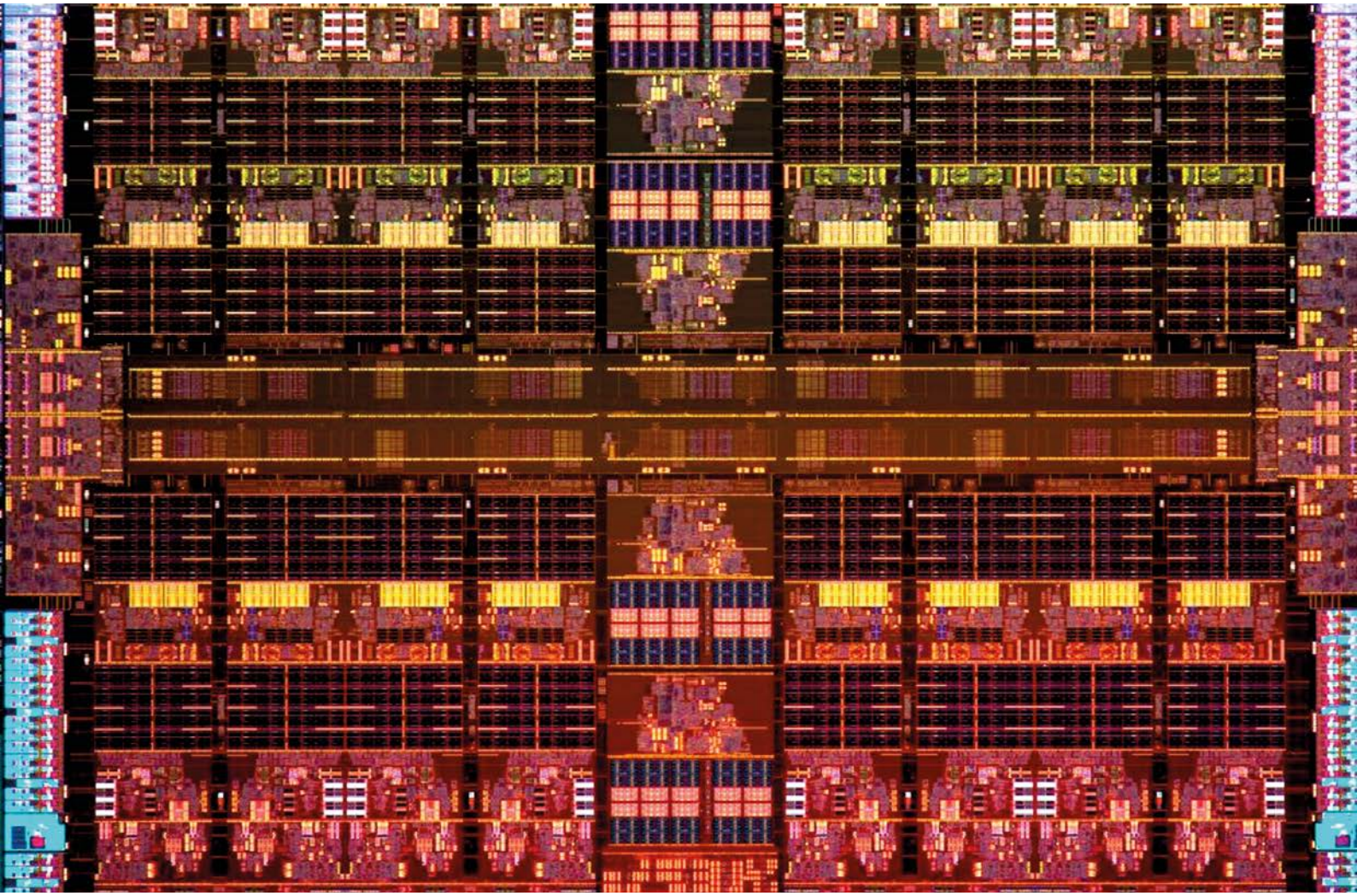

Sicherheit und SQL in Silizium

Franz Haberhauer, ORACLE Deutschland B.V. & Co. KG

Nach der Übernahme von Sun entschied Oracle, die SPARC-Prozessoren nicht nur traditionell weiterzuentwickeln – also mit dem Ziel, den Durchsatz durch den Prozessor und die Schnelligkeit bei der Bearbeitung eines Befehlsstroms („Single Thread Performance") generell zu steigern –, sondern darüber hinaus gezielt Software direkt durch neue Prozessor-Funktionalitäten zu unterstützen.

Über diese „Software in Silicon" lassen sich in spezifischen Bereichen deutlich größere Performance-Sprünge und außerdem weitere Mehrwerte erzielen. Dabei standen Anforderungen aus der Oracle-Datenbank-Entwicklung im Fokus; insbesondere die neuen Sicherheitsmechanismen kommen aber ganz allgemein zum Tragen: Damit werden ganze Kategorien von Sicherheitsproblemen direkt vom System unterbunden – ganz unten im Stack transparent für die höheren Schichten. Rechtzeitig zur Oracle OpenWorld 2015 wurden nun die neuen T7- und M7-Systeme mit dieser Technologie im SPARC-M7-Prozessor verfügbar.

Der SPARC M7 ist bereits der sechste Prozessor, den Oracle seit der Übernahme von Sun in den Markt einführt. Mit einer Verdopplung der Anzahl an Kernen von 16 auf 32, einer neuen Pipeline, höherer Taktung, größeren Caches, schnellerem Hauptspeicher und weiteren Verbesserungen wird gegenüber dem T5 der

Durchsatz um einen Faktor 2,5 bis 3 und die Single-Thread-Performance um 30 bis 40 Prozent signifikant gesteigert.

