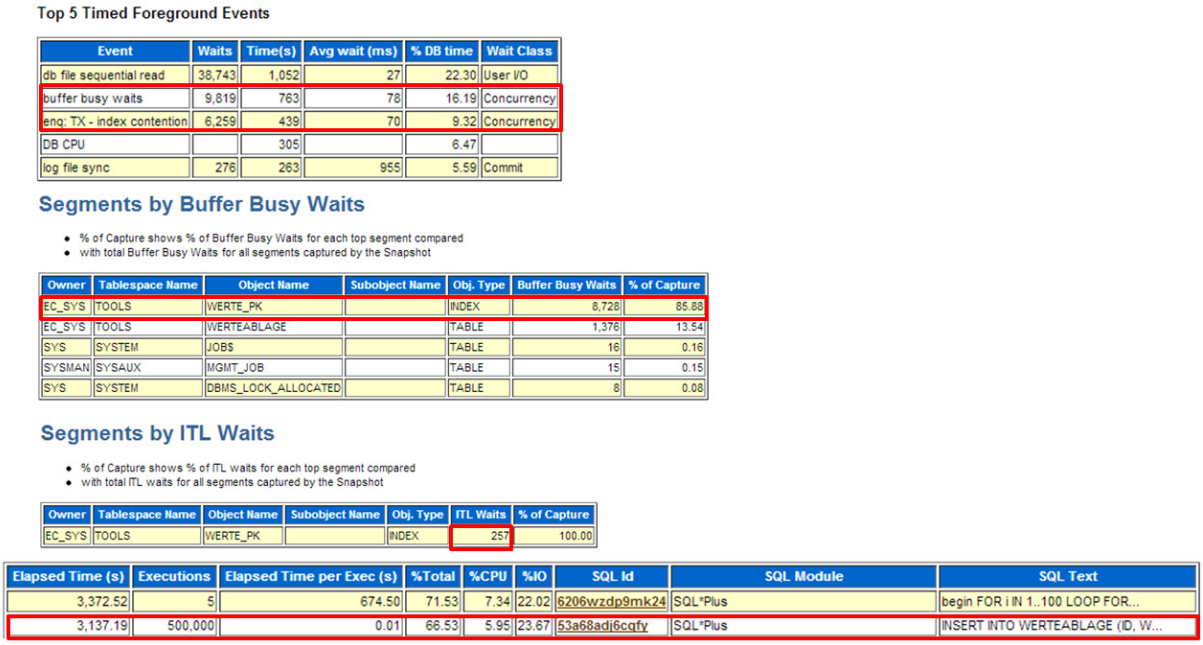
Je nach Datenbankversion, Lizenzierung und Vorlieben des jeweiligen DBAs stehen unterschiedliche Werkzeuge für die Analyse bereit. Mit Installation der ORACLE Datenbank werden Werkzeuge wie

das Database-Control, AWR- bzw. ASH-Reports oder verschiedener Advisors quasi Out-Of-The-Box ausgeliefert.
Auf vielen IT-Landschaften, insbesondere bei Standard-Editionen, ist STATSPACK im Einsatz.
Zudem gibt es eine Vielzahl von 3rd Party-Tools welche sich mit Performanceanalysen beschäftigen.
Zu guter Letzt besitzt jeder DBA eine mehr oder weniger umfangreiche Sammlung von Scripten.

Tuningmaßnahmen

Unter Laborbedingungen, wie im vorliegenden Fall, ist der Einsatz von Snapshots per AWR oder STATSPACK sinnvoll. Dadurch erhält man für den zu untersuchenden Testfall eine umfassende Übersicht.

Im ersten Testszeanrio soll die Einfügeperformance untersucht und verbessert werden. Der Testlauf findet also zwischen 2 Snapshots statt über die anschließend ein Report erstellt wird. Die folgende Abbildung zeigt Ausschnitte aus einem AWR-Report welche die Engpässe verdeutlicht:



Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
db file sequential read	38,743	1,052	27	22.30	User I/O
buffer busy waits	9,819	763	78	16.19	Concurrency
enq: TX - index contention	6,259	439	70	9.32	Concurrency
DB CPU		305		6.47	
log file sync	276	263	955	5.59	Commit

Segments by Buffer Busy Waits

- % of Capture shows % of Buffer Busy Waits for each top segment compared
- with total Buffer Busy Waits for all segments captured by the Snapshot

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Buffer Busy Waits	% of Capture
EC_SYS	TOOLS	WERTE_PK		INDEX	8,728	85.88
EC_SYS	TOOLS	WERTEABLAGE		TABLE	1,376	13.54
SYS	SYSTEM	JOB$		TABLE	16	0.16
SYSMAN	SYSAUX	MGMT_JOB		TABLE	15	0.15
SYS	SYSTEM	DBMS_LOCK_ALLOCATED		TABLE	8	0.08

Segments by ITL Waits

- % of Capture shows % of ITL waits for each top segment compared
- with total ITL waits for all segments captured by the Snapshot

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	ITL Waits	% of Capture
EC_SYS	TOOLS	WERTE_PK		INDEX	257	100.00

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
3,372.52	5	674.50	71.53	7.34	22.02	6206wzdp9mk24	SQL*Plus	begin FOR i IN 1..100 LOOP FOR...
3,137.19	500,000	0.01	66.53	5.95	23.67	53a68adj6cqfy	SQL*Plus	INSERT INTO WERTEABLAGE (ID, W...

