Klonen von Umgebungen mit Exadata und ZS-Appliances

Jan Schreiber
Loopback.ORG GmbH – Database Intelligence
Hamburg / Frankfurt / München

Schlüsselworte
Oracle Exadata, BI/DWH, ZS Appliance, ZFS Cloning, dNFS, RDMA, IB

Einleitung
Ein großer Telekommunikationsdienstleister betreibt seine BI/DWH-Landschaft auf mehreren Exadata Systemen. Für die Datensicherung wurde eine doppelköpfige ZFS Storage Appliance (ZA) angeschafft, die per Infiniband direkt an die Exadata-Systeme angeschlossen wird.

Dieser Projektbericht beschreibt die Erfahrungen, verwendete Best Practices und Stolperstricke beim Aufbau einer ZS-Appliance-basierten Sicherungs-Infrastruktur, sowie bei der Erstellung von Test- und Entwicklungs-Umgebungen mit RMAN und ZFS Snapshots.

Produkteigenschaften
Laut dem Oracle Marketing wurden ZS Appliances gerade im Exadata-Umfeld in großen Stückzahlen abgesetzt. Durch die Möglichkeit, diese Storage-Systeme über Infiniband direkt an die Exadata-Plattform anzuschließen, verspricht der Einsatz sehr gute Durchsatzraten gerade bei der Datensicherung. Oracle selbst verwendet ZS Appliances als Storage für die eigene Infrastruktur[1]. Auch die Möglichkeit, Snapshots und Klone anzufertigen, um Datenbanken zu klonen, wird explizit beworben. Diese Features sind zwar auch in EMC's VNX oder NetApp's WAFL zu finden, aber die Exadata-IB-Integration out of the box ist ein Alleinstellungsmerkmal.

Ein wesentlicher Grund für die Auswahl dieser Lösung ist auch dir Verfügbarkeit der Hybrid Columnar Compression (HCC), die auf der ZA unterstützt ist. HCC-komprimierte Daten aud der Produktion können in den Dev/Test Datenbanken verwendet werden, ohne ausgepackt werden zu müssen.

Mit Hilfe der Fähigkeit der ZS Appliance, Snapshots und Klone zu erstellen, sollte in unserem Projekt das Exadata-Backup auf der ZS-Appliance als Grundlage für die Erstellung einer Reihe von Test- und Standby-Datenbanken dienen. Hierfür wurden Sun X4-2 Server angeschafft, die ebenfalls in den IB-Verbund aufgenommen wurden.

Mit diesem Setup sollen die Bereitstellungszeiten für neue Test- und Entwicklungsumgebungen dramatisch gesenkt werden. Ebenso sollen die Wiederherstellungszeiten aus der Datensicherung

[1] „*As part of Oracle IT's ongoing initiative to transition from legacy storage systems (primarily from NetApp and EMC) and standardize on the Sun ZFS Storage Appliance, significant performance and efficiency benefits have been realized*." (Sun ZFS Storage Appliance and Oracle IT: Use Cases and Benefits", 2012

verringert werden. Durch die Ablösung des bisherigen FC-Storage-Systems sollen außerdem Kosten für Wartung und Lizenzen gespart werden.

ZA-Storage im Infiniband-Verbund mit der Exadata
Oracle erlaubt keine beliebigen Eingriffe in die IB-Verkabelung an den Exadatas. Für das eigene Storage-Produkt gibt es allerdings eine Freigabe, diese mit einer oder mehreren Exadata Maschinen zu „verschmelzen“:

„Two Exadata racks can be cabled together to share the same IB fabric mesh. The merged Exadata racks can then be connected to a single clustered Oracle ZFS Storage Appliance. Successful setup requires adherence to some critical prerequisites and instructions, including physical cabling procedures. Be sure to reference the Oracle Exadata Database Machine Owner's Guide, Part IV: Extension Configuring a Single Oracle ZFS Storage Appliance into an InfiniBand Fabric with Multiple Oracle Exadata Machines of the Oracle Database Machine and Oracle Exadata Storage Expansion Rack“[2]