Hinzu kommen die „Software in Silicon"-Funktionalitäten: „SQL in Silicon" mit den Data Analytics Accelerators (DAX), die die In-Memory-Option in der aktuellen Version 12*c* unterstützen, sowie „Security in Silicon" mit Silicon Secured Memory, das ab Solaris 11.3 fehlerhafte Speicherzugriffe (insbesondere Puffer-Überläufe) entdecken und verhindern kann. Damit wird eine Vielzahl von Sicherheitsproblemen – unter anderem auch das berühmt-berüchtigte Heartbleed – direkt vom System automatisch und ohne Änderungen des Quellcodes von Anwendungen abgewehrt.

Zudem unterstützt diese Technologie die Software-Qualitätssicherung. Durch Programmierfehler fehlgeleitete schreibende Speicherzugriffe können nicht mehr so einfach Speicherbereiche korrumpieren. Eine derartige Datenkorruption fiel bislang – wenn überhaupt – erst bei einem späteren lesenden Zugriff auf. Zu diesem Zeitpunkt war dann aber die Ursache oft nur noch mühselig zu diagnostizieren. Bei Silicon Secured Memory wird dagegen bereits der fehlerhafte Zugriff abgefangen und das Programm abgebrochen, bevor es zu einer Korrumpierung kommt. Ein Beta-Kunde konnte so in einer Anwendung gleich mehrere Fehler, die sich schon seit geraumer Zeit im Code befanden, lokalisieren und beseitigen – und zwar Plattform-übergreifend und nicht nur SPARC/Solaris-spezifisch.

SQL in Silicon

Die spaltenorientierte Anordnung von Daten im Hauptspeicher, die in der Datenbank 12*c* mit der In-Memory-Option implementiert wurde, bietet neben den bekannten Vorteilen bei der Abarbeitung analytischer Abfragen auch ein größeres Potenzial für die effiziente Ausführung von Teilen der Anfragen direkt im Prozessor als die zeilenorientierte *[1]*. Wesentlich ist dabei die kompaktere und einfacher strukturierte Anordnung von Daten, auf die in Scans zugegriffen wird.

