Wenn sich ein Solaris-System-Administrator an die Worte „don't change a running system" erinnert, ist es meistens schon zu spät. Eine einfache Umkehr ist nicht immer möglich. Der Autor hat routinemäßig ein Solaris „CPU OS Patchset" eingespielt und ist danach in Teufels Küche geraten.

Was sie von Oracle über ZFS nicht hören werden

Roman Gächter, Trivadis AG

Dieser Artikel beschreibt die Erfahrungen, die der Autor bei einem Kunden mit dem „Zetabyte File System" (ZFS) gemacht hat. ZFS wurde von Sun Microsystems für Solaris entwickelt und gehört heute Oracle. Der Kunde muss anonym bleiben und kann nicht erwähnt werden.

ZFS ist unbestritten ein außerordentlich gutes Produkt. Eine der innovativsten Erfindungen in diesem Bereich der letzten Jahre. Auch wenn das Aufsetzen von ZFS aus der Sicht des Administrators unglaublich leicht vonstattengeht, darf nicht vergessen werden, dass in einem Enterprise-Datenbank-Umfeld zusätzliches Tuning notwendig ist und man von Anfang an ein gutes Konzept für das Storage-Layout ausarbeiten muss.

Eine Banken-Plattform, die seit knapp einem Jahr in Betrieb war und problemlos lief, bekundete drei Wochen nach dem Einspielen des Solaris CPU OS Patchset vom Januar 2012 massive Performance-Probleme. Diese führten zu Verbindungsabbrüchen zum SWIFT-Netzwerk, dem Worst Case für das Banken-Business. SWIFT steht für „Society for Worldwide Interbank Financial Telecommunication" und ist eine Plattform, über die Banken untereinander Finanztransaktionen abwickeln.

Die einzige Änderung, die zuvor am System vorgenommen wurde, war das Solaris-Upgrade. Für das Business, den Software-Lieferanten und das Management war somit die Ursache des Problems gefunden. Nun wurde der Ball den Solaris-System-Administratoren zugespielt, die gewaltig unter Druck gerieten, das Performance-Problem zu lösen.

Die System-Architektur

Verteilt auf zwei Rechenzentren, ist die Plattform redundant aufgebaut. Es handelt sich um mehrere SPARC Enterprise M4000 Server, auf denen Solaris 10 installiert ist. Es wird die Solaris-Virtualisierungs-Lösung mit Solaris Zonen genutzt. Die verschiedenen Systeme der Applikationen laufen alle in Solaris Containern.

Die Daten der Banken-Applikationen befinden sich in einer Oracle-Datenbank 11g. Die gesamte Oracle-Datenbank-Installation ist in einem Zpool konzentriert. Dieser wird mit der Replikations-Lösung SNDR von Solaris synchron in das andere Rechenzentrum repliziert. Es ist bekannt, dass sich für solche Zwecke Oracle Data Guard besser eignen würde, die Lösung mit SNDR wurde jedoch vom Software-Hersteller empfohlen und vom Kunden gewünscht.

Die Daten liegen auf einem SAN. Es wird ausschließlich das Zetabyte File System (ZFS) verwendet. Die zu replizierende Datenmenge ist relativ klein (50 GB). Die Verteilung zwischen Schreib- und Lese-Operationen auf der Datenbank ist ausgeglichen. Die Antwortzeiten der Applikationen für die Benutzer verhalten sich in etwa linear zu den Antwortzeiten der Datenbank. Die Systeme sind in Bezug auf CPU nur wenig ausgelastet. Aufgrund der synchronen Replikation war früher die Netzwerk-Verbindung der Flaschenhals zwischen den Rechenzentren.

