

# Oracle Database In-Memory vs. SAP HANA

Stefan Menschel, ORACLE Deutschland B.V. & Co. KG

Vieles wurde in der jüngsten Vergangenheit zum Thema „In-Memory“ veröffentlicht. Allen voran hat es SAP geschafft, durch ein enormes Marketing die Aufmerksamkeit auf ihre SAP-HANA-Technologie zu lenken. So ist es nicht verwunderlich, dass ein großer Teil der Anwender In-Memory Computing zunächst mit dem Produkt der SAP verbindet. Aus meiner Sicht gibt es in der Technologie des In-Memory Computing grundsätzliche Unterschiede. SAP spricht bei SAP HANA von einer „In-Memory-Datenbank“, während bei Oracle von „Database In-Memory“ die Rede ist.

Die Entwicklung der SAP-HANA-In-Memory-Datenbank, auf Basis der am Hasso Plattner-Institut in Potsdam entwickelten Sanssouci-Datenbank, ist eine Verschmelzung von drei unterschiedlichen Technologien. Der zeilenorientierte Anteil stammt von „P\*Time“, einem kleinen, Memory-optimierten OLTP-RDBMS; der spaltenorientierte Anteil kommt vom SAP-eigenen Produkt TREX des SAP BWA, während der Persistence Layer von der SAP-eigenen MaxDB abstammt. Mit SAP HANA wird versucht, alle Daten einer Datenbank hochkomprimiert in der Form von indizierten Spalten im physikalischen Speicher eines Datenbank-Servers abzulegen.

Spaltenorientierte Ablage der Daten ist ein weiteres Merkmal des In-Memory Computing. Dies ist nicht neu, wurde jedoch aufgrund deutlicher Nachteile im OLTP-Umfeld von allen namhaften Datenbank-Herstellern bisher nicht eingesetzt. Erst das enorme Wachstum zu analysierender Daten und der Preisverfall für RAM haben hier zu Veränderungen/Ergänzungen der bisherigen Datenbank-Technologie geführt.

Die SAP-HANA-Lösung hält die Daten primär im Hauptspeicher und verarbeitet sie dort. SAP spricht deshalb von einer In-Memory-Datenbank. Sehr große Datenbanken benötigen hier trotz einer initial hohen Komprimierung enorm viel Hauptspeicher. Der erwähnte Preisverfall der RAM-Module ist nicht ausreichend, da hier doch hohe Kosten allein für die Hardware einer SAP

HANA Appliance oder einer SAP Tailored Datacenter Integration (TDI) anfallen.

Da HANA eine reine In-Memory-Datenbank ist, erfolgt die Verfügbarkeit nach einem Server Crash erst nach dem Laden der Objekte in den Arbeitsspeicher. Das ist bei einer kritischen Anwendung wie der SAP Business Suite wichtig, da hier in Abhängigkeit von der Größe zum Teil mehrere Stunden für das Recovery anfallen. Alternativ können kostenintensive Disaster-Recovery-Lösungen vor einer längeren Auszeit schützen.

Die SAP-HANA-Plattform kann in verschiedenen Szenarien, etwa als Accelerator, als Datenbank für das SAP Business Warehouse oder die SAP Business Suite zum Einsatz kommen. Hierfür ist eine vollständige Migration der bestehenden Datenbank mit den SAP-Landscape-Transformation-Services notwendig. Einmal migriert, gibt es bislang keine technisch unterstützten Möglichkeiten, diesen Weg wieder rückgängig zu machen. Dies kommt einer Einbahnstraße gleich und sollte vorher bestens überlegt sein.

