

# Die Datenbank 12c auf Solaris 11.1

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

Schon vor der Übernahme von Sun durch Oracle bestanden enge Kontakte zwischen den Entwicklungsgruppen für das Betriebssystem Solaris bei Sun und für die Datenbank bei Oracle. Solaris unterstützt seit Langem betriebssystemseitig gezielt das Lastprofil der Oracle-Datenbank.

Im Jahr 1993 wurde in Solaris 2.2 Intimate Shared Memory (ISM) eingeführt. ISM kann große, im Hauptspeicher residente und durch eine Vielzahl von Prozessen gemeinsam genutzte Speicherbereiche – genau die Anforderungen für die System Global Area (SGA) der Datenbank – besonders effizient verwalten. Seit der Übernahme arbeiten die Teams noch enger zusammen – etwa hinsichtlich eines frühzeitigen uneingeschränkten Austauschs über neue, bahnbrechende Technologien bis hin zum direkten Zugriff auf den aktuell entwickelten Quellcode. Mit der Freigabe der Datenbank 12c und von Solaris 11.1 werden erste Ergebnisse dieser engen Zusammenarbeit nun sichtbar.

Einige dieser Optimierungen sind für den System- wie auch den Datenbank-Administrator transparent. So wurde durch eine Parallelisierung das Anlegen und Auflösen großer Shared-Memory-Bereiche allgemein um den Faktor sechs bis elf beschleunigt und damit auch das Starten und Herunterfahren von Datenbanken [1]. Während das bereits angesprochene ISM nur statisch dimensionierte SGAs unterstützte, war bei der dynamischen Variante DISM die Performance nicht ganz auf

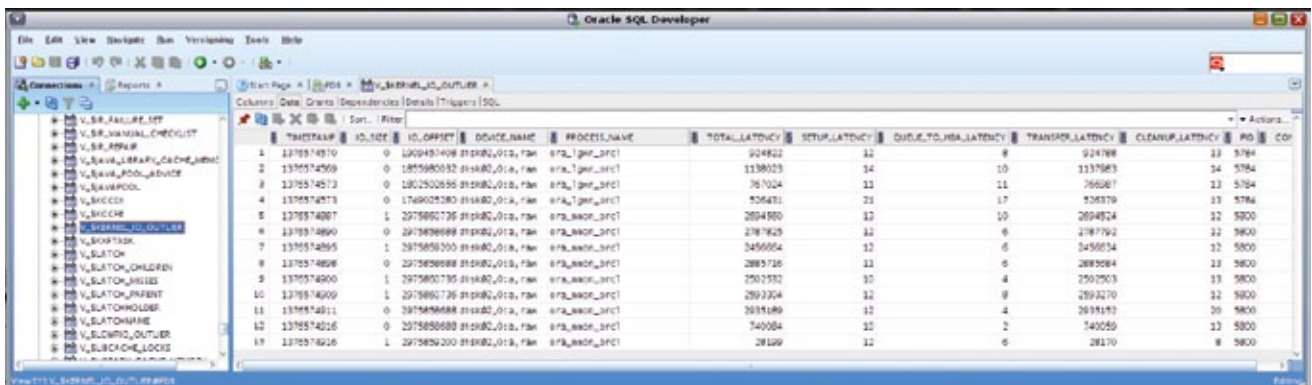
demselben Niveau. Das Automatic Memory Management (AMM) der neuen Datenbank 12c nutzt nun auf Solaris die neue Schnittstelle Optimized Shared Memory (OSM), um die Größe der SGA ohne Zugeständnisse an die Performance online ohne Neustart zu ändern. Damit kann eine Datenbank schnell mit einer kleinen SGA starten und diese nach Bedarf vergrößern.

In Solaris 11.1 wurde zudem eine neue Architektur zur virtuellen Speicherverwaltung eingeführt, die insbesondere die großen Hauptspeicher heutiger Systeme von bis zu mehreren Terabytes (32TB in einer M5-32) wesentlich effizienter verwalten kann und die auf die NUMA-Charakteristika moderner Hauptspeicher-Architekturen abgestimmt ist. Auch hierbei ist eine schnellere Bereitstellung großer Speicherbereiche – etwa für SGAs – und damit ein beschleunigter Systemstart ein unmittelbarer Vorteil.

Durch eine von zehn Millisekunden auf eine Millisekunde verfeinerte Auflösung von Anwendungs-Timern – ermöglicht durch eine effizientere und besser skalierbare Implementierung – wurde die Performance von Anwendungen verbessert, die Polling-Mechanismen nutzen. Davon profitiert auch der Oracle-Log-Writer-Prozess, der bei hoher Last in einen Polling-Modus schaltet [2].