Abb. 1: Ausschnitt AWR-Report

Die TOP5 Foreground Events bieten einen guten Einstieg in den AWR-Report. Unter Umständen sieht man sofort, wie hier im Bild dargestellt, das Problem. Die Wait-Events „buffer busy waits“ sowie „enq: TX – index contention“ deuten auf Hot Blocks sowie Indexzugriffe hin – bzw., wie in den Segmentstatistiken sichtbar , aus einer Kombination aus beidem. Der Primary Key WERTE_PK scheint der begrenzende Faktor zu sein.

Die Lösung des Problems ist, die Schreiboperationen auf dem Primary Key so zu gestalten, dass gleichzeitige Änderungen an gleichen Blöcken so selten wie möglich stattfinden. Das technische Mittel dazu ist ein Reverse Key Index.

Im zweiten Testszenario soll das pessimistische Locking der Anwendung per DBMS_LOCK untersucht und verbessert werden.
Nach mehreren Testläufen war erkennbar, dass die Mapping-Tabelle DBMS_LOCK_ALLOCTED ein Problem darstellt. Viele Parallele Sessions Lesen und Schreiben in diese Tabelle. Somit stellt diese die begrenzende Ressource da. Das Problem kann umgangen werden, indem im Programmcode auf DBMS_LOCK.ALLOCATE_UNIQUE verzichtet und anstelle dessen numerische IDs als Sperridentifikator benutzt.

Wie bereits im Vorfeld beschrieben reden wir im EXADATA-Umfeld typischerweise über eine RAC-Datenbank. Im Dritten Testfall soll das Verhalten von Sequenzen untersucht werden. Diese werden typischerweise zur Generierung von Primary-ID-Werten benutzt.
Standardmäßig nutzt im RAC-Umfeld jede Instanz ein eigenes Set von ID-Werten aus der Sequence (in Abhängigkeit des eigestellten CACHE-Wertes). Bei der Auswertung von Datensätzen ist zu beachten, dass dadurch die Reihenfolge der IDs nicht mehr gewährleistet ist.

EXADATA-Spezifika

Im Folgenden soll auf 3 Key-Features der Exadata eingegangen werden.

Mit Smart Scan stellt das Exadata Storage Grid eine Möglichkeit bereit, Daten auf Storageebene zu filtern. Dabei werden Vergleichsoperatoren bzw. logische Operatoren unterstützt.
Ebenso bei administrativen Tätigkeiten (Backup, Optimizerstatistiken) bietet dieses Feature deutliche Performancesteigerungen.

Mit der EXADATA X2-2 bietet der Flash Cache einen zusätzlichen Second Level Cache. Insbesondere OLTP-Systeme profitieren von dieser zusätzlichen, schnellen Datenablage. Mit EXADATA X3 wurde dieser Cache beschreibbar gemacht. Damit profitieren dann ebenfalls DML-Operationen.

Mit Hybrid Columnar Compression (HCC) steht ein neuer, spaltenbasierter Komprimierungsalgorithmus zur Verfügung. Je nach Komprimierungstyp findet eine deutliche Reduzierung von I/O-Operationen bei marginal mehr CPU-Verbrauch statt. Im Konkreten Anwendungsfall konnte eine Tabelle von 10,5 GB auf 1,1 GB reduziert werden.

Zusammenfassung

Im ersten Schritt ist ein für den Analysefall geeignetes Werkzeug auszuwählen oder eine Kombination aus Werkzeugen.

Es sollte stets eine Top-Down-Analyse des Problems durchgeführt werden. Tuningmaßnahmen sollten so durchgeführt werden, dass die positiven Auswirkungen am Größten sind: von der Anpassung des Anwendungscodes über das Datenbankdesign und letztlich Hardwarekonfiguration.

Es ist sehr häufig notwendig in enger Abstimmung mit dem Softwareentwickler zu stehen, da Codeänderungen durchgeführt werden müssen.

Unter Umständen müssen mehrere Optimierungsläufe durchgeführt werden um ein optimales Ergebnis zu erhalten.

In verteilten Umgebungen (RAC) treten Performanceprobleme deutlicher zum Vorschein. Hier gilt es frühzeitig im Entwicklungsprozess entgegenzusteuern.

Die bereitgestellten EXADATA-Features können zur deutlichen Performanceverbesserung beitragen.

Kontaktadresse:

Thomas Lehmann
Robotron Datenbank-Software GmbH
Stuttgarter Straße 29
D-01189 Dresden

Telefon:	+49 (0) 351-25859 2782
Fax:	+49 (0) 351-25859 3696
E-Mail	Thomas.Lehmann@robotron.de
Internet:	www.robotron.de