Die erste Form der Nutzung spezieller CPU-Funktionalitäten in der In-Memory-Option war die Nutzung der SIMD- und Vektor-Erweiterungen, die sich heute in vielen CPUs finden – etwa die SSE- und AVX-Erweiterungen auf Intel-CPUs oder VIS auf SPARC. Diese Erweiterungen waren ursprünglich für Grafik- und Multimedia-Anwendungen entwickelt worden, werden aber inzwischen unter anderem auch im High-Performance-Computing genutzt, um während eines Prozessor-Takts dieselbe Instruktion parallel gleich auf einer Reihe von Daten auszuführen (Single Instruction – Multiple Data) oder einen Daten-Vektor über eine Pipeline effizient zu bearbeiten. Die Oracle-Datenbank nutzt diese Mechanismen insbesondere für effiziente, schnelle Vergleiche beim Scan von Spalten, sofern die Daten in geeigneter Form vorliegen.

Der In-Memory Column Store besteht analog zu den Extents in Tablespaces auf Platte aus mehreren In-Memory-Compression-Units (IMCUs). Wie bereits aus der Benennung ersichtlich, spielt hier Kompression eine große Rolle: Dabei geht es heutzutage nicht mehr nur darum, Platz bei der Speicherung zu sparen (um mehr Daten In-Memory vorhalten zu können), vielmehr sind Scans und Filter auf komprimierten Daten oft schneller, weil die Bandbreite zum Speichermedium der limitierende Faktor ist.

Für In-Memory kommt ein ganzes Spektrum an Kompressionsverfahren zum Einsatz – gesteuert durch das Schlüsselwort „MEMCOMPRESS" als Subclause des Attributs „INMEMORY". Voreingestellt ist die Option „FOR QUERY LOW" mit Techniken wie „Dictionary Encoding", „Run Length Encoding" und „Bit-Packing", womit Anfragen direkt gegen komprimierte Spalten ausgewertet und nur die Daten, die für die Ergebnismenge erforderlich sind, dekomprimiert werden.

Bei der Option „FOR CAPACITY LOW" kommt zusätzlich „OZIP" zum Einsatz, ein Bitmuster-orientiertes, für die Oracle-Datenbank optimiertes Verfahren, das insbesondere eine sehr schnelle Dekompression erlaubt. Die Optionen „FOR CA-