Eingrenzung des Problems

Es kristallisierte sich schnell heraus, dass der Engpass auf der I/O-Seite der Oracle-Datenbank lag. Die aufgezeichneten „System Activity Report"-Daten (5-Minuten-Mittelwerte) zeigten zum Teil 100 Prozent „busy"-Werte auf den Device-Files des Oracle Zpool und entsprechend auf dem Replikations-Device von SNDR. Obwohl der gesamte Daten-Durchsatz bescheiden war, zeigten die Oracle AWR-Reports sehr schlechte Antwortzeiten („AVERAGE WAIT") – zum Teil über 50 Millisekunden. Auch mit „iostat" (Mess-Intervall 1 Sekunde) waren viele hohe durchschnittliche Antwortzeiten zu beobachteten, also Werte von „asvc_t" („average service time of active transactions in milliseconds") von über 50. Weil für die Plattform eine dedizierte Netzwerk-Verbindung zwischen den Rechenzentren verfügbar war und die gemessenen Bandbreiten nur einen Teil der möglichen Kapazität ausmachten, war ein Problem mit der synchronen Replizierung unwahrscheinlich. Man konnte die weitere Analyse also auf Oracle und das ZFS fokussieren.

Übersicht ZFS

Auszug aus Wikipedia: „ZFS ist ein von Sun Microsystems entwickeltes transaktionales Dateisystem, welches zahlreiche Erweiterungen für die Verwendung im Server- und Rechenzentrums-Bereich enthält. Hierzu zählen die enorme maximale Dateisystemgröße, eine einfache Verwaltung selbst komplexer Konfigurationen, die integrierten RAID-Funktionalitäten, das Volume-Management sowie der prüfsummenbasierte Schutz vor Datenübertragungsfehlern."

Die Vorteile von ZFS sind unbestritten und der Autor möchte sie nicht

mehr missen. Insbesondere die einfache Administration, die optimale Integration mit Solaris und die eingebaute Funktionalität von Snapshots sind von großem Nutzen.

Man sollte sich der diversen ZFS-Filesystem-Chaches bewusst sein. Diese können durch geschickte Konfiguration und Nutzung von schnellen Devices wie „solide state disks“ die Performance erheblich verbessern. Sie können sich aber auch kontraproduktiv auswirken:

- Behinderung anderer Applikationen durch Speicher-Verbrauch des ARC-Cache
- Verdopplung der Schreib-Operationen für „synchrone writes“ im ZIL Log
- Abbremsen von Datenbanken durch „file level prefetching“- und „device level read ahead“-Mechanismen beim Lesen

Nachfolgend eine Liste von ZFS-Caches:

- Der „first level cache” (ARC cache) befindet sich im Memory. Es handelt sich um eine Variante des ARC-Algorithmus (Adaptive Replacement Cache)
- Optional können „second level disk caches” definiert werden:
  - Dafür eignen sich schnelle Disks wie SSD
  - Der „read cache“, als „L2ARC“ bezeichnet, wird über das Zpool-Property „cachefile“ aktiviert und über das ZFS-Property „secondarycache (all | none | metadata)“ beeinflusst
  - Der „write cache“ wird als „ZFS Intent Log“ (ZIL) bezeichnet und befriedigt POSIX-Requirements für synchrone Transaktionen. Ist kein separates ZIL-Device definiert, wird der ZIL ein Teil des Zpool. „zpool status” zeigt die „log devices” an.
  - Die „second level“-Chaches lassen sich während des Betriebs einfach hinzufügen, konfigurieren oder entfernen