Gerade für sehr große Datenbanken im Business-Warehouse-Umfeld ist es nicht erforderlich und sinnvoll, die kompletten Daten dieses Systems in einer reinen In-Memory-Datenbank zu betreiben. Technisch bietet SAP hier ein Nearline-Storage-System auf Basis einer weiteren Datenbank in Form von Sybase IQ an, um diese an die bestehende HANA-Lösung anzudocken. In diesem Fall kann man

nicht mehr von einer reinen In-Memory-Datenbank sprechen. Im Gegenteil, für die Haltung der Daten einer einzigen SAP-Anwendung (hier: Business Warehouse) sind zwei getrennt arbeitende Datenbanken notwendig. Das hat Konsequenzen in Form höherer Aufwände für Administration, Disaster-Technologien, Backup und Recovery.

## In-Memory und OLTP

Rein spaltenorientierte Datenbank-Systeme – wie auch SAP HANA – haben deutliche Nachteile im Bereich der transaktionalen Prozesse gegenüber klassischen, zeilenorientierten Datenbank-Systemen, etwa wenn Daten angelegt, verändert und gelöscht werden. Hier werden wenige Zeilen mit vielen Spalten gleichzeitig verändert, während man im analytischen Umfeld wenige Spalten mit vielen Zeilen lesen möchte.

Eine grundlegende Veränderung findet nun in HANA in der Umsetzung von Update- oder Delete-Anforderungen statt. Die SAP-HANA-In-Memory-Datenbank arbeitet nach dem „INSERT-ONLY“-Ansatz, um alle bisher bekannten Performance-Probleme einer spaltenorientierten Datenbank zu umgehen. Das bedeutet, aus jedem „UPDATE“ wird intern ein „INSERT“. Diese oder auch richtige „INSERTs“ werden in einen schreib- und zeilenorientierten, nicht komprimierten „Delta Store!“ geladen. Von dort werden die Daten in einen zweiten, nun spaltenorientierten,

nicht sortierten „Delta Store2“ übertragen. Was folgt, ist eine finale Übertragung in den „Main Store“. Das Wachsen der Delta-Stores führt zunehmend zu einer schlechteren Performance weiterer analytischer Abfragen. Aus diesem Grund muss intern ein Merge-Prozess stattfinden, der die Einträge in den jeweiligen Delta-Stores verarbeitet. Der Prozess vom „Delta Store2“ in den „Main Store“ ist dabei sehr kritisch und beeinträchtigt die Performance der SAP-HANA-Lösung (siehe Abbildung 1).

Dieser notwendige Merge-Prozess sowie weitere interne Services erfordern zusätzlich etwa 50 Prozent RAM zu den im Main Store abgelegten operativen Daten. Die Delta-Stores führen zu einer zeitlich begrenzten, mehrfachen Ablage von gleichen Daten im Memory. Alle von der Datenbank mit einem „Commit“ abgeschlossenen Transaktionen werden in den Redo-Log-Dateien vor Datenverlust gesichert. Zusätzlich wird alle fünf Minuten ein Savepoint eines kompletten konsistenten Image der In-Memory-Datenbank auf die vorhandenen Disks geschrieben.

**SAP HANA und „Code-Push-Down“**

Mit der HANA-Lösung kommt nicht nur eine neue Datenbank, sie beinhaltet zunehmend auch Teile des Applikationscodes. SAP führt hier einen Wechsel weg von einem lange propagierten Paradigma durch. Bisher wurde der Applikationscode immer auf dem SAP-Applikationsserver ausgeführt. Mit SAP-HANA wird versucht, immer mehr Anwendungscode auf der SAP-HANA-Plattform auszuführen. Dies hängt zum einen damit zusammen, dass bisherige Statements fast immer mit einem „SELECT \*“ erfolgten. Das hat/hatte zur Folge, dass alle Spalten einer Tabelle sofort in das RAM geladen werden/wurden. Um dieses Verhalten zu umgehen, werden Zugriffe auf die notwendigen Spalten über verschiedene Views, Attribute-, Analytical- oder Calculation-Views realisiert. Dies ist auch der Grund, aus dem Anwender ihre eigenen Skripte entsprechend anpassen müssen, um die Vorteile der HANA-Datenbank optimal nutzen zu können.