Eine weitere Optimierung in Solaris kommt auf aktuellen SPARC-Servern (T4, T5 und M5) zum Tragen, wo in den Datenbank-Versionen 12c und ab 11.2.0.4.0 (11g R2 Patchset 3) die Log Writer und LMS-Prozesse in Solaris als „Critical Threads“ priorisiert sind. Dabei weist der Solaris-Scheduler diesen Performance-kritischen Threads auf nicht voll ausgelasteten Systemen möglichst ganze Cores exklusiv zu. So müssen sie Ressourcen wie Caches und Ausführungseinheiten nicht mit anderen Threads teilen und können damit die maximal mögliche Hardware-Performance abrufen [3,4].

Neben diesen administrativ transparenten Optimierungen können mit 12c Solaris-Funktionalitäten direkt aus der Datenbank heraus genutzt werden. In Solaris 10 wurde mit DTrace eine sehr mächtige und zugleich leichtgewichtige Performance-Instrumentierung eingeführt. Oracle 12c nutzt auf der Solaris-Plattform DTrace, um nähere Informationen über I/Os auf ASM-verwalteten Geräten zu gewinnen, die sehr lange dauern (mehr als 500 Mil-



ID	INST	IO_OFFSET	DEVICE_NAME	PROCESS_NAME	TOTAL_LATENCY	SETUP_LATENCY	DISK_TO_HBA_LATENCY	TRANSFER_LATENCY	CLEANUP_LATENCY	PO	CO
1	1378574210	0	1209481408 s10x80_01a, raw	ora_lgwr_prc1	924822	22	0	924788	22	5784	22
2	1378574209	0	1805980352 s10x80_01a, raw	ora_lgwr_prc1	1138025	14	10	1137983	14	5784	14
3	1378574213	0	1802502856 s10x80_01a, raw	ora_lgwr_prc1	767024	23	11	766987	23	5784	23
4	1378574213	0	1769025380 s10x80_01a, raw	ora_lgwr_prc1	556431	23	17	556376	23	5784	23
5	1378574207	1	2075880736 s10x80_01a, raw	ora_lgwr_prc1	269490	22	10	269452	22	5800	22
6	1378574209	0	2075880888 s10x80_01a, raw	ora_lgwr_prc1	2787925	22	6	2787792	22	5800	22
7	1378574205	1	2075880200 s10x80_01a, raw	ora_lgwr_prc1	3456054	22	6	3456034	22	5800	22
8	1378574208	0	2075880888 s10x80_01a, raw	ora_lgwr_prc1	2895726	22	6	2895684	22	5800	22
9	1378574209	1	2075880736 s10x80_01a, raw	ora_lgwr_prc1	2502552	22	4	2502508	22	5800	22
10	1378574209	1	2075880736 s10x80_01a, raw	ora_lgwr_prc1	2932204	22	8	2932170	22	5800	22
11	1378574211	0	3075880888 s10x80_01a, raw	ora_lgwr_prc1	3031189	22	4	3031183	22	5800	22
12	1378574216	0	2075880888 s10x80_01a, raw	ora_lgwr_prc1	740084	22	2	740058	22	5800	22
13	1378574216	1	2075880200 s10x80_01a, raw	ora_lgwr_prc1	28129	22	6	28170	22	5800	22

Abbildung 1: V\$KERNEL\_IO\_OUTLIER im Oracle SQL Developer

lisekunden). Diese werden in der neuen dynamischen View „V\$KERNEL\_IO\_OUTLIER“ bereitgestellt, die auf Solaris wesentlich detailliertere Informationen bietet als die Views „V\$IO\_OUTLIER“ und „V\$LGWRIO\_OUTLIER“, die auf allen Plattformen vorhanden sind.

„V\$KERNEL\_IO\_OUTLIER“ ist dort als View zwar ebenfalls verfügbar, bleibt aber leer. „V\$KERNEL\_IO\_OUTLIER“ liefert zusätzlich eine Aufspaltung der Zeit-Anteile, die auf den Datenpfad sowie auf das Gerät selbst entfallen, was im Fehlerfall eine Diagnose signifikant vereinfacht und beschleunigt [5]. **Abbildung 1** zeigt ein Beispiel mit einigen Events, dargestellt mit dem SQL Developer. Es ist deutlich erkennbar, welche Prozesse wann I/O-Probleme hatten und welche Platte daran beteiligt war.

Diese gezielten Optimierungen in Solaris für die Datenbank sowie die Nutzung spezifischer Features in Solaris durch die Datenbank stehen noch am Anfang. Bereits bei der Ankündigung von Solaris 11.1 wurde kurz auf ein weiteres Projekt hingewiesen: Die Verlagerung einiger Funktionen aus dem RAC-Lock-Management in ein Solaris-Kernel-Modul („Kernel Mode RAC Acceleration“), wovon eine Durchsatz-Steigerung um 20 Prozent erwartet wird [6].

