

„Die Data Warehouse/BI-Nutzer werden diese Technologie lieben ...“

In Kürze bringt Oracle die In-Memory-Datenbank-Option auf den Markt. Wolfgang Taschner, Chefredakteur der DOAG News, sprach darüber mit Günther Stürner, Vice President Sales Consulting und Leiter der Business Unit Server Technologies bei der ORACLE Deutschland B.V. & Co. KG.

Welche Motivation hatte Oracle für die Entwicklung der neuen In-Memory-Datenbank-Option?

Stürner: Ich habe manchmal den Eindruck, dass die Idee der Spalten-orientierten Verarbeitung erst in den letzten drei Jahren erfunden wurde. Dies ist jedoch absolut nicht der Fall. Bereits Anfang der 1970er Jahre wurde heftig diskutiert, welche der beiden Varianten – die Satz-orientierte oder die Spalten-orientierte Speicherung von Daten – die bessere ist. Die meisten relationalen Systeme haben sich die Satz-orientierte Speicherung zu eigen gemacht, einige wenige haben die Spalten-orientierte Linie verfolgt. Beide Arten haben ihre Vor- und Nachteile. Die Idee bei der Entwicklung der In-Memory-Datenbank-Option (IMDB) war nun, ein System zu schaffen, das die jeweiligen Vorteile nutzt, ohne dass die Nachteile der Satz- beziehungsweise Spalten-orientierten Technik zum Tragen kommen. Das Ziel war also nicht weniger als die Überwindung des „Row Column“-Dilemmas.

Welche Rolle spielte dabei SAP HANA?

Stürner: Wir arbeiten schon seit etlichen Jahren an diesem Projekt. Der Auftritt von SAP HANA hat jedoch die Anstrengungen massiv beschleunigt. SAP HANA hat allerdings nicht nur die interne Priorität des Projekts erhöht, sondern unsere Datenbank-Entwickler besonders motiviert, die beste Lösung in diesem Umfeld zu schaffen. SAP hat durch eine massive Marketing-Kampagne die Themen „Columnar“ und „In Memory“ einer breiten Öffentlichkeit bekanntgemacht. Diese Vorarbeit nutzen wir gerne.

Was sind die grundsätzlichen Unterschiede zwischen Oracle-In-Memory und SAP HANA?

Stürner: Die Grundidee der Systeme ist gleich. Beide nutzen die Vorteile der Spalten-orientierten Speicherung, um bei bestimmten Operationen kurze Antwortzeiten zu erreichen. Der gravierendste Unterschied zwischen beiden Lösungen ist, dass wir eine weitere Technologie in unsere Enterprise-Datenbank integrieren, während SAP mit HANA eine Technologie bereitstellt, um die eine Enterprise-Datenbank noch gebaut werden muss. Hier hat SAP eine Menge Arbeit vor sich und wird noch viel „Blut und Tränen“ vergießen müssen, bis aus HANA eine Enterprise-Datenbank geworden ist. Unser Ziel ist es, die beste In-Memory-Columnar-Technologie anzubieten, die von allen Anwendungen – bestehende wie neu zu entwickelnde – unmittelbar ohne weiteren Aufwand genutzt werden kann. Wir werden diese Technologie mit neuen Algorithmen und der heutigen Hardware-Technologie so verknüpfen, dass völlig neue Möglichkeiten der Verarbeitung von Daten möglich sind.

Wie unterscheidet sich die neue In-Memory-Datenbank-Option von Oracle Times Ten?

Stürner: Times Ten ist vom Design her eine In-Memory-Datenbank für den High-End OLTP-Betrieb. Man spricht hier auch häufig vom Near-Realtime OLTP-Betrieb. Times Ten findet man immer dort, wo extrem viele kurze Transaktionen extrem schnell bearbeitet werden müssen. Wir sehen Times Ten in drei Einsatzszenarien: als Stand-Alone System im High-End-Transaktionsbereich, als Front-End Datenbank-Cache für eine Oracle-Datenbank (Cache Connect Option) und als In-Memory-Datenbank innerhalb unseres Exalytics-Systems. Dies wird sich auch durch die neue In-Memory-Datenbank-Option nicht ändern.

Gibt es die Möglichkeit, zwischen den beiden In-Memory-Datenbanken zu migrieren?

Stürner: Hier gibt es keinerlei Unterschiede zu den bisherigen Vorgehensweisen. Auch die Kombination von beispielsweise Oracle-Datenbank mit aktivierter In-Memory-Option in Kombination mit einem Times-Ten-Memory-Cache ist denkbar und in bestimmten Fällen absolut sinnvoll.

Nach welchem Grundprinzip arbeitet die neue In-Memory-Datenbank-Option?