Zusätzlich wird mit SAP-ABAP-Core-Data-Services versucht, vorhandene Funktionalitäten der HANA-Datenbank wie „REPLACE“, „SUBSTR“ oder „LPAD“ zu nut-

zen, oder es werden Funktionen zur Umwandlung von Einheiten, Währungen sowie Datumsformaten in der Datenbank abgelegt und dort ausgeführt. Diese Services stehen nicht nur ausschließlich für die HANA-Datenbank zur Verfügung, sondern werden ebenso bei den traditionellen Datenbanken wie Oracle zunehmend eingesetzt. Das kann zu Vorteilen in der Performance bei der Ausführung von Applikationslogik führen, muss es jedoch nicht zwangsläufig. Denn heute gibt es für sehr große SAP-Systeme eine Vielzahl von Applikationsservern, um die Last der Anfragen Tausender Anwender über viele Systeme zu verteilen. In Zukunft würde diese Last auf einem einzigen Server abgearbeitet, der zusätzlich neben dem In-Memory Computing die CPU noch stärker belastet. Die genauen Auswirkungen eines derart massiven „Code-Push-Down“-Prinzips sind heute im SAP-Umfeld überhaupt noch nicht absehbar.

**Die Oracle-Database-In-Memory-Option**

Bisher war die Oracle-Datenbank nicht rein Disk-basiert. Optimierungen mithilfe von Speicher-Architekturen wie Buffer-, Row-, Result-Cache etc. waren Voraussetzungen für eine optimale Performance bei mehr als 95 Prozent aller transaktionalen und analytischen Anforderungen an die Datenbank. Mit der Oracle-Datenbank-Version 12.1.0.2 steht eine neue Memory-Technologie zur Verfügung. Oracle Database In-Memory adressiert genau den

Bereich von Anforderungen, in dem heute und zukünftig schnellere Analysen oder Aggregationen über große Datenmengen und wenige Spalten erforderlich sind. Anderweitige allgemeine Vorstellungen über generell deutliche Verbesserungen in der Performance durch SAP HANA oder auch durch die neue Oracle-Datenbank-In-Memory-Technologie haben nichts mit der Realität zu tun.

Die Datenbank-Version 12c erhält zusätzlich zu den bisherigen Elementen in der System Global Area (SGA) einen weiteren Bereich, den In-Memory Column Store. In diesem Bereich des Hauptspeichers können Tabellen vollständig, einzelne Spalten einer Tabelle oder einzelne Partitionen im Spaltenformat abgelegt werden. Allein die Bezeichnung „Database In-Memory“ soll verdeutlichen: Die Oracle In-Memory-Technologie ist etwas anderes als eine „In-Memory-Datenbank“. Mit dieser Option verbindet Oracle als einziger Datenbank-Hersteller beide notwendigen Technologien in einer einzigen Datenbank.

Die bisherige Praxis einer zeilenorientierten Pufferung im Hauptspeicher und deren Speicherung auf Disk oder Flash bleibt vollwertig bestehen. Dieser Ansatz hat sich über viele Jahre bis heute für transaktionale Anforderungen bestens bewährt und bleibt unverändert. Im Gegensatz zu dem vorhandenen Buffer Cache wird der Column Store nicht auf den Storage-Systemen (Disk oder Flash) gesichert, sondern liegt ausschließlich im

