Software in Silicon

Martin Müller
Oracle Deutschland

Schlüsselworte

In-memory, database, SPARC M7, acceleration,

Einleitung

Mit der in-memory Option stellt Oracle zusätzlich zum gewohnten zeilenorientierten Speicherformat einen parallelen zeilenweisen Zugriffspfad auf Daten im Hauptspeicher zur Verfügung. Dieses Format ist zusätzlich komprimiert, so kann mit nur geringem Einsatz von Hauptspeicher große Vorteile erzielt werden.

Die hier beschriebenen neuen Funktionen der zukünftigen SPARC M7 CPU dienen der Beschleunigung des spaltenweisen Zugriffs, und helfen die Folgen *logischer* Fehler zu minimieren. Mit diesen Erweiterungen werden Teile der Software direkt im Silizium implementiert, daher „Software in Silicon“

„Application Data Integrity“

Ein weitverbreitete Klasse von Sicherheitslücken oder Fehlerquellen von Softwareprodukten sind Referenzierungsfehler und „Buffer overflows“. Hier wird von einem Referenzierungsfehler gesprochen, wenn eine Zeigervariable weiter genutzt wird, obwohl der zugehörige Speicher nicht mehr verwendet sollte oder wenn im Falle einer Shared Memory Anwendung ein Thread einen Speicherbereich nutzt, den ein anderer Thread angelegt hat, und der Programmierer das so nicht vorgesehen hat. Ein „Buffer overflow“ entsteht, wenn absichtlich versucht wird, über einen Ein- und Ausgabepuffer hinaus gelesen wird und in dem Programm keine Sicherheitsvorkehrungen getroffen werden. Durch „Application Data Integrity“ ist es erstmals möglich diese Klasse von Fehlern auch im produktiven Einsatz zu überwachen und zu erkennen.

Mithilfe einer neuen Funktionalität der M7 CPU kann jeder Hauptspeicherlokation 64byte-weise eine Markierung (oder „Farbe“) zugewiesen werde, sowie jedem Instruktionsstrom. Die CPU überprüft nun transparent ob diese beiden Markierungen übereinstimmen

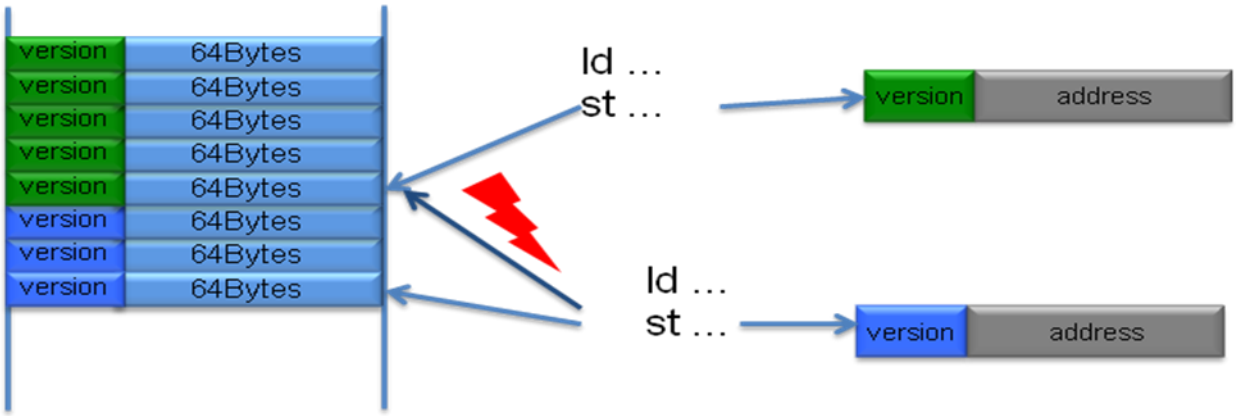

Abb. 1: „Einfärben“ von Hauptspeicher und Threads

Solange die „Farbe“ des Threads/Instruktionsstromes und der zugegriffenen Hauptspeicherlokation übereinstimmen, greift die CPU nicht in die Ausführung ein. Sollten aber die beiden Farben voneinander abweichen, wird ein Trap ausgelöst. Dieser Trap kann dann entweder nur eine Fehlermeldung ausgeben, oder eine andere Aktion auslösen.