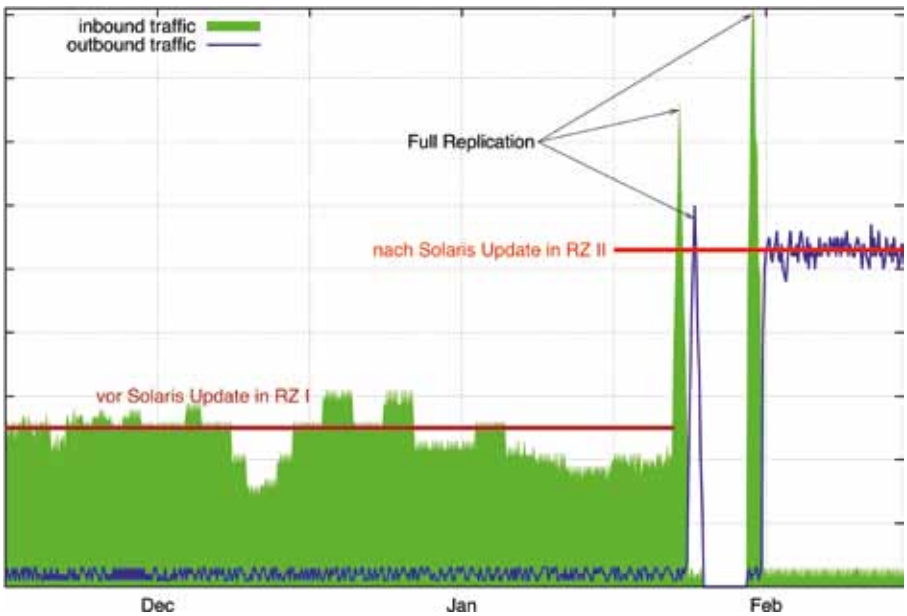

Abbildung 1: Replikations-Durchsatz auf der Leitung zwischen den Rechenzentren

ZFS und Oracle

In den Anfangszeiten von ZFS gab es Statements von Sun Microsystems, die darauf hinausliefen, ZFS nicht für Oracle-Datenbanken zu verwenden. Dies hat sich schon seit einiger Zeit geändert. Heute ist ZFS offiziell von Oracle für Datenbanken unterstützt und auch empfohlen. Wichtig ist jedoch, das ZFS nach „best practice“ aufzusetzen. Oracle hat diverse Whitepapers dazu verfasst – und diese sollten auch berücksichtigt werden. Die wichtigsten Punkte sind:

- ZFS „record size“ an „db block size“ anpassen
- Überwachen des „ARC cache“ im Memory und, wenn nötig, begrenzen
- ZFS intend log (ZIL) für „Oracle data files“ umgehen
- Den ZFS-Füllgrad (Usage) überwachen, immer mindestens unter 80 Prozent oder besser noch darunter bleiben
- Wenn möglich, bei Oracle-Datenbanken immer dedizierte Zpools für „redo logfiles“, „data files“ und „archivelog files“ mit dedizierten SAN LUNs verwenden
- Für Zpools nur ganze LUNs verwenden, keine Partitionen

Analyse von ZFS-Performance-Problemen

Wie bereits erwähnt, startet man hier am besten damit, die Oracle-„Best Practice“-Whitepaper zu studieren und zu untersuchen, ob die eigene Installation davon abweicht. Das beste Solaris-Bordmittel für die Analyse der I/O-Performance ist „iostat“. Man sollte ein Skript aufsetzen, das die „extended“-iostat-Werte im 1-Sekunden-Intervall rund um die Uhr aufzeichnet. Zudem sollte man sich protokollieren lassen, wie viel Memory sich der ZFS-ARC-Cache reserviert und ob das Memory auch schnell wieder freigegeben wird – sofern anderweitig gebraucht. Mit einer Beschränkung des ARC-Cache bewegt man sich auf der sicheren Seite. Der Befehl, um den ARC-Cache anzeigen zu lassen, lautet „root# echo „::memstat“ | mdb –k“.

Wichtige Informationen gewinnt man durch Analysieren. Das im Solaris-Betriebssystem eingebaute Dynamic Tracing (Dtrace) ist ein sehr mächtiges Tool. Es bietet die Möglichkeit, in laufenden Prozessen den Arbeitsspeicher, die Prozessorzeit, das Dateisystem und die Netzwerk-Ressourcen zu untersuchen. Im Zusammenhang mit der ZFS-Performance interessieren folgende Dtrace-Ergebnisse:

- Welche Programme verursachen “top writes“ und ”top reads“?
- Wie sieht die “byte size“-Verteilung aus?
- Sind „ganging“-Operationen zu beobachten? Dies wäre ein Zeichen von fragmentierten Zpools.