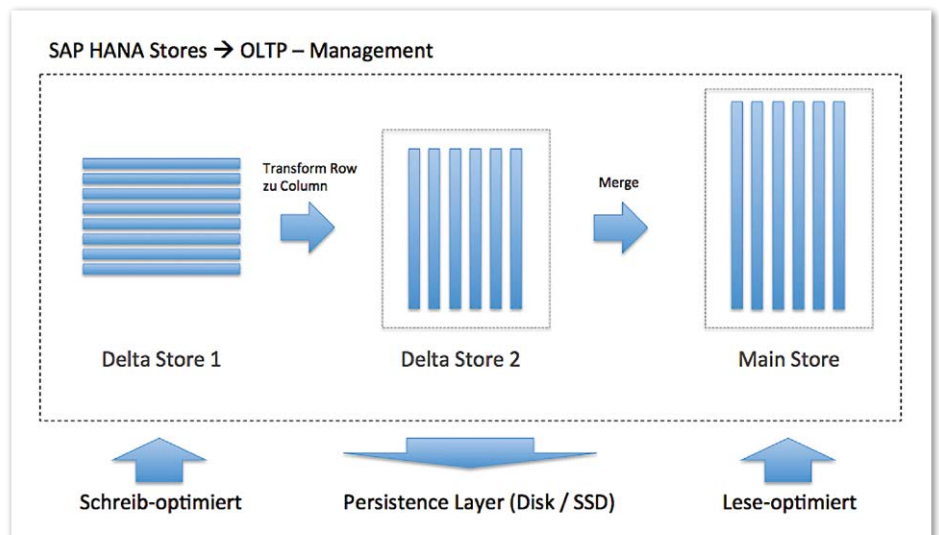


Abbildung 1: Verarbeitung der OLTP-Daten in verschiedenen Stores

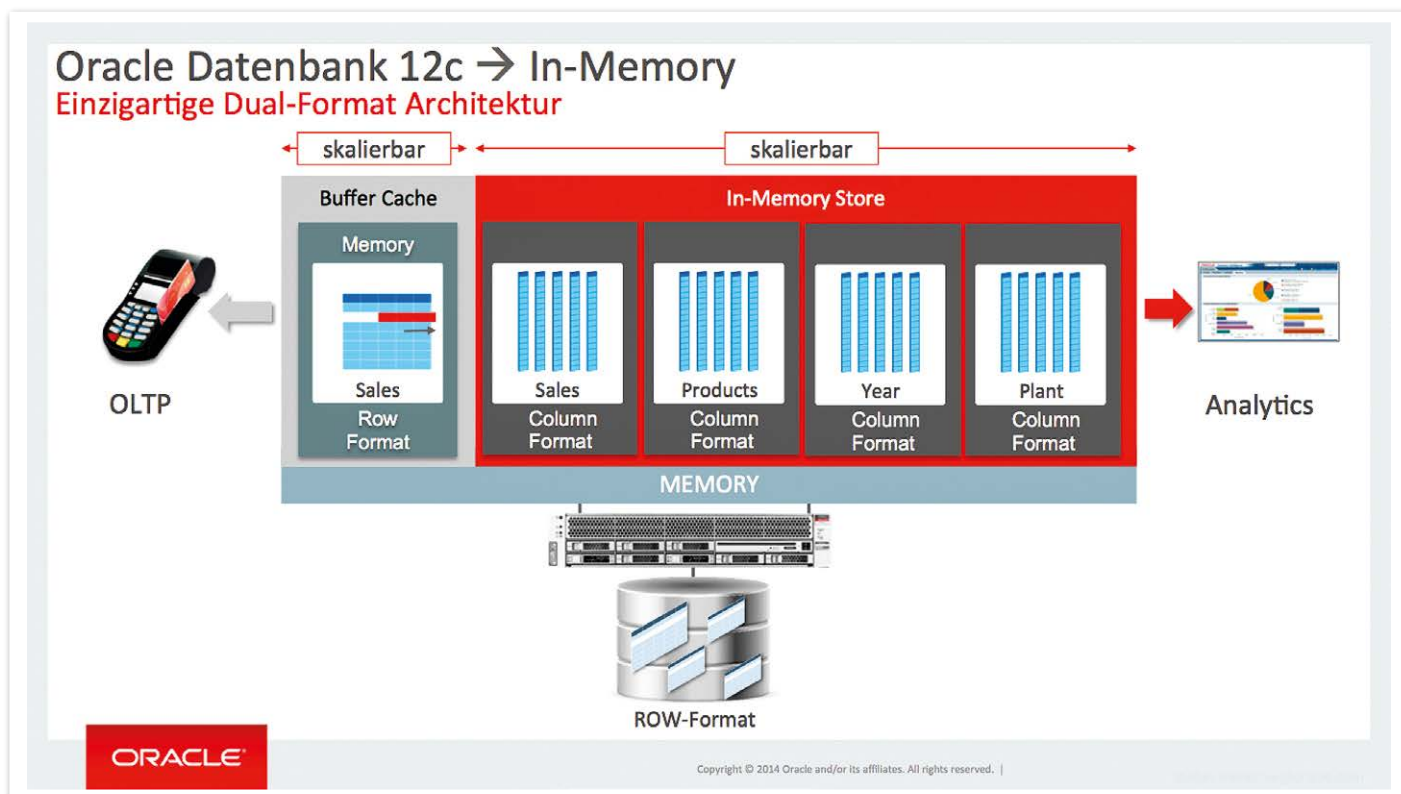


Abbildung 2: Oracle Database In-Memory

Hauptspeicher in dieser Form vor. Die Oracle-Datenbank ist somit eine hybride Datenbank, bestehend aus einem zeilenorientierten Datenbestand und einer spaltenorientierten, hoch komprimierten Repräsentation im Hauptspeicher. Damit ist die Datenbank in der Lage, alle Anforderungen hinsichtlich OLTP wie gewohnt weiterhin zu unterstützen.

Zusätzlich entstehen nun über den Column Store Verbesserungen für analytische Abfragen. Der Oracle-Datenbank-Optimizer erhält mit der Nutzung des Column Store eine weitere Möglichkeit, einen optimalen Zugriffspfad für die jeweilige Anforderung zu finden. Dies muss nicht immer der Bereich im Column Store sein; bisherige Technologien wie „UNIQUE INDEX“ oder auch „BITMAP INDEX“ können in der Performance für „Single Row Select“-Anforderungen durchaus schneller sein als aufwändige Joins über mehrere Spalten im Column Store.

Synchronisierungen im Buffer Cache infolge von Daten-Veränderungen werden innerhalb des Memory durch Transaktions-Journale sehr effektiv und performant ausgeführt. Beide Elemente des Hauptspeichers – sowohl Buffer Cache als auch In-Memory Store – lassen sich