Die Nutzung dieser Funktion ist natürlich optional, kein Programm **muß** davon Gebrauch machen bzw. auf M7 angepasst werden.

Beschleunigung von Datenbank In-Memory Operationen

Durch die Oracle „in-memory“ Option wird der traditionellen zeilenweisen Darstellung der Daten im Hauptspeicher eine spaltenweise Repräsentation hinzugefügt, die durch ihre Art der Implementirtung und Kompression auch schon mit einem geringe zusätzliche Einsatz von Hauptspeicher erhebliche Geschwindigkeitsvorteile bei analytischen Abfragen realisieren kann. Diese spaltenweise Darstellung basiert auf Bitmustern, die auf spaltenweise Einträge verweisen. In vielen Fällen sind diese Bitmuster sehr kurz, und der Einsatz von SIMD Operationen erzielt schon gute Geschwindigkeitssteigerungen.

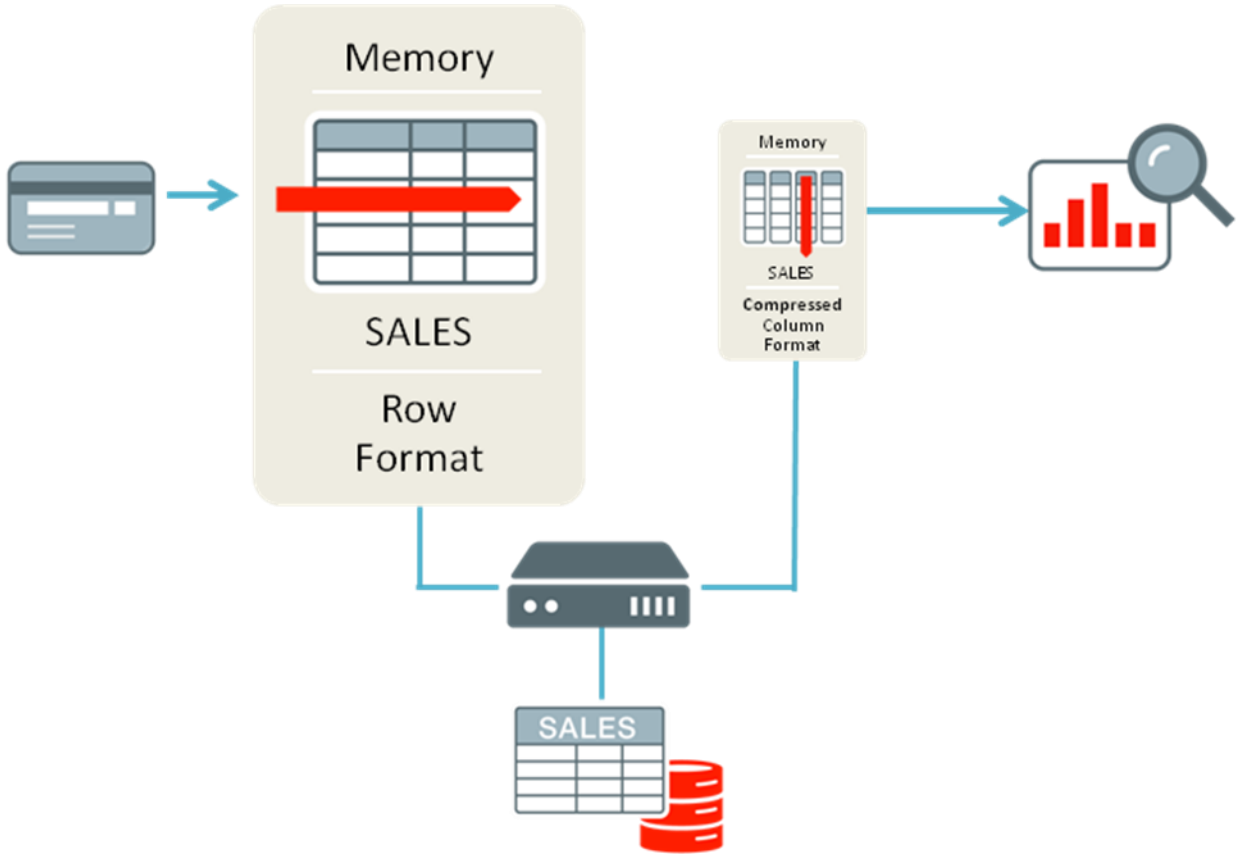

Abb. 2: Hauptspeicherbasiertes Spaltenformat

Die „Software in Silicon“ Eigenschaften der M7 CPU erlauben auf einer komprimierten Darstellung dieser Spaltendarstellung zu arbeiten. Die Kompression ermöglicht mit dem gleichen Einsatz von Hauptspeicher mehr Daten