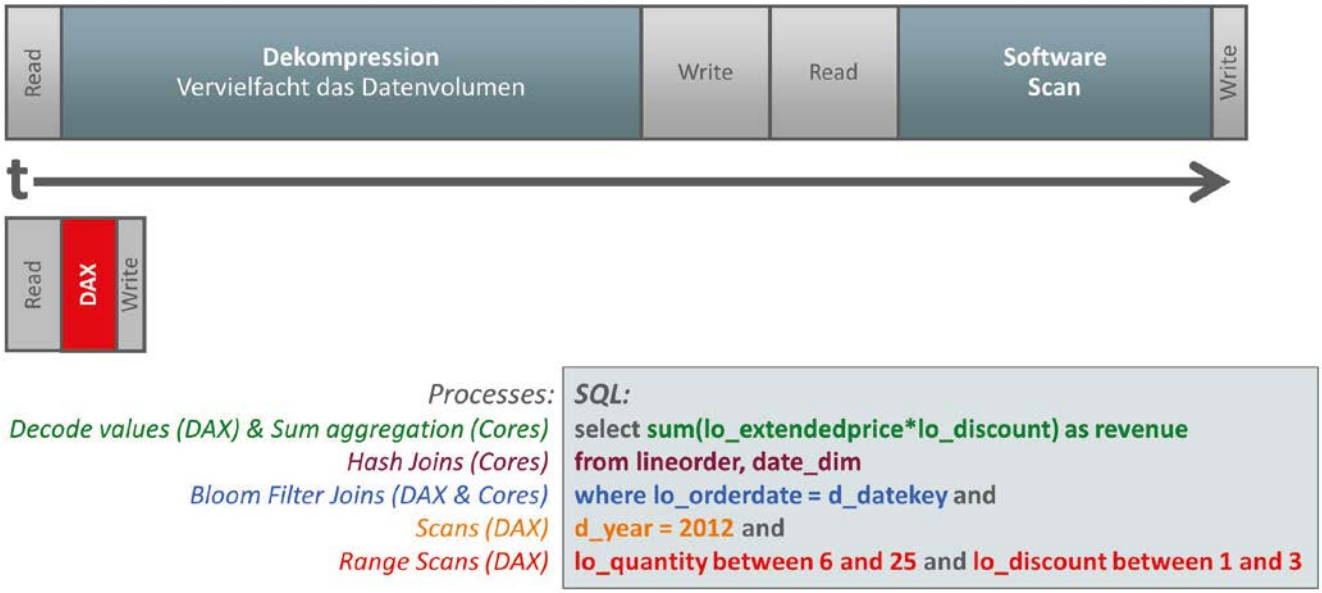

Abbildung 1: Bearbeitung einer In-Memory-Query unter Nutzung eines SPARC-M7-Data-Analytics-Accelerator (DAX)

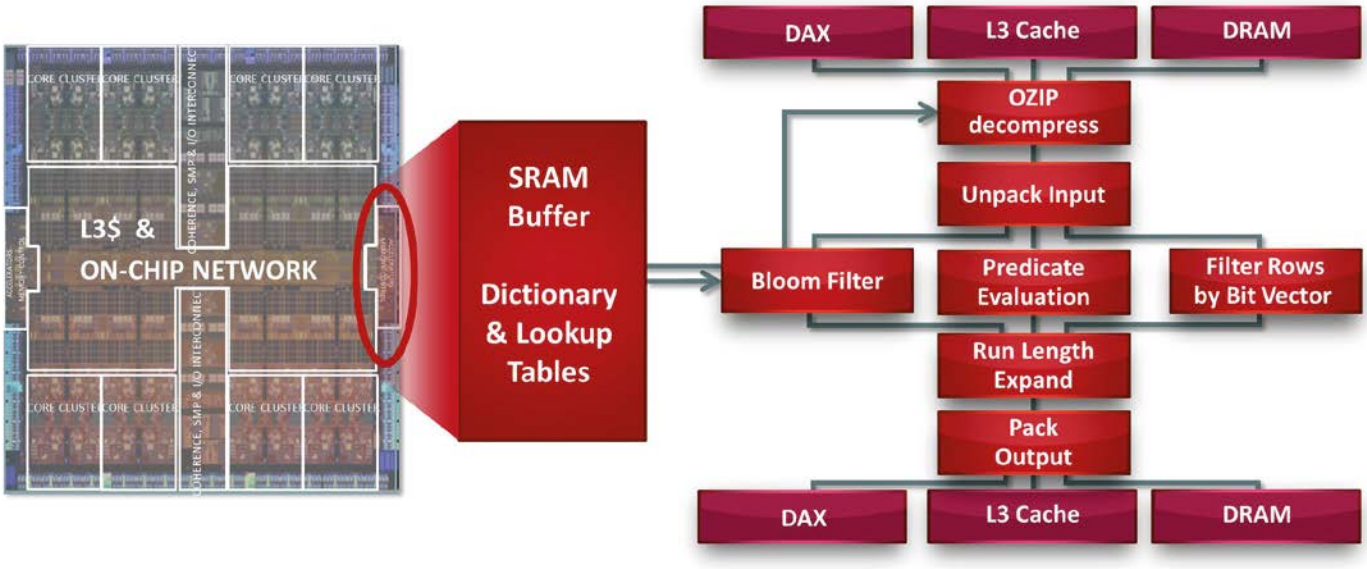

Abbildung 2: Architektur eines SPARC M7 Data Analytics Accelerator (DAX)

PACITY" erfordern für die Auswertung von „WHERE"-Klauseln die Dekompression der Daten, bieten dafür aber eine stärkere Kompression.