In der Oracle-Datenbank kommt man nicht darum herum, AWR Reports zu generieren, um aussagekräftige Bewertungen zur I/O-Performance machen zu können. Bei ZFS spricht man von „ganging“, wenn die Daten nicht mehr an einen zusammenhängenden Platz im ZFS geschrieben werden können und kleinere Gaps verwendet werden müssen. Die Anzahl der „ganging“-Operationen beim Schreiben ist linear zum Fragmentierungsgrad eines Zpool. Besteht der Verdacht auf eine Fragmentierung in einem Zpool, sollte man das entsprechende Tracing durchführen. Die Dtrace-Syntax “dtrace -qn ‚fbt::zio_gang_tree_issue:entry { @[pid]=count(); }‘ -c “sleep 300““ zeigt die „ganging“-Operationen in einem System an.

Problemlösung

Zurück zum Performance-Problem: Es wurde festgestellt, dass sich die über die Netzwerk-Verbindung zum anderen Rechenzentrum replizierte Datenmenge nach dem Solaris-Update sprunghaft fast um den Faktor zwei erhöht hatte. Abbildung 1 zeigt den Replikations-Durchsatz auf der Leitung zwischen den Rechenzentren. Grüne Kurve „inbound“, blaue Kurve „outbound“. Vor und nach dem Upgrade wurden Failover-Tests durchgeführt und jeweils eine volle Replikation (Spitzen) gefahren. Nach dem Upgrade liefen die Applikationen auf der anderen Seite also unter der blauen Kurve; die Richtung der Replikation hatte sich geändert.

Die Kernel Patches 147440-10 und 144500-19 des Januar-Patch-Bundle führten im ZFS neue Properties ein, unter anderem das Property „logbias“. Für Datenbank-Files sollte dieses auf „throughput“ gesetzt sein, es war jedoch nach der Patch-Installation auf dem Default-Wert „latency“. „throughput“ bedeutet: Der ZFS Intend Log wird für synchrones Schreiben nicht verwendet, womit sich die „writes“ quasi um die Hälfte reduzieren. Anmerkung: In Oracle Solaris10/08- bis 10/09-Installationen wurde ein ähnliches Verhalten durch das Setzen des Kernel-Parameters im „/etc/system“ durch „set zfs:zfs_immediate_write_sz=8000“ erreicht. Dieser Kernel-Parameter war hier jedoch nicht gesetzt. Nach der Änderung von „logbias“ auf „througput“ im ZFS mit den Oracle-Daten-Files hat sich tatsächlich die replizierte Bandbreite wieder auf den normalen Wert eingestellt. Leider war das Performance-Problem damit aber noch nicht gelöst.

Es wurde festgestellt, dass das ZFS mit den „redolog files“ fälschlicherweise auf eine „recordsize“ von 8 K anstatt 128 K eingestellt war. Zudem waren die Messungen der „bitesize“-Verteilung im Oracle-Zpool anders als erwartet. Die größte Verteilung sollte bei 8 K, entsprechend zur „recordsize“ des ZFS mit den Oracle „datebase files“ sein. Tabelle 1 zeigt jedoch ein ganz anderes Bild, wie mit dem Dtrace-Toolkit-Programm „bitesize.d“ gemessen wurde.

6. May		
371 zpool-Oracle_R\0		
value	**----- Distribution -----**	**count**
256		0
512	@@@	4498
1024	@@@@@	7236
2048	@@@@@@@@@@	14340
4096	@@@@@@	8388
8192	@@@@@	7461
16384	@@@@@	7710
32768	@@@@@	7068
65536		425
131072		40
262144		0

Tabelle 1

11. May		
371 zpool-Oracle_R\0		
value	**----- Distribution -----**	**count**
256		0
512	@	4498
1024	@	7236
2048	@@	14340
4096	@@@@@	8388
8192	@@@@@@@@@@@@@@@	7461
16384	@@@@@@	7710
32768	@@@@@@	7068
65536	@@	425
131072	@@	40
262144		0

Tabelle 2

Daraufhin wurden der „recordsize“-Wert im „redolog file“ ZFS auf 128 K geändert sowie anschließend in einem Servicefenster der ganzen Oracle-Zpool gesichert und wiederhergestellt, um diese Änderung wirksam zu machen. Danach kam das große Aufatmen. Die Performance bewegte sich wieder in einem Bereich wie vor dem Upgrade. Die WAR-Werte für „AVERAGE WAIT“ lagen bei 14 Millisekunden. Nun sahen auch die Resultate der Bitesize-Verteilung besser aus (siehe Tabelle 2).