in dieser Darstellung zu halten, durch den Einsatz der spezialisierten Datenbankbeschleuniger der M7 CPU kann fast transparent auf der Spaltendarstellung gearbeitet werden. Zusätzlich kann der Beschleuniger mit größeren Bitmustern als die SIMD Operationen arbeiten, so daß auch umfangreichere Abfragen beschleunigt abgearbeitet werden können.

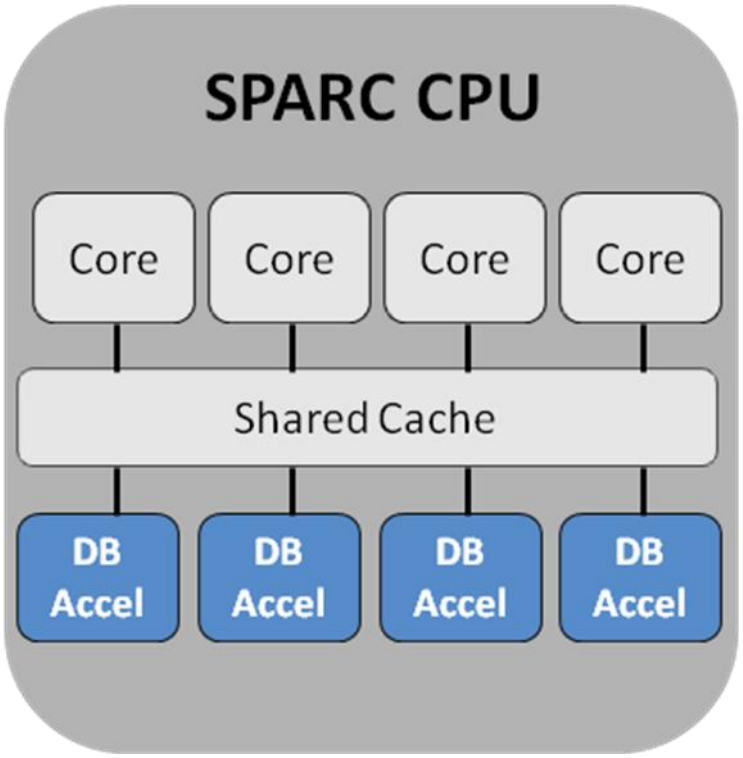

Abb. 3: Datenbankbeschleunigung

Die Beschleunigung wird durch eine Erweiterung des „Main Memory Controllers“ realisiert: ein CPU Kern kann eine spezielle Einheit programmieren, direkt nach Mustern im Hauptspeicher zu suchen oder einen gewissen Hauptspeicherbereich zu dekomprimieren. So werden nur die gewünschten Ergebnisse an den Kern weitergegeben, und die Caches werdennicht durch ungewünschte Daten „verschmutzt“

Jedem der vier Memory Controller einer M7 CPU stehen zwei dieser Einheiten (genannt „query engine“) zur Seite, jede enthält vier Pipelines zur Abarbeitung dieser Hautpspeicheroperationen. So kann im Durchschnitt jeder Kern einer M7 CPU eine Beschleunigungseinheit nutzen

