

# Eine für Alle - Oracle DB für Big Data, In-Memory & Exadata

Dr.-Ing. Holger Friedrich  
sumIT AG

Baden - Schweiz

## Schlüsselworte

Oracle Datenbank, In-Memory, Big Data, Big Data SQL, Exadata

## Einleitung

Oracle bietet mittlerweile verschiedene Speichertechnologien als Quellen für die Oracle Datenbank an. Insbesondere sind dies die Oracle Big Data Appliance, Exadata Storage und der Columnar Store der In-Memory Datenbankoption. Jede dieser Technologien, erweitert um spezielle Oracle Software, z.B. Big Data SQL, bietet eigene Performance-Feature. Die Oracle Datenbank kann auf alle diese Speicherarten transparent mit Standard-SQL zugreifen, was einzigartig im RDBMS-Markt ist. Kunden müssen daher keine einzige Zeile Applikationscode verändern oder neu schreiben, um auf Daten zuzugreifen, ganz egal wo diese gespeichert sein mögen. Die Entscheidung wo welche Daten aufbewahrt werden sollen, kann der Kunde nunmehr aufgrund anderer Kriterien, wie Performance, Kosten, Menge etc. treffen, der Zugriff per SQL ist über die Oracle Datenbank immer gewährleistet. Entscheidend ist zu wissen, welche der Speichertechnologien, welche spezifischen Eigenschaften für die Oracle Datenbank zur Verfügung stellt, welche Vorteile sich daraus ergeben, aber auch welche Grenzen existieren. Sind den Architekten und den Entwicklern diese Grenzen bewusst, kann eine Systemarchitektur aufgebaut und effizient genutzt werden, welche die Anforderungen des Kunden mit dem besten Kosten-/Nutzenverhältnis erfüllt.

## Alte Zeiten

Bis zur Einführung der Exadata Storage Technologie zur Oracle Open World 2008, also vor sieben Jahren, basierte die Oracle jahrzehntelang auf derselben Speicherarchitektur und daraus resultierenden Verarbeitungslogik. Diese bestand primär aus den folgenden drei Speicherarten:

- SGA (System Global Area)
- PGA (Program Global Area)
- Persistenter Speicher (Festplatten, Flash, etc)

Der spannende Aspekt der PGA, im Rahmen unserer Betrachtung alter und neuer Speicherformen ist, dass hier bezüglich der logischen Abarbeitung von SQL-Abfragen hauptsächlich die Musik spielt. Hier wird bzw. wurde traditionell gefiltert, projiziert, sortiert, aggregiert, partitioniert, gehasht und gejoint, werden Muster gematcht, reguläre Ausdrücke verarbeitet, wird gerechnet und werden alle weiteren Verarbeitungsschritte ausgeführt.

So wie oben skizziert, mit diesen Speicherbereichen, arbeitet auch eine aktuelle Oracle 12c Datenbank, so man keine der neuen Speicher- und Speicherzugriffstechnologien, namentlich Exadata, Big Data SQL oder den In-Memory Columnar Store nutzt. Diese Architektur hat zwei prinzipielle Nachteile, einen der sich auf Performance und einen der sich auf die Datenverfügbarkeit und Kosten auswirkt.

*Performance:* In der klassischen Oracle-Architektur werden einzig im Bereich des individuellen Verarbeitungsprozesses und seiner PGA die Daten wirklich verarbeitet. Die beiden anderen Speicherbereiche, SGA und persistenter Speicher sind de facto Ablagen. Das heisst aber, dass eine Unmenge an Daten vom persistenten Speicher zum DB-Rechner und vom SGA in den jeweiligen DBA transportiert werden müssen, ohne dass sie zur Ergebnisberechnung benötigt werden. Grund

dafür sind die zeilenorientierte Speicherdarstellung und die Verwendung von Datenbankblöcken fixer Grösse bei der persistenten Speicherung sowie im Block Buffer des SGA.

*Verfügbarkeit und Kosten:* Die Oracle Query-Engine kann in der klassischen Architektur ausschliesslich auf Daten arbeiten, welche in Oracle's Datenbankblock-Format vorliegen. Das heisst, die Datensätze müssen in Zeilenform strukturiert und in Oracle-Datenblöcke zusammengefasst sein. Werden sie persistent gespeichert, müssen sie im Oracle-Datenbankfileformat formatiert gesichert werden. Diese Einschränkung bedingt aufwendige Vorarbeiten (laden, transformieren, speichern) bevor neue Daten und Datenquellen für Abfragen erschlossen sind, was sowohl hohe Kosten verursacht, als auch eine längere Totzeit erzeugt, bis Daten für Auswertungen zur Verfügung stehen.