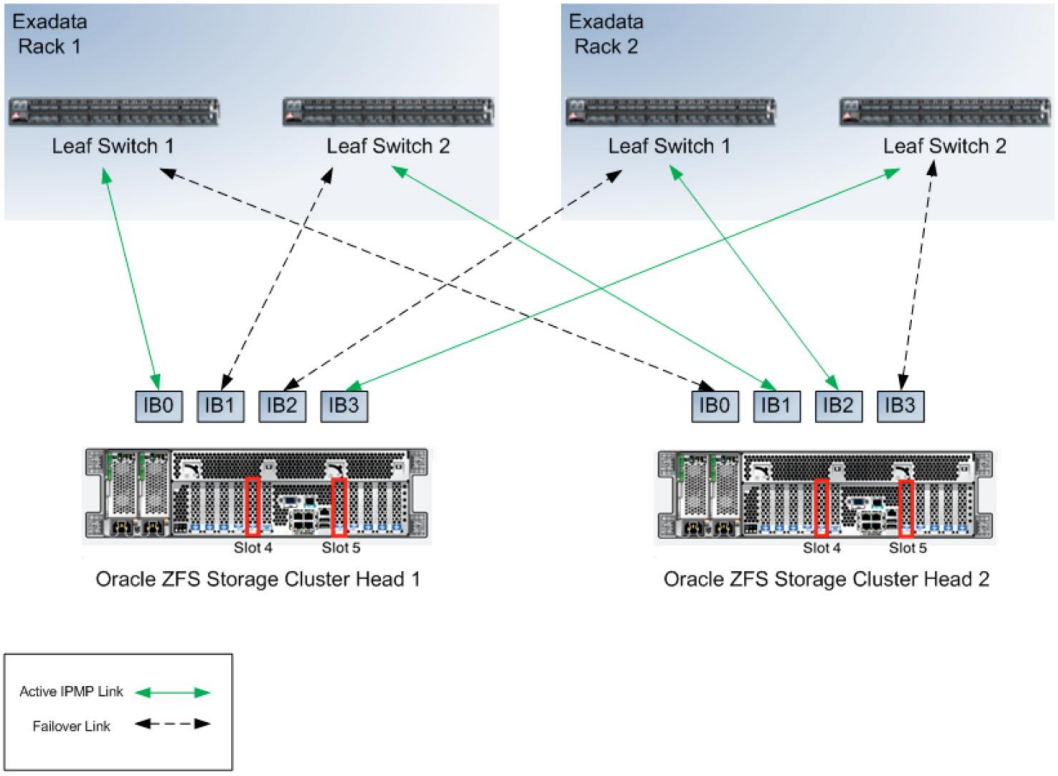

Abb. 1: Infiniband-Verkabelung

Die Infiniband-Fabric kann auch auf Client-Systeme weitergeführt werden. In unserem Fall wurden mehrere X4-2 Server, auf denen sich später die Test- und Entwicklungs-Datenbanken aufhalten sollten, direkt angeschlossen.

[2] Configuring a Single Oracle ZFS Storage Appliance into an InfiniBand Fabric with Multiple Oracle Exadata Machines, http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/multiple-exadata-zfssa-121013-2080035.pdf

Der Infiniband (IB) Bus mit Native Quad Data Rate (QDR, 40Gb/s) Infiniband Ports hat eine Daten-Transferrate von 32Gb/s. In der Exadata Database Machine X4-2 sind zwei IB-Switches 2 x 36 port QDR (40 Gb/sec) verbaut. Die ZS Appliance 3-4 besteht aus zwei Köpfen, die zusammen einen Cluster bilden. Pro Kopf sind zwei Infiniband Host Channel Adapter (HCA) vorhanden.

Die ZS wird dabei auf Leaf Switch Ebene in den Exadata-Verbund eingearbeitet (siehe Abbildung 1). Die Neuverkabelung kann ohne Downtime der Exadata-System erfolgen, während der Wartungsarbeiten wird allerdings die Redundanz in der IB-Fabric aufgehoben.

Storage-Konfiguration für das Kloning auf der ZFS Appliance

ZFS arbeitet nach dem Allocate-On-Write-Verfahren. Hierbei werden Blöcke nicht aktualisiert, sondern Änderungen werden stets auf neu zugewiesene Blöcke geschrieben. Nach dem Schreiben gibt es daher zwei Versionen des betreffenden Blocks: Die Version, die vor dem Schreibzugriff existiert hat, und die geänderte Version. Der Speicherort jedes Blocks wird zusammen mit der Version in den sogenannten Metadaten abgelegt. Ist nun ein neuer Block geschrieben worden, müssen diese Metadaten aktualisiert werden. Auch dies geschieht durch Anlegen eines neuen Metadaten-Blocks. Diese Änderungen setzen sich in der Baumstruktur des ZFS nach oben fort, bis schließlich der Uber-Block, der die Wurzel darstellt, in einer neuen Version geschrieben wird. Der bisherige Uber-Block zeigt in diesem Fall auf den bisherigen Zustand in der Vergangenheit.

So ist es einfach, eine Ansicht auf den bestehenden Zustand des Dateisystems „einzufrieren“, also einen Snapshot zu erstellen, indem einfach ab dem Zeitpunkt, zu dem der Snapshot erstellt wurde, keine obsoleten Blöcke mehr gelöscht werden. Im Laufe der Zeit, in dem die Snapshots existieren, wird der frei Speicherplatz, wie bei jedem Snapshot-fähigen Storage-System, daher immer geringer. Das Erstellen des Snapshots selbst dauert keine wahrnehmbare Zeit, und es können beliebig viele[3] Snapshots angelegt werden. Das Schreiben auf ein Dateisystem, welches einen oder mehrere Snapshots aufweist, ist in ZFS nicht langsamer als ohne Snapshots, da nicht doppelt geschrieben werden muss.