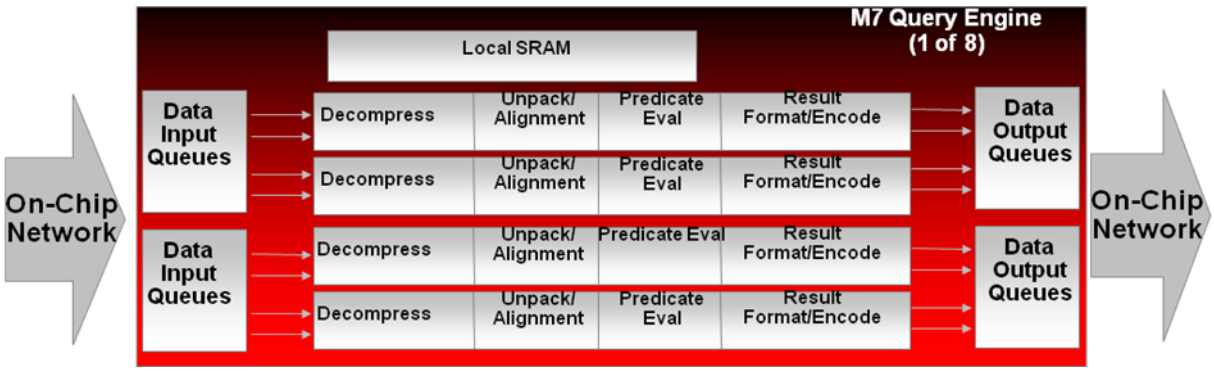

Abb. 4: Schematische Darstellung der Query Engine

Kommunikation zwischen Clusterknoten

Ein Teil der Systeme, die die M7 CPU enthalten werden, wird auch von einem neuen Serverinterconnect Gebrauch machen. Kleinere Systeme mit bis zu 8 M7 CPUs benötigen keine speziellen Kommunikationsbausteine zwischen den CPUs. Systeme mit mehr als 8 M7 CPUs kommunizieren intern über einen neuen Serverinterconnect, der nicht nur die Kohärenz des Hauptspeichers über das gesamte System sicherstellt. (Über dieses System muß nicht notwendigerweise ein einziges Betriebssystemabbeild aufgespannt werden, es können genauso mehrere logische Domains mit eigenen Betriebssystemabbildern betrieben werden)