Für einzelne Spalten einer Tabelle können unterschiedliche Kompressionsstufen gewählt oder auch die Kompression ganz abgeschaltet werden. Die Dekompression ist zwar schnell, benötigt aber signifikant CPU-Leistung: Um Daten mit der vollen Datenrate des Hauptspeichers zu dekomprimieren, sind etliche konventionelle Kerne erforderlich. Hier setzt nun Software in Silicon an. Die SPARC-M7-CPU verfügt über 32 Decompress-Engines, die eine Dekompression mit der Datenrate des Hauptspeichers erlauben. Ein weiterer Vorteil liegt darin, dass die Daten nach der Dekompression nicht erst wieder in den Hauptspeicher zurückgeschrieben und dann zur weiteren Auswertung erneut daraus gelesen werden müssen, sondern direkt in den Data Analytics Accelerators (DAX) auf der CPU weiterverarbeitet werden können (Fused Execution). Dabei werden insbesondere Selektionsprädikate in Scans sowie Bloom-Filter unterstützt *(siehe Abbildung 1)*.

Bereits ein einzelner DAX erlaubt die direkte Hardware-Unterstützung für wesentlich komplexere Operationen, als es mit einfachen SIMD- oder Vektoroperationen möglich ist *(siehe Abbildung 2)*. Darüber hinaus kann die Ausgabe eines DAX wiederum direkt als Eingabe in einen weiteren DAX dienen, wobei die Kommunikation direkt über das schnelle On-Chip-Netzwerk läuft *[2]*. Die Datenbank nutzt ab der Version 12.1.0.2.13 (und Patches *[3]*) die 32 DAX-Pipelines auf jeder M7-CPU, die insgesamt weniger als ein Prozent der Chip-Fläche einer CPU ausmachen, aber leistungsmäßig allein bei der Dekompression einem Äquivalent von 64 Cores entsprechen. Bei einem komplexen Benchmark mit mehr als 2.300 analytischen Anfragen auf einem Real Cardinality Database (RCDB) Star-Schema lieferte ein T7-1-Server mit einer M7-CPU mit 32 Kernen gegenüber einem x86-Server mit zwei Intel Haswell CPUs (E5-2699 v3) mit je 18 Kernen mehr als den fünffachen Durchsatz *[4, 5]*. In diesem Benchmark wurde als In-Memory-Kompressionsoption „FOR QUERY HIGH" genutzt, die für SQL in Silizium generell ein guter Ausgangspunkt ist, da hierbei auf der M7-CPU „OZIP" für die Kompression in Verbindung mit „Fused Execution" zum Tragen kommt und eine gute Kompression erreicht wird (im Benchmark Faktor sechs bis zehn).

Security in Silicon

Security in Silicon hat zwei Aspekte: zum einen die Hardware-Unterstützung für Verschlüsselung, die Sun bereits auf dem UltraSPARC T1 eingeführt und seither mit jeder Prozessor-Generation weiterentwickelt hat, und zum anderen Silicon Secured Memory, worüber der neue SPARC M7 fehlerhafte Speicherzugriffe erkennen und unterbinden kann.

Seit der SPARC-T1-CPU wurde mit jeder Generation das Spektrum an unterstützten Verschlüsselungs-Algorithmen erweitert. Jeder der 32 Kerne des SPARC M7 verfügt über eine Krypto-Einheit. Sie ist nicht als Co-Prozessor implementiert, sondern effizient in der Pipeline. Die Krypto-Beschleuniger werden über das Crypto-Framework im Oracle-Solaris verwaltet – darüber kann insbesondere transparent für Anwendungen ausgewählt werden, welche Krypto-Algorithmen konkret verwendet werden sollen.