Im September 2008 jedoch begann für die Oracle Speicherarchitektur eine neue Zeitrechnung. Die seitdem eingeführten Neuerungen werden in den folgenden Abschnitten vorgestellt und ihre spezifischen Eigenschaften diskutiert.

### **Exadata Storage**

Auf der Oracle Open World 2008 kam Oracles CEO Larry Ellison auf die Bühne und versprach eine neue Appliance, mit der insbesondere datenintensive Abfragen 10 Mal schneller verarbeitet würden als je zuvor. Diese ‚Oracle Database Machine‘ bestand aus Datenbankrechnern, einer Infiniband-basierten Vernetzung und den sogenannten Exadata Speicherzellen. Der Schlüssel zur zehnfachen Beschleunigung von Abfragen lag in dieser Architektur nicht in den Rechnerknoten und auch nicht primär in der Anbindung der Exadata-Speicherzellen per Infiniband. Der Hauptgrund für die überlegene Performance der, heute einfach Exadata genannten Appliance ist Aufbau und Funktionalität der Exadata-Storagezellen.

Eine Exadata-Storagezelle besteht aus 18 Festplatten (rotierend oder Flash), einer grösseren Menge Flashspeicher und insbesondere einem eigenen Rechnerknoten, der diese Speichereinheit bearbeitet und administriert. Auf dem Rechnerknoten jeder Exadata Speicherzelle läuft die Exadata Storage Software. Diese bietet der Oracle-Datenbanksoftware, welche auf den eigentlichen, klassischen Datenbankservern läuft weitaus komplexere Funktionen, als nur das lesen und schreiben von Datenbankblöcken an.

Um diese Funktionen möglichst effizient zu implementieren nutzt die Exadata-Speicherzelle verschiedene Techniken, wie z.B. Storage Indizes, lokales Scanning und Hybrid Columnar Compression. Aus den Ergebnissen der Verarbeitung werden abfragespezifische Zeilen gebildet und diese im regulären Oracle-Datenbankblockformat zusammengefasst. Lediglich diese Teilergebnisse zu einer Gesamtabfrage werden schliesslich an die Datenbankserver übertragen.

### **Big Data SQL**

Während Exadata Storage erfolgreich Performance-Probleme bei der Nutzung von Oracle Datenbanken adressiert, bleibt das Problem des schnellen, kostengünstigen Zugriffs auf neue Daten von dieser Technologie unberührt.

Für die schnelle Analyse grosser Mengen rapide eintreffender Daten, welche nicht aufwendig vorverarbeitet werden sollen, hat sich innerhalb der letzten Jahre die Big-Data-Technologie etabliert. Diese Technologie kombiniert einen redundanten, verteilten Dateispeicher mit verteilter Datenverarbeitung auf kostengünstigen Rechnerknoten. Die Technologie ist sehr gut für die dauerhafte Speicherung und Verwaltung auch massivster Datenmengen geeignet. Auf dem Hadoop Distributed File System (HDFS) und No-SQL Datenbanken, z.B. HBASE, gab es zur Datenverarbeitung und –abfrage zunächst lediglich die batchorientierte Verarbeitungsmethode des javabasierten MapReduce. Mittlerweile jedoch ist die ganze Community dabei typisch relationale Datenbanktechnologien, wie synchrone Datenabfrage mit Impala und programmatischen Zugriff mit Spark zu entwickeln.

Ein Big-Data-System zu administrieren bleibt jedoch aufwendig. Es ist gegenüber einer relationalen Datenbank sehr langsam, beim Datenzugriff recht ineffizient (keine Indizes und andere Optimierungen out-of-the-box) und benötigt einen ganzen Zoo von Werkzeugen, um verschiedene Aspekte, z.B. Sicherheit zu bearbeiten. Schliesslich ist die Verbindung von Daten eines Big-Data-Systems mit anderen Datenquellen zum Zwecke der Auswertung eine Herausforderung.

Oracle bietet ihren Kunden eine speziellen Big Data Appliance (BDA) an, um den Kunden, den Engineeringaufwand abzunehmen, administrative Tätigkeiten zu vereinfachen und gute Cluster-Performance zu garantieren. Noch wichtiger als die blosse Appliance war jedoch die Einführung der ‚Big Data SQL‘-Software (BDS). BDS ist eine Adaption der Exadata-Storage Software, die analog zu anderen Hadoop-Erweiterungen, wie Spark oder Yarn auf den Clusterknoten eines Hadoop-Clusters installiert wird. Wie in den Exadata-Systemen bietet BDS einer Oracle Datenbank nicht nur Services zum Zugriff auf Daten, sondern auch zur Vorverarbeitung derselben an. In der gegenwärtigen Ausbaustufe bietet BDS noch nicht den gesamten Umfang, welcher auf Exadata Storage-Zellen zur Verfügung steht. Die wichtigsten Performancebeschleuniger zum Smart Scanning, nämlich Filterung, Projektion, scoring von Modellen etc. stehen jedoch bereits zur Verfügung.