Solaris dient einerseits als Betriebssystem auf General-Purpose-Servern direkt als Plattform für die Datenbank – 12c ist für Solaris auf SPARC- sowie auf x86-Servern (und damit für Virtualbox) verfügbar –, andererseits dient Solaris auch als Betriebssystem für die ZFS Storage Appliance (ZFS SA, vormals S7000), auf die als Network Attached Storage (NAS) von der Datenbank aus über dNFS zugegriffen wird.

Auf der ZFS SA war als einzigem NAS-System bereits mit der Datenbank 11g Hybrid Columnar Compression (HCC) nutzbar. Mit der Datenbank 12c wird jetzt das Oracle Intelligent Storage Protocol (OISP) eingeführt, das der Datenbank ein automatisches Performance-Tuning von Datasets auf der ZFS SA ermöglicht. Solaris ZFS verwendet Copy-on-Write-Technologien [7], die bereits mit der Datenbank 11g verwendet werden konnten, um Datenbanken auf dNFS etwa für Testzwecke schnell zu

klonen [8], bei 12c ist das schnelle und platzsparende Klonen von Pluggable Databases von besonderem Interesse.

## Fazit

Dieser Artikel zeigt einige der wechselseitigen Optimierungen der Datenbank 12c und des Betriebssystems Solaris. In der SPARC-Prozessor-Roadmap werden bereits einige Funktionen für „Software in Silicon“ skizziert, dem nächsten Schritt des „Engineered to Work Together“.

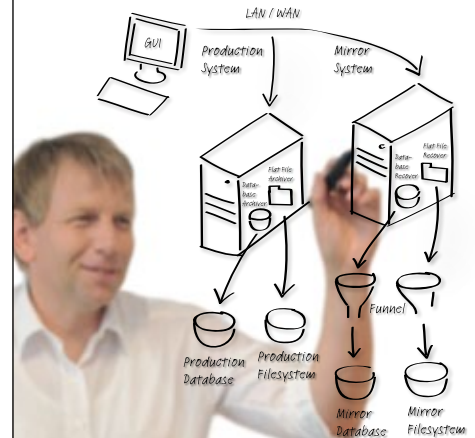
## Literatur

- [1] Faster Memory Allocation Using vmtasks: [https://blogs.oracle.com/sistare/entry/faster\\_memory\\_allocation\\_using\\_vmtasks](https://blogs.oracle.com/sistare/entry/faster_memory_allocation_using_vmtasks)
- [2] High Resolution Timeouts: [https://blogs.oracle.com/sistare/entry/high\\_resolution\\_timeouts](https://blogs.oracle.com/sistare/entry/high_resolution_timeouts)
- [3] Critical Threads Optimization: [https://blogs.oracle.com/observatory/entry/critical\\_threads\\_optimization](https://blogs.oracle.com/observatory/entry/critical_threads_optimization)
- [4] Reducing High Waits on 'log file sync' on Oracle Solaris SPARC by Increasing Priority of Log Writer (My Oracle Support, Note-ID 1523164.1)
- [5] Oracle Database 12c and Oracle Solaris 11 Dtrace: [https://blogs.oracle.com/solaris/entry/database\\_12c\\_and\\_solaris\\_dtrace](https://blogs.oracle.com/solaris/entry/database_12c_and_solaris_dtrace)
- [6] Oracle Solaris 11.1 – What's New: <http://www.oracle.com/technetwork/server-storage/solaris11-1-what-new-1732377.pdf>
- [7] Franz Haberhauer: Best Practices für Datenbanken auf ZFS, DOAG News Ausgabe 3/2013
- [8] Clone Your dNFS Production Database for Testing (My Oracle Support, Note-ID 1210656.1)

Franz Haberhauer  
[franz.haberhauer@oracle.com](mailto:franz.haberhauer@oracle.com)



## Libelle BusinessShadow®



Unabhängig bezüglich

- Fehlerursache
- Entfernung
- Hardware / Architektur
- Komplexer Systeme

Schnelle Arbeitsaufnahme

- Mit konsistenten Daten
- Auf Knopfdruck
- Automatisiert
- ...

Hans-Joachim Krüger  
Chief Technology Officer  
Libelle AG

Erfahren Sie mehr:  
[www.Libelle.com/business](http://www.Libelle.com/business)

Besuchen Sie uns auf dem  
DSAG-Jahreskongress

17. – 19. September 2013  
NCC Nürnberg Convention Center  
Halle 12, Stand-Nr. B 32

ORACLE Gold Partner



Libelle

Libelle AG

Gewerbestr. 42 • 70565 Stuttgart, Germany  
T +49 711 / 78335-0 • F +49 711 / 78335-148  
[www.Libelle.com](http://www.Libelle.com) • [sales@libelle.com](mailto:sales@libelle.com)