Mit fünfzehn Verschlüsselungs-Algorithmen (AES, Camellia, CRC32c, DES, 3DES, DH, DSA, ECC, MD5, RSA, SHA-1, SHA-224, SHA-256, SHA-384 und SHA-512) sowie der Generierung von Zufallszahlen wird ein breiteres Spektrum als durch andere CPUs unterstützt. Insbesondere werden damit die im Oracle-Software-Stack gängigen Krypto-Algorithmen abgedeckt. Die Implementierung ist so effizient, dass auf einer T7-1 für einen AES-128-CBC-verschlüsselten Datenstrom von 8 GB/s vom Netzwerk eingehend über zehn 10-Gigabit-Ethernet-Schnittstellen und abgehend auf Network Attached Storage (NAS) über weitere zehn 10-GbE gerade einmal 19 Prozent einer M7-CPU für die Entschlüsselung und erneute Verschlüsselung benötigt werden. Damit geht eine höhere Sicherheit aufgrund durchgängiger Verschlüsselung praktisch nicht mehr zu Lasten der Performance.

Oracle hat mit zwei T7-1 mit jeweils einem M7-Prozessor einen neuen Rekord

beim SPECjEnterprise2010-Benchmark für Applikations-Server mit 1 bis 4 Chips erzielt (eine T7-1 als J2EE-Server und eine als Datenbank-Server) – und zwar mit Oracle Advanced Security Transparent Data Encryption (TDE) zur Verschlüsselung der Datenbank und einer über Secure JDBC verschlüsselten Netzwerk-Verbindung. Damit wurden 25.093 EjOPS erreicht – gerade einmal knapp 3 Prozent weniger als ganz ohne Verschlüsselung. Ohne Verschlüsselung schafften zwei IBM S824 mit je vier Power 8 CPUs 22.543 EjOPS und zwei x86-Server mit je zwei Intel Haswell (E5-2699 v3) CPUs 21.504 EjOPS – also jeweils mehr als 10 Prozent weniger als das SPARC-System mit Verschlüsselung *[6]*. Nicht nur Software von Oracle nutzt auf SPARC-CPUs transparent diese Hardware-Unterstützung für Verschlüsselung, sondern beispielsweise auch die aktuelle Version des IBM Global Security Kit (GSKit), das als Security Framework in vielen IBM-Software-Produkten genutzt wird *[14]*.

Silicon Secured Memory

Viele Fehler in Programmen, die in klassischen Programmiersprachen mit expliziter Speicherverwaltung wie C oder C++ geschrieben sind, resultieren aus fehlerhaften Speicherzugriffen innerhalb von Prozessen, etwa durch das Lesen oder Schreiben über Puffergrenzen hinaus oder auf nicht allokierte oder bereits wieder freigegebene Speicherbereiche. OpenSSL-Heartbleed oder Venom sind berühmt-berüchtigte, sicherheitskritische Beispiele, aber auch viele schwer zu diagnostizierende Programmabstürze haben derartige Ursachen. Software in Silicon auf dem SPARC M7 kann solche Szenarien durch eine Validierung von Speicher-Zugriffen in der Hardware verhindern.

Es gibt Entwicklungsumgebungen, die solche Mechanismen in der Software implementiert haben. Deren Einsatz in produktiven Umgebungen scheitert aber am zu großen Overhead. Silicon Secured Memory (SSM), das vor der OpenWorld 2015 weniger prägnant als Application Data Integrity (ADI) bezeichnet worden war, lässt sich dagegen transparent für Anwendungen produktiv ohne Leistungsverlust in Echtzeit einsetzen. Damit kann die Systemsicherheit signifikant gesteigert werden – ein Angriff gegen ein System mit einem nicht gefixten Heartbleed-Bug wäre an Silicon Secured Memory gescheitert.