Mit BDS hat Oracle also auch für die Hadoop-Welt Verarbeitungsfunktionalität in den Storage-Layer verschoben. Die Query-Engine kann Teile ihrer Arbeit an den Storage-Layer delegieren. Wertvoller Prozessspeicher wird entlastet, die Datenbank-Server-CPU's werden entlastet und weniger Daten belasten die Netzwerkkapazität. Für den Anwender ergibt sich daraus eine extrem verbesserte Nutzung von Ressourcen, bei gleichzeitiger Erhöhung der applikatorischen Sicherheit. Ausserdem resultiert eine massive Erweiterung der applikatorischen Möglichkeiten durch einfachere Datenintegration.

### **In-Memory Columnar Store**

Mit Big Data SQL hat Oracle die Welt der Hadoop Cluster für die Oracle Datenbank erschlossen. Gleichzeitig entwickelte sich aber an anderer Stelle, nämlich der analytischen High-End-Performance ein interessanter Wettbewerb der Hersteller und Technologien. Dieser Wettbewerb führte zu einer Renaissance spaltenbasiert arbeitender Datenbanken.

Der Vorteil spaltenbasierter Datenbanken erschliesst sich bei der Anwendung für analytische Queries sofort. Da analytische Abfragen in der Regel wenige Spalten einer Tabelle nutzen, dafür aber eine sehr grosse Anzahl von Zeilen, ist das Scanning für analytische Queries in spaltenorientierten Datenstrukturen wesentlich schneller. Dies gilt prinzipiell für Implementierungen aller Hersteller, seien es SAP's Hana, Exasol oder Andere. Wie leistungsfähig ein Columnar Store im Vergleich zum Wettbewerb, bei gleichen Abfragen und vergleichbarer Infrastruktur ist, hängt daher hauptsächlich davon ab,

- welche weiteren Technologien, neben dem einfachen Scanning zur Verfügung stehen und
- wie der Columnar Store ins Gesamtsystem eingebettet ist.

Oracles In-Memory Columnar Store besitzt eine Vielzahl zusätzlicher Feature, neben dem einfachen Scanning, welche die Performance erhöhen, z.B. SIMD-Vektorverarbeitung und testen von Filterbedingungen direkt auf binär komprimierten Daten. Entscheidender jedoch ist, dass auch der In-Memory Columnar Store lokale Funktionalität zur autonomen Bearbeitung von Teilabfragen besitzt. Die Query-Engine der Datenbank kann also vom In-Memory Columnar Store, nicht nur Blöcke anfordern, wie es beim klassischen Block Buffer der SGA der Fall oder Teile eines Datenbankfiles, wie es bei klassischem Platten- oder Flashspeicher der Fall ist. Stattdessen ist die Situation ähnlich wie bei Exadata Storage und Big Data SQL. Die Query-Engine kann Teilaufgaben direkt an den spaltenorientierten Speicher vergeben. Dieser löst die Aufgaben selbständig, höchst effizient und übergibt danach die Ergebnisse, zur weiteren Verarbeitung in der PGA, an den anfragenden Serverprozess.

Teilaufgaben können von grosser Komplexität sein, wie zum Beispiel In-Memory-Aggregationen. Diese berechnen Ergebnisse komplexer dimensionaler Queries in einem Bruchteil der Zeit, welche bislang bei der Verwendung zeilenorientierter Star-Schemata gebraucht wurde, ohne dass der grosse Aufwand für die Modellierung und Wartung unterstützender Strukturen, z.B. Bitmap-Indizes anfiel.

### Unter einem Kommando

In den vorigen Paragraphen wurden drei Speichertechnologien, bzw. Erweiterungen von Speichertechnologien vorgestellt, um welche Oracle ihr Portfolio im Verlaufe der vergangenen Jahre ergänzt hat. Jede dieser Technologien für sich löst Performanceprobleme, erschliesst neue Möglichkeiten und, im Falle von Big Data SQL auch ganz neue Datenformen und –mengen.

Für Anwender richtig wertvoll werden diese Technologien aber erst durch die Tatsache, dass alle ihre Eigenschaften zentral und transparent aus dem Oracle Datenbankserver genutzt werden können. Das heisst, dass die Nutzung der Kombination von Query-Offloading, dynamischer Quell-Integration, In-Memory-Performance und aller weiteren Eigenschaften ohne Änderung auch nur einer Zeile applikatorischen Codes möglich ist. Dies erlaubt Verarbeitungen welche bislang so nicht möglich waren.