Klone sind beschreibbare Kopien von Snapshots. Das Erstellen eines Klones ist ebenfalls ohne Zeitaufwand möglich, und es sind beliebig viele Klone erstellbar.

Für das Lesen sieht ZFS einen Cache vor, der im Hauptspeicher (ARC) sowie mit Hilfe von SSDs („Logzilla/L2ARC“) realisiert werden kann. In einem Solaris-System verwendet ZFS in der Regel sämtlichen vorhandenen Speicher für den Cache, wenn nicht der Parameter zfs_arc_max zur Eindämmung gesetzt ist. Dies wird in der Regel notwendig, wenn Datenbanken auf dem ZFS-Server selbst laufen.

Für das Schreiben sieht ZFS einen weiteren Cache vor, das ZFS Intent Log (ZIL, oder Writezilla). Dieser entspricht in seiner Funktion dem Redo Log einer Datenbank. Alle Änderungen im ZFS werden normalerweise, beim asynchronen Schreiben, zunächst im Hauptspeicher abgelegt. In regelmäßigen Abständen, alle 5 Sekunden, werden sie auf Disk weggeschrieben. Beim Synchronen Schreiben[4] geschieht dies allerdings sofort, und zwar in das ZIL.

[3] 2^64 (18 Quantillionen)

[4] Synchones Schreiben wird durch einen fsync()-Call oder O_DSYNC ausgelöst.

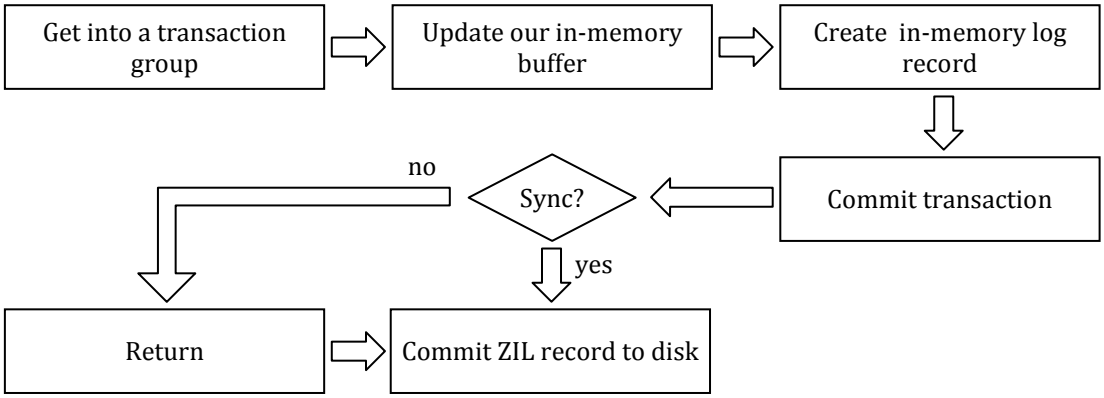

Abb. 3: Schematische Erklärung: Schreiben in ZFS

Dieses enthält lediglich die Änderungs-Vektoren und wird meist auf SSDs angelegt[5]. Pro Zpool gibt es ein ZIL. Geht der Inhalt des RAM verloren, ist es möglich, aus dem ZIL einen konsistenten und aktuellen Zustand des ZFS Datasets zu erzeugen.

Werden die SSDs für das Writezilla gestriped aufgesetzt, erhält man zwar die maximale Performance, aber es darf keinen Doppelfehler auf beiden SSDs geben, denn sonst entsteht Datenverlust. Werden sie gespiegelt aufgesetzt, entspricht die maximale Schreibrate im Latency-Modus der Bandbreite, mit der auf eine SSD geschrieben werden kann. Diese ist in der Regel weit geringer als die Bandbreite zu einem Array aus mehreren HDDs. Daher bietet ZFS die Konfigurationsoptionen *throughput* und *latency* an, mit denen man auf Dataset-Ebene vorgeben kann, ob man den maximalen Durchsatz (ohne Nutzung der SSDs zum Schreiben) oder die geringste Latenz erhalten, also die SSDs verwenden möchte.

Einrichten der ZFS-Shares für die Datenbanken
Folgendes Layout hat sich in unserer Umgebung als sinnvoll erwiesen:

Share	*Logbias*	*Recordsize*	*Primarycache*	*Compression*
DATAFILES	latency[6]	db_blocksize	all	LZJB
INDIZES	latency	db_blocksize	all	off
TEMP	latency	128k	all	off
LOGFILES	latency	128k	all	off
CONTROLFILES	latency	128k	all	LZJB
ARCHIVELOGS	throughput	128k	all	LZJB
RMAN	throughput	128k	all	LZJB

Wie viele Pools sollten erzeugt werden?
Das Layout der