SSM kennzeichnet Speicherbereiche mit „Farben" (vier dedizierten Bits in Speicheradressen) und prüft beim Zugriff, ob die Farbe der verwendeten Adresse dazu passt. Passt sie nicht, wird ein Trap getriggert, der in der Regel in einem Programm-Abbruch resultiert. Die Verwaltung der Farben übernimmt beispielsweise die Bibliothek „libadimalloc(3LIB)", die „malloc()", „free()" etc. implementiert. Benachbarte und wieder freigegebene Speicherbereiche erhalten unterschiedliche Farben, was bereits eine Vielzahl an Fehlerszenerien erfasst. Selbst zufällige Speicherzugriffe werden mit hoher Wahrscheinlichkeit erfasst.

Zur Nutzung wird das Programm nur gegen die Bibliothek gebunden, gegebenenfalls auch dynamisch durch Setzen der Umgebungsvariablen „LD_PRELOAD". Änderungen am Quellcode sind nicht erforderlich. Für die Entwicklung gibt es in Solaris Studio 12.4 beziehungsweise 12.5, das Anfang Dezember als Beta freigegeben wurde *[7]*, eine weitergehend instrumentierte Bibliothek „libdiscoveradi". Dadurch können fehlerhafte Zugriffe in einer Entwicklungsumgebung analysiert werden.

Ein empfehlenswertes White Paper *[8]* zeigt den praktischen Einsatz und darüber hinaus, wie ein eigener Memory-Allocator für Silicon Secured Memory instrumentiert werden kann. Der Blog „Hardening allocators with ADI" *[9]* geht ausführlicher auf Angriffsvektoren und Härtung bei der Speicherverwaltung ein. Raj Prakash vertieft in seiner Blog-Reihe *[10]* das Thema „Silicon Secured Memory" weiter. Für Entwickler, die nicht selbst über ein SPARC-M7-basiertes System verfügen, besteht die Möglichkeit, Silicon Secured Memory in der Software in Silicon Cloud *[11]* kennenzulernen.

Systeme mit dem SPARC-M7-Prozessor

Rechtzeitig zur Oracle OpenWorld 2015 wurden die neuen Server mit dem M7-Prozessor verfügbar: T7-1, T7-2, T7-4, M7-8 und M7-16 – eine Server-Linie mit Systemen von einem Sockel bis zu 16 Sockeln *[12, 13]*. Alle Systeme können über Logical Domains (LDOMs) virtualisiert werden, die M7-8 und M7-16 zusätzlich in zwei beziehungsweise bis zu vier elektrisch separierte Physical Domains aufgetrennt werden.

Abbildung 3: Systeme mit dem M7-Prozessor: Systeme mit einem bis zu 16 Sockeln, von 2 Rack Units Bauhöhe bis hin zu ganzen Racks.

Der SuperCluster M7 erhält seine besondere Leistungsfähigkeit für Datenbanken durch dieselben Extreme Flash beziehungsweise High Capacity Storage Server wie die Exadata und kann zusätzlich mit SQL in Silicon punkten. Dazu ist der SuperCluster von vornherein darauf ausgelegt, neben der Datenbank auch weitere Lasten auszuführen. Der SuperCluster M7 bietet durch flexiblere und kleinere Konfigurationsoptionen auch einen niedrigeren Einstiegspunkt als der Vorgänger *(siehe Abbildung 3)*.