Stürner: Eine Oracle-Tabelle wird als „In-Memory“ definiert und dann als Spalten-orientierte Repräsentation in einem Bereich der System Global Area im Hauptspeicher abgelegt. Die Zeilen-orientierte Repräsentation auf den Platten und im Hauptspeicher – wir sprechen hier vom Row-Cache – bleibt davon unberührt. Eine als In-Memory definierte Tabelle hat also gleichzeitig beide Eigenschaften – Zeilen- und Spalten-Orientierung. Der SQL-Optimizer ist dafür verantwortlich, welche Datenstruktur bei den jeweiligen Operationen zum Einsatz kommen soll. Bei Transaktionslast wird also der Row-Cache eine wichtige Rolle spielen und bei Lese- und Auswerte-Operationen kommt der Column-Store der jeweiligen Tabellen verstärkt zum Zuge. Dieser duale Ansatz ist einzigartig am Markt. Nicht „entweder/oder“, sondern „sowohl/als auch“ ist die Philosophie, die sich hinter dieser Implementierung verbirgt.

Für welche Anwendungen macht die neue In-Memory-Datenbank-Option Sinn?

Stürner: Zum einen werden gemischte OLTP-, Analytic/Reporting-Systeme enorm von dieser Technologie profitieren. Wir gehen hier von verbesserten Transaktionsraten bei gleichzeitiger Beschleunigung der Reports aus. Zum zweiten werden die Data Warehouse/BI-Nutzer aller Analytic/Reporting-Systeme diese Technologie lieben. Ich möchte an dieser Stelle noch einen Punkt zum Unterschied zwischen der Oracle-In-Memory-Datenbank-Option und SAP HANA erwähnen. Bei unserer Implementierung kann eine komplette Datenbank – also alle Objekte einer Datenbank – als In-Memory definiert werden. Bei SAP HANA müssen zwangsläufig alle Daten einer Datenbank im Hauptspeicher sein, egal ob sie genutzt werden oder nicht. „Alles oder nichts“ lautet bei HANA die Devise. Der Unterschied liegt in den Wörtern „kann“ und „muss“ – ein kleiner Unterschied, aber eine große Wirkung.

Welche Voraussetzungen müssen erfüllt sein, um die In-Memory-Datenbank-Option zu nutzen?

Stürner: Die IMDB-Technologie wird als Option der Enterprise Edition angeboten,

die zu lizenzieren ist. Dann können beliebig viele Tabellen, Partitionen von Tabellen, einzelne Spalten von Tabellen oder auch Materialized Views als In-Memory-Objekte definiert werden. Diese Objekte sind dann als Spalten-orientierte Repräsentation innerhalb der SGA in einer hochkomprimierten und spezialisierten Form vorgehalten. Die so definierten Objekte – das geschieht übrigens mit einem einzigen Befehl – können mit beliebigen Operationen (wie insert, update, delete) bearbeitet werden. Die transparente Nutzung aller Datenbank-Funktionen ist garantiert.

Wird für die Nutzung der In-Memory-Datenbank-Option mehr Hauptspeicher benötigt?

Stürner: Um die Vorteile der In-Memory-Technologie nutzen zu können, muss man dem System entsprechend mehr Hauptspeicher zur Verfügung stellen. Wie viel das im Einzelfall ist, hängt davon ab, wie viele Objekte als In-Memory definiert sind. Wichtig bei der Abschätzung ist auch die Kompressionsrate, bei der ein Faktor zwischen drei und zehn möglich ist.

Wie verhalten sich Datenbank-Optionen wie RAC, Verschlüsselung, Data Guard etc. beim Einsatz der In-Memory-Datenbank-Option?

Stürner: Eines der Hauptziele bei der Entwicklung der IMDB-Technologie war die tiefe und optimale Integration in das Oracle-Datenbank-Ecosystem. Alle bekannten Funktionen müssen ohne Einschränkungen und völlig transparent funk-



Günther Stürner demonstriert die Skalierbarkeit der In-Memory-Datenbank



Wichtig ist die Abschätzung der Kompressionsrate

tionieren. Oracle 12.1.0.2. als Ergebnis ist eine Enterprise-Datenbank mit einer der besten In-Memory-Columnar-Technologien. Es ist keine isolierte und einschränkende Technologie und vor allem kein Balkon, um den in den nächsten Jahren erst das Haus gebaut werden muss. So kann die In-Memory-Datenbank-Option eine RAC-Umgebung optimal nutzen, indem ein In-Memory-Objekt in allen SGAs der entsprechenden RAC-Knoten als Kopie abgelegt ist. Bei sehr großen Objekten kann ein solches Objekt über eine gesamte RAC-Farm verteilt/segmentiert sein. Das ist Scale-out-Architektur vom Feinsten. Auch die Verschlüsselung der Daten mit ASO und In-Memory ist kein Problem, der Einsatz von Data Guard ist mit allen Varianten ohne Einschränkungen möglich. Genauso steht diese Option für alle Oracle-Enterprise-Edition-Plattformen zur Verfügung.