Leider gab es bald eine bittere Enttäuschung – die Lösung war immer noch nicht gefunden, denn drei Wochen später war das Performance-Problem zurückgekehrt. Es blieb nur der bekannte Workaround: Servicefenster beantragen, Datensicherung des Oracle Zpool und Wiederherstellen der Daten. Das brachte wieder für drei Wochen Ruhe.

Abbildung 2 zeigt, wie sich die Performance-Werte innerhalb von zwei Wochen verschlechtert haben. Die Grafik zeigt die Werte für „AVERAGE WAIT (ms) for LOG FILE SYNC (time to wait before writing into RedoLog files)“.

Ein Oracle-Consultant erklärte vor Ort, dass das Performance-Problem durch die fortschreitende ZFS-Fragmentierung des Oracle-Zpool verursacht wird. Verstärkt wird das Problem, weil das Storage-Layout nicht den Best-Practice-Richtlinien folgt und keine dedizierten Zpools für eine Separierung von „redolog files“ und „database files“ verwendet werden. Bei dieser Umgebung wird wegen der SNDR-Replikation absichtlich nur ein Oracle-Zpool verwendet, da die Dauer einer vollständigen SNDR-Replikation abhängig von der Anzahl und der Größe der zu replizierenden Zpools ist.

ZFS überschreibt nie Daten, sondern folgt dem Copy on Write-Prinzip. Dies ist optimal für die Daten-Integrität und auch notwendig, damit Techniken wie „snapshots“, „cloning“, „shadow copy“ und „zfs send/receive“ überhaupt funktionieren. Leider handelt sich ZFS damit das Fragmentierungsproblem ein, sobald in einem Zpool regelmäßig geschrieben wird.

Die Empfehlung von Oracle war: „Storage-Layout ändern und Trennen beziehungsweise Verteilen von Oracle

„redolog files“ und „database files“ auf mehrere Zpools mit dedizierten LUNs.“ Diese Umorganisation wurde in die mittelfristige Planung aufgenommen.

Durch Benchmarks auf Test-Systemen und „ganging“-Messungen mit Dtrace wurde festgestellt, dass sich die Fragmentierungs-Problematik in der Umgebung durch die Erhöhung des ZFS-„freespaces“ entschärfen ließ. Die Oracle-Administratoren haben daraufhin zunächst die Daten im Oracle-Zpool so weit wie möglich reduziert. Dadurch verlängerte sich die Zeitspanne auf zwei Monate, bis eine neue Defragmentierungsaktion mit Service-Fenster notwendig wurde.

Mit einer Usage von 82 Prozent im Oracle-Zpool dauerte es drei Wochen, bis die Fragmentierung die Performance beeinträchtigte, mit 67 Prozent Usage acht Wochen. Abbildung 3 zeigt sehr schön die Auswirkung der schleichenden Fragmentierung. Erster Knick am 6. Juni durch Defragmentierung, zweiter Knick am 13. Juni durch Defragmentierung und mehr freien Platz im Zpool. Abbildung 4 zeigt die „ganging“-Operationen vor (rote Kurve) und nach einer Defragmentierung des Oracle-Zpool.