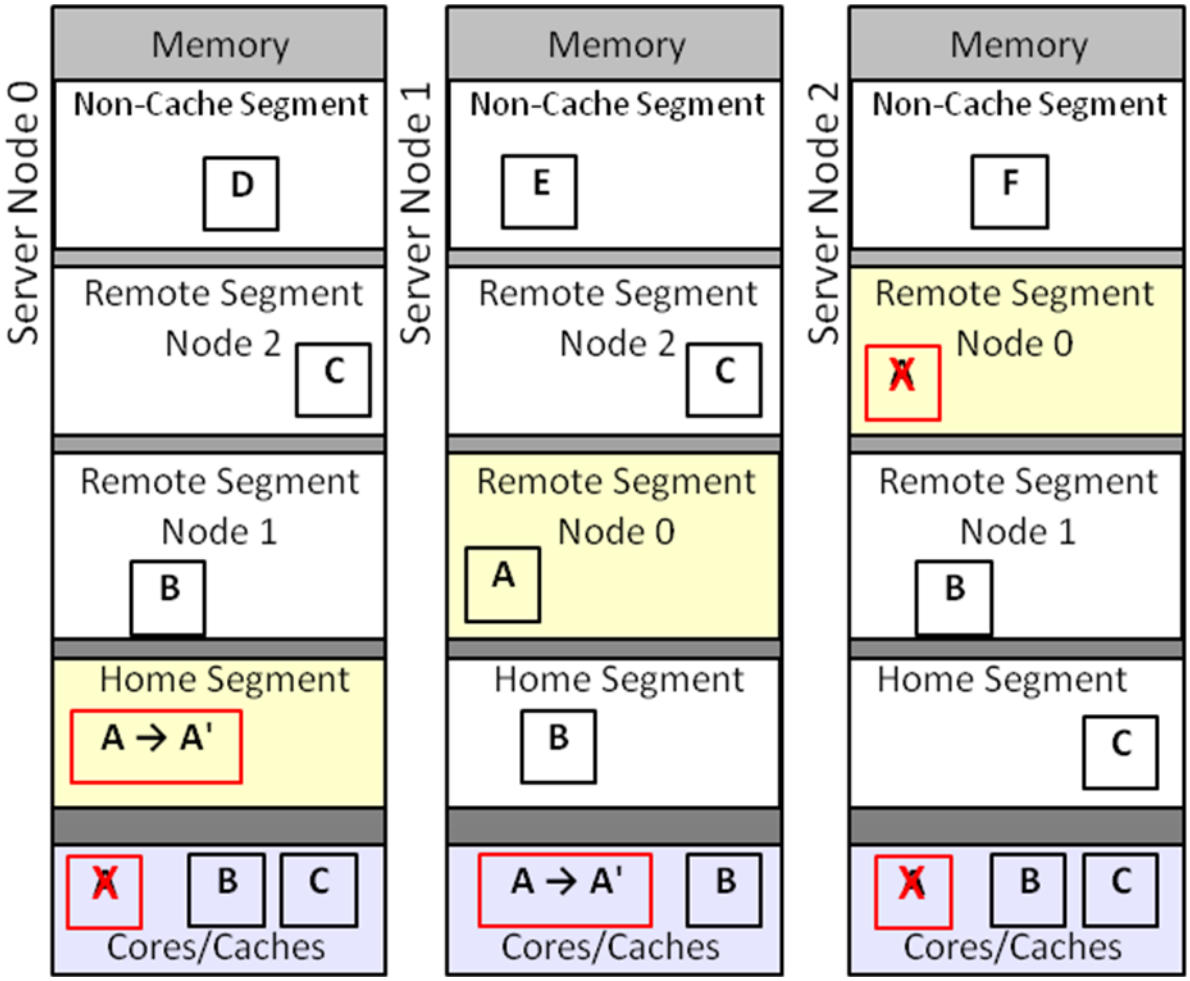

Abb. 5: Details zur Kommunikation zwischen Clusterknoten

Der neue Interconnect kann auch zur Konfigration einer sehr schnellen Kommunikation zwischen unterschiedlichen SMP Systemen genutzt werden. Dazu werden Teilbereiche des Hauptspeichers der miteinander verbundenen Systeme in die anderen Clusterknoten eingeblendet. Dies stellt in Anbetracht der zuvor geschilderten „Application Data Integrity“ Technologie eine besondere Herausforderung dar, da die Markierungen die im Rahmen der „Application Data Intergrity“ vergeben werden von der Clusterverbindung respektiert werden müssen.

Die Abbildung zeigt schematisch drei verschiedene Server, die als Cluster miteinander verbunden sind. Im Falle der „non-cacheable“ Bereiche erfolgt jeder Zugriff transparent auf den jeweiligen

Heimatknoten mit Hilfe des Interconnects. „Cacheable“ Bereiche werden 64byte-weise auf allen Clusterknoten zwischengespeichert, so daß die Latenzzeit beim Zugriff auf einem entfernten Knoten genauso groß wie die lokale Zugriffszeit ist. Die Änderung einer Cachezeile eines entfernten Knotens invalidiert alle Kopien dieser Cachezeile in anderen Caches, aktualisiert die Cachezeile im Hauptspeicher auf dem Heimatknoten und invalidiert entfernte Kopien. (Die Hauptspeicherkopie im Hauptspeicher des ändernden Caches wird erst beim „victimizing“ der Cachezeile durchgeführt)

Kontaktadresse:
Martin Müller
Oracle Deutschland
Hamborner Str. 51
40472 Düsseldorf
Telefon: +49 211 74839-853
E-Mail: martin.x.mueller@oracle.com
Internet: www.oracle.com