flexibel auf die jeweiligen Anforderungen hin dimensionieren. Dabei kann die Datenbank individuell auf ihre Hauptaufgabe eingestellt werden – entweder mehr transaktional oder mehr analytische Last. Beide Elemente des Hauptspeichers erfordern keine permanente Redundanz der Objekte (siehe Abbildung 2).

Die Aktivierung dieser neuen Technologie erfolgt in zwei einfachen Schritten. Ein Parameter für die Dimensionierung des In-Memory Store in der Initialisierungsdatei der Datenbank sowie die Auswahl der Tabellen für ein Spaltenformat reichen aus. Dies bedeutet: Alle bestehenden Anwendungen können diese neue Technologie ohne Veränderung im Code auf allen traditionellen Infrastrukturen nutzen. Alle bisher nutzbaren Eigenschaften der Oracle-Datenbank wie Komprimierung, Verschlüsselung, Real Application Clusters oder Data Guard können unverändert weiter genutzt werden.

#### Fazit

Mit der neuen In-Memory-Technologie sind mit Oracle keine aufwändigen Migrationen, keine neuen Infrastrukturen oder Änderungen an der Applikation und den bestehenden Prozessen für Backup und

Recovery notwendig. Für SAP-Anwendungen wird die Freigabe dieser Technologie Ende des ersten Halbjahres 2015 erwartet. Ein unbedingter Wechsel auf eine aus Walldorf vorgegebene neue Datenbank mit allen Risiken wird damit unnötig. Es besteht kein Grund, eine zuverlässige Oracle-Datenbank auszutauschen.



Stefan Menschel  
stefan.menschel@oracle.com