Nun fiel der Entschluss, den Oracle-Zpool in dem Maße zu vergrößern, dass die Zeit für eine volle Replikation gerade noch akzeptabel war, und den Freespace auf 50 Prozent heraufzusetzen. Mit dieser Aktion war das Fragmentierungsproblem gelöst. Einerseits sind keine „ganging“-Operationen mehr zu beobachten, andrerseits hat sich auch die Performance massiv verbessert. Die AWR-Werte „AVERAGE WAIT“ lagen bei 6 Millisekunden. Abbildung 5 zeigt die Entwicklung der „average wait times“. Links bis zum 13.6. mit einer ZFS-Usage von 80 Prozent, in der Mitte nach dem Löschen von Daten, „file relocation“ und ZFS-Usage von 67 Prozent und ganz rechts nach einer weiteren „file relocation“ und mit einer Usage von nur noch 50 Prozent.

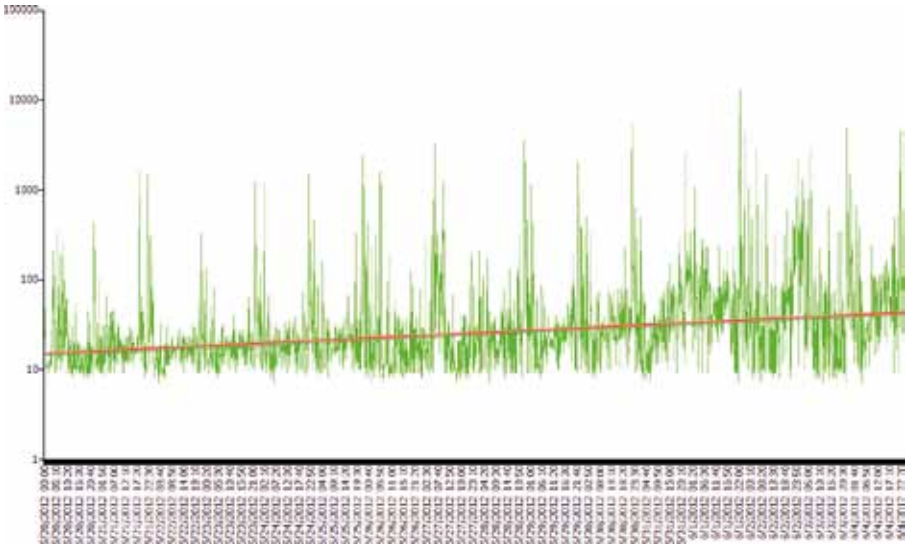

Abbildung 2: Verschlechterung der Performance-Werte innerhalb von zwei Wochen