- Transparente Abfragen über eine Struktur, deren Teile in mehreren verschiedenen Speichertechnologien vorliegt. Beispielsweise liegen mehr Spalten einer Struktur im In-Memory-Columnar-Speicher, andere als Zeilen im Exadata-Storage oder im Block-Buffer vor. Oder Partitionen mit neusten Daten befinden sich im Columnar Store, etwas ältere im Exadata Storage Layer in Flash, noch ältere im Exadata Storage Layer auf den Festplatten und sehr alte Partitionen auf gewöhnlichem Plattenspeicher.
- Abfragen über mehrere Strukturen, die in verschiedenen Technologien liegen. Ein Beispiel ist der Join von Kundendaten, welche im Speicher der Datenbank liegen mit Call-Data-Records, welche auf einem Hadoop-Cluster gespeichert sind. Abbildung 1 zeigt einen solchen Anwendungsfall.

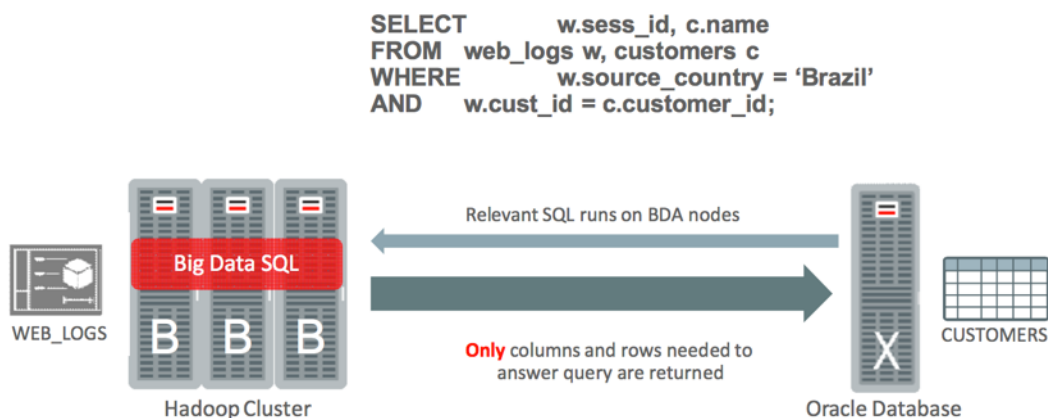


Abb. 1: Integration von Hadoop- und Exadata-Daten unter Verwendung von Big Data SQL

- Erweiterung des Sicherheitsmodells und der Policies von der Datenbank auf einen Hadoop-Cluster. So können Daten auf der Big-Data-Welt in Datenbankapplikationen eingebunden werden, ohne dass zusätzliche Fähigkeiten geschult werden müssen und administrative Aufwände entstehen.
- Ausführung von Abfragen zwischen Datenbank und Big-Data-Welt, welche bislang aufgrund von Sicherheitsanforderungen unmöglich waren. Sollen zum Beispiel Transaktionen aus dem Hadoop-Cluster zu einem Kunden gejoint werden, welche nur über die Kreditkartennummer verbunden sind, die Kreditkartennummer, aber aus Sicherheitsgründen nicht dargestellt werden darf, ist dies mit traditionellen Mitteln

(Sicherheitsmanagement auf Hadoop und der DB getrennt) kaum möglich. Mit Big Data SQL kann die Datenbank den Join durchführen und erst danach im Result-Set die Join-Spalte unterdrücken oder per Redaction maskieren.

### **Fazit**

Ein Nachteil der schönen neuen Welt ist, dass für alle oben diskutierten Technologien bestimmte Infrastrukturvoraussetzungen gelten.

- In-Memory Columnar Store benötigt mindestens den Datenbank-Release 12.1.0.2.
- Exadata Storage gibt es ausschließlich in Exadata und Super Cluster Appliances,
- Big Data SQL erfordert die Verwendung einer Big Data Appliance. Zudem muss der zugreifende Datenbankserver auf einer Exadata Appliance laufen

Sind diese Voraussetzungen allerdings erfüllt, bekommt der Anwender durch Kombination der Speichertechnologien eine Vergrößerung der Performance, der Möglichkeiten und der Freiheitsgrade bei der Applikationsentwicklung und -nutzung, wie sie bislang unmöglich gewesen ist. Dies ist ein Punkt an dem sich Oracle eindeutig vom Feld der Wettbewerber abgesetzt hat.

### **Kontaktadresse:**

Dr.-Ing. Holger Friedrich  
sumIT AG  
Täfernstrasse 28  
CH-5405 Baden-Dättwill

Telefon: +41 (0) 56 – 470 2500  
Fax: +41 (0) 79 – 320 8179  
E-Mail: [holger.friedrich@sumit.ch](mailto:holger.friedrich@sumit.ch)  
Internet: [www.sumit.ch](http://www.sumit.ch)