Wie skalierbar ist eine In-Memory-Datenbank?

Stürner: Eine Oracle-Datenbank mit In-Memory-Option hat die gleichen Eigenschaften in Bezug auf Skalierbarkeit wie eine bisherige Oracle-Datenbank. Die Nutzung von mehr CPUs, mehr Hauptspeicher oder mehr Knoten innerhalb einer RAC-Umgebung ist vollständig gegeben. Ich muss mich wiederholen: Die In-Memory-Datenbank-Option ist in das Oracle Ecosystem tief integriert. Was bisher ging, geht auch dann, wenn man mit der In-Memory-Option arbeitet.

Wie sichert man persistente Daten beim Einsatz der In-Memory-Option?

Stürner: Die Daten auf den Speichermedien werden auch bei der Nutzung der

In-Memory-Option – wie bisher – in den bekannten Oracle-Storage-Formaten gehalten. Diese sind die Basis für alle möglichen Varianten des Backups oder eines eventuell notwendigen Recovery. Die In-Memory-Datenbank-Option schränkt die Nutzung und die Prozesse im Bereich „Backup“ in keiner Weise ein.

Gibt es Möglichkeiten, die In-Memory-Datenbank-Option wieder rückgängig zu machen?

Stürner: Wie bereits erwähnt, können eine Tabelle, Partitionen einer Tabelle, einzelne Spalten einer Tabelle oder eine Materialized View als In-Memory-Objekt definiert sein. Dabei sind die Daten eines solchen Objekts Spalten-orientiert im Hauptspeicher abgelegt, was mit dem Befehl „alter table xyz inmemory“ geschieht. Dies lässt sich jederzeit und ohne Probleme durch den Befehl „alter table xyz no inmemory“ wieder rückgängig machen. Die Spalten-orientierte Repräsentation im Hauptspeicher wird für dieses Objekt unmittelbar gelöscht, während die Daten dieses Objekts weiterhin als Zeilen-Repräsentation im Row-Cache beziehungsweise auf den Platten vorliegen. Es ist keine Konvertierung, kein Umspeichern von Formaten, kein Entladen oder Neuladen erforderlich. Hier tritt das duale Konzept in seiner vollen Schönheit zu Tage.

Kann eine mit der In-Memory-Option eingerichtete Tabelle auch auf eine Datenbank ohne In-Memory-Option übertragen werden?

Stürner: Kein Problem! Wie gesagt, jedes Objekt, das als In-Memory-Objekt definiert ist, behält seine Zeilen-orientierte Eigenschaften innerhalb der Datenbank und auf den Platten und kann somit mit allen bekannten Möglichkeiten bearbeitet werden. Auch hier gilt, wie bereits mehrfach ausgeführt: Alles, was man heute ohne In-Memory-Option machen kann, ist auch mit dieser Option durchführbar.

In welche Richtung wird sich die In-Memory-Datenbank-Option weiterentwickeln?

Stürner: Mit Verfügbarkeit der In-Memory-Option im zweiten Halbjahr 2014 werden

wir eine Vielzahl von wichtigen und interessanten Funktionen bereitstellen. Nicht nur die Speicherung der Daten in Spalten-orientierter Form im Hauptspeicher macht den Unterschied, sondern auch die Algorithmen für die Komprimierung, für die Join- und Gruppierungs-Operationen etc. sind entscheidend für die optimale Verarbeitung. Selbst die Integration dieser Datenstrukturen und Algorithmen mit den Hardware-Systemen und deren spezifischen Fähigkeiten ist bereits weit fortgeschritten, aber bei Weitem noch nicht vollständig ausgereizt. Man kann erwarten, dass dieser ersten Version noch viele weitere Entwicklungs-Iterationen folgen werden.

Steht die In-Memory-Datenbank-Option auch im SAP-Umfeld zur Verfügung?

Stürner: Wir arbeiten seit vielen Jahren mit SAP sehr eng und vertrauensvoll zusammen. Bei jeder neuen Datenbank-Version entwickelt ein kleines Team bestehend aus Entwicklern und Verantwortlichen von



Zur Person: Günther Stürner

Günther Stürner arbeitet bereits seit September 1985 für Oracle; sein beruflicher Werdegang begann als Sales Consultant Deutschland. Von 1987 bis 1993 widmete er sich dem Aufbau der Abteilung „Strategisch Technische Unterstützung“ (STU) und war anschließend sechs Jahre lang Leiter des Oracle SAP Competence Centers sowie des Oracle SAP Entwicklungszentrums. Er ist heute Vice President Server Technologies und Sales Consulting. Günther Stürner hat mehrere Fachbücher zur Oracle-Datenbank und zu SQL geschrieben sowie zahlreiche Fachartikel veröffentlicht.