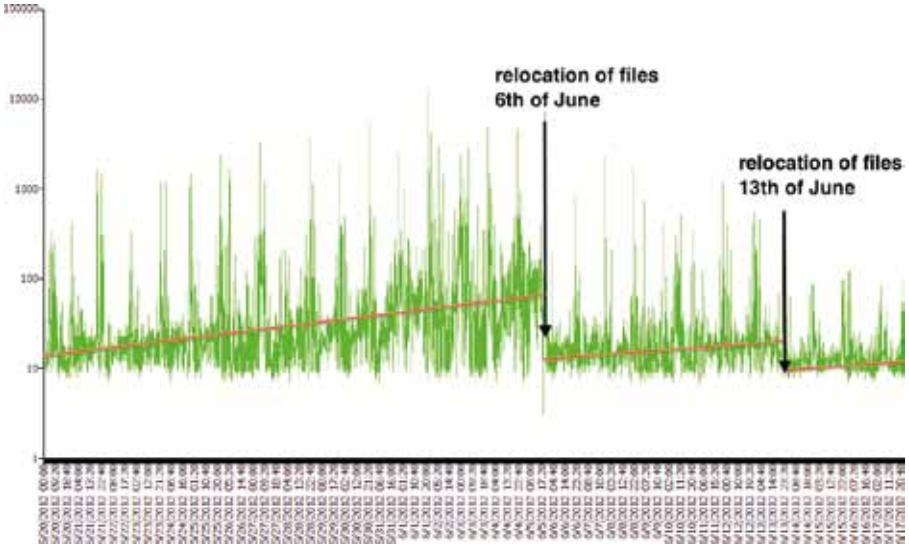

Abbildung 3: Die Auswirkung der schleichenden Fragmentierung

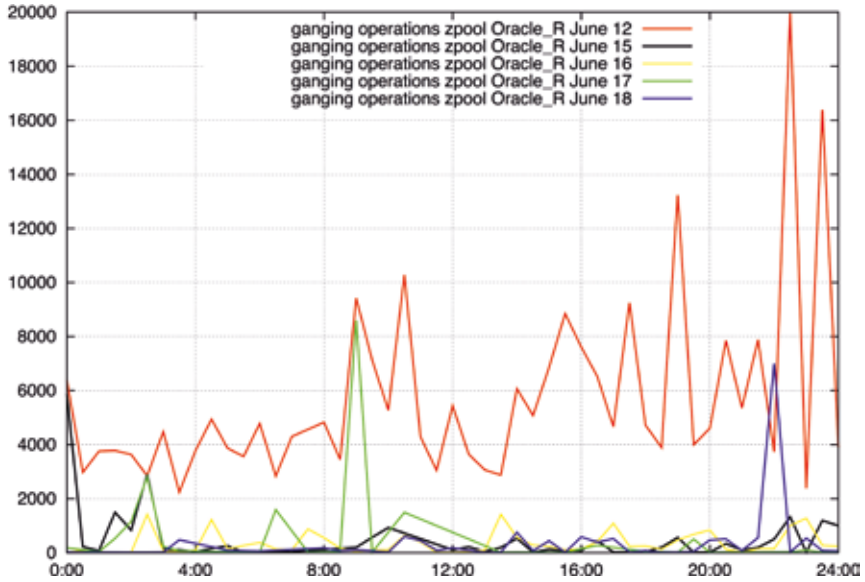

Abbildung 4: „ganging“-Operationen vor (rote Kurve) und nach einer Defragmentierung des Oracle-Zpool

Fazit

Die Gretchenfrage, die sich stellt: Standen die Performance-Probleme im Zusammenhang mit den applizierten Solaris-Patches oder entstanden sie durch eine langsame Reduktion des freien Plat-

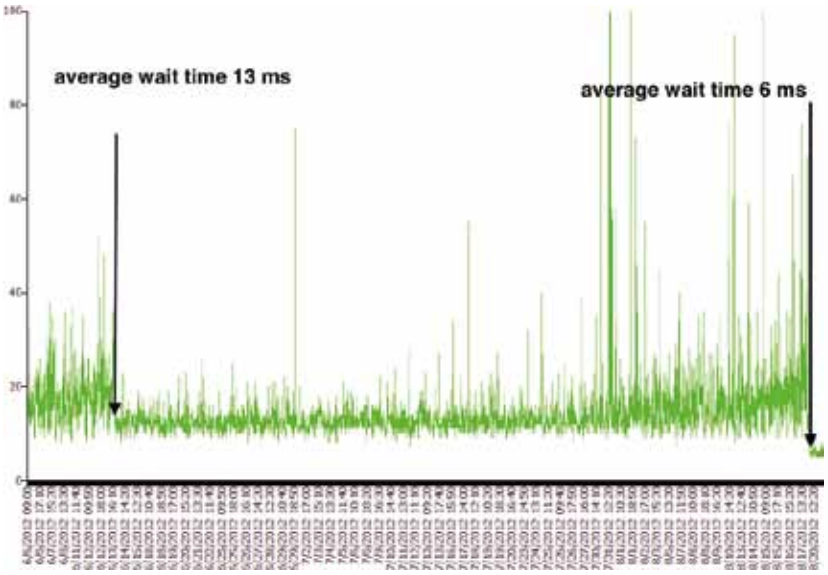

Abbildung 5: Die Entwicklung der „average wait times“

zes im Zpool? Gemäß den Aufzeichnungen hat sich der Füllgrad des Zpool nur minimal verändert. Die Vermutung liegt nahe, dass mit den Patches 147440-10 und 144500-19 Änderungen im ZFS vorgenommen wurden, die die Fragmentierungsproblematik in unserer spezifischen Umgebung akzentuiert haben.