Literatur

[1] Oracle Database In-Memory, Oracle White Paper, July 2015, *http://www.oracle.com/technetwork/database/in-memory/overview/twp-oracle-database-in-memory-2245633.html*

[2] Juan Loaiza: Accelerate Database Processing with SQL in Silicon, Video - 6min., *http://medianetwork.oracle.com/video/player/4574027564001*

[3] 12.1.0.2 Readme: 2.4 Data Analytics Accelerators on SPARC for Oracle Database Overview, *http://docs.oracle.com/database/121/READM/chapter12102.htm#READM122*

[4] In-Memory Database: SPARC T7-1 Faster Than x86 E5 v3, Performance & Best Practices Blog, Oktober 2015, *https://blogs.oracle.com/BestPerf/entry/20151025_imdb_t7_1*

[5] ESG Lab Review: Redefining Real-time Database Performance with the SPARC M7 Processor from Oracle, *http://www.esg-global.com/lab-reports/esg-lab-review-redefining-real-time-database-performance-with-the-sparc-m7-processor-from-oracle/*

[6] SPECjEnterprise2010: SPARC T7-1 World Record with Single Application Server Using 1 to 4 Chips, Performance & Best Practices Blog, Oktober 2015, *https://blogs.oracle.com/BestPerf/entry/20151025_jent_t7_1*

[7] Oracle Solaris Studio, *http://www.oracle.com/technetwork/server-storage/solarisstudio*

[8] Liang Chen, Raj Prakash, Ikroop Dhillon: Dev Tip: Using Application Data Integrity and Oracle Solaris Studio to Find and Fix Memory Access Errors, April 2015, *https://community.oracle.com/docs/DOC-912448*

[9] Enrico Perla: Hardening allocators with ADI, Blog, November 2014, *https://blogs.oracle.com/enrico/entry/hardening_allocators_with_adi*

[10] Raj Prakash: Oh, no! What Have I Done Now? - Common Types of Memory Access Errors, Blogreihe, Oktober 2015, *https://blogs.oracle.com/raj/entry/common_types_of_memory_access*

[11] Software in Silicon Cloud, *https://swisdev.oracle.com*

[12] Oracle SPARC Servers, *https://www.oracle.com/servers*

[13] Oracle SPARC T7 and SPARC M7 Server Architecture – Software in Silicon Secure Clouds for the Real-Time Enterprise, Oracle White Paper, Oktober 2015, *http://www.oracle.com/technetwork/server-storage/sun-sparc-enterprise/documentation/sparc-t7-m7-server-architecture-2702877.pdf*

[14] IBM GSKit Supports SPARC M7 Hardware Encryption, Solaris and Systems Information for ISVs Blog, Dezember 2015: *https://blogs.oracle.com/partnertech/entry/ibm_gskit_supports_sparc_m7*

Franz Haberhauer
franz.haberhauer@oracle.com

Was DBAs über das neue MySQL 5.7 wissen sollten

Mario Beck und Carsten Thalheimer, ORACLE Deutschland B.V. & Co. KG

Nach zweieinhalbjähriger Entwicklungszeit hat Oracle mit MySQL 5.7 das dritte große MySQL-Release seit der Übernahme von Sun Microsystems veröffentlicht. Es bringt nicht nur zahlreiche neue Funktionalitäten rund um InnoDB, Replikation und den neuen Optimizer, sondern öffnet sich mit dem JSON-Datentyp auch weiteren Anwendungen.

Mit großer Spannung richtet sich alljährlich im Herbst der Blick auf die in San Francisco stattfindende Oracle OpenWorld. Oracle nimmt diese Veranstaltung traditionell zum Anlass, um Produktneuheiten anzukündigen und ausführliche Details zu präsentieren. Auch die MySQL-Gruppe nahm diese Veranstaltung im Oktober 2015 zum Anlass, um fast zwanzig MySQL-Produkt-Announcements bekannt zu geben *(siehe „http://forums.mysql.com/list.php?3")*. Wie erwartet, steht auch das Release 5.7 wieder im Source Code unter dem Filehosting-Dienst GitHub oder direkt unter *„http://www.mysql.com"* zur Verfügung. Auch bei den unterstützten Betriebssystemen (Linux, Windows, Solaris, Apple OS X und FreeBSD) gibt es kaum Unterschiede zur Version 5.6.

Die Neuerungen sind für Kenner des Produkts nicht wirklich neu: Da MySQL ein Open-Source-Projekt ist, werden sie schon