Im Nachhinein betrachtet ist die Lösung des ZFS-Performance-Problems einfach: Reduktion des Füllgrades von ZFS von 80 auf unter 50 Prozent. Damit hat man die vorher immer wiederkehrende Fragmentierung des Oracle-Zpool, verbunden mit Performance-Problemen in der spezifischen Umgebung, gänzlich eliminiert und macht damit regelmäßige Defragmentierungs-Aktionen mit Service-Fenstern unnötig. Leider musste die Lösung selbst gefunden werden. Die Durchführung ist in dieser Umgebung mit relativ kleinen Datengrößen möglich, wäre hingegen bei größeren Datenmengen nicht praktikabel.

Es gab mehrere Service Requests beim Oracle Support – kein Wort über ZFS-Fragmentierung. Erst ein Oracle-Consultant vor Ort nahm das Wort in den Mund. Zudem muss man sehr lange suchen, um die Fragmentierungs-Problematik in der Oracle-Dokumentation zu finden – in den ZFS-Manuals herrscht hier großes Schweigen. Schön wäre es, wenn Oracle ein Tool bereitstellen könnte, das ähnlich wie ein ZFS-„Scrubbing“ im Hintergrund im Online-Betrieb laufen und den Zpool automatisch defragmentieren würde.

Literatur und Links

- Oracle ZFS Whitepaper: http://www.oracle.com/technetwork/server-storage/solaris/config-solaris-zfs-wp-167894.pdf
- Dtrace: http://www.brendangregg.com/dtrace.html
- ZFS Fragmentation: http://wildness.espix.org/index.php?post/2011/06/09/ZFS-Fragmentation-issue-examining-the-ZIL
- AWR: http://www.oracle-base.com/articles/10g/automatic-workload-repository-10g.php

Roman Gächter
Roman.Gaechter
@trivadis.com

Vertrauen in Performance, weniger in Oracle

Insbesondere in den letzten 12 Monaten hat das Interesse der IT-Anwender und Datacenter-Betreiber an sogenannten Appliances – integrierte Server/Storage/Network-Systeme – stark zugenommen. Oracle gehört zu den Trendsettern in diesem Bereich – deren Engineered Systems bauen auf hochintegrierten Oracle-Technologien auf, die deshalb auch gut optimiert sind. Darüber hinaus verfügen sie über einen One-Vendor-Support.

Um Interesse und Erfahrungen mit diesen Systemen zu eruieren, führte die DOAG gemeinsam mit der Experton Group eine Online-Befragung bei den DOAG-Mitgliedern durch. Insgesamt beteiligten sich über 500 Unternehmen an der Befragung, darunter 290 Anwender und 137 Partner, zusätzlich auch Consultants und Wettbewerber. Der Fragebogen umfasste insgesamt 15 Fragen, die nach den einzelnen Zielgruppen (Anwender, Anbieter, Partner, Berater) ausgewertet wurden. Ausgewählte Ergebnisse wurden auf dem DOAG 2013 IMC Summit am 6. Juni in Mainz von der Experton Group präsentiert und diskutiert.

Sehr wichtig ist den Anwendern insbesondere die Senkung der IT-Betriebskosten. Ungefähr die Hälfte der Unternehmen ist der Meinung, dass Appliances hierfür einen Beitrag liefern können. Als Vorteile sehen die Anwender insbesondere die einfache Implementierung und bessere Antwortzeiten der IT-Systeme – wogegen das Argument „Einstieg in die Private Cloud“ kaum Beachtung findet. Als Nachteil wird die Abhängigkeit vom Hersteller gesehen – sowohl Technologie wie Support betreffend, wogegen ein sehr hohes Vertrauen in die Leistungsfähigkeit/Performance der Systeme besteht.

Weitere Ergebnisse der Befragungs-Analyse sind unter http://engsys.doag.org zu finden.

Andreas Zilch
Experton Group AG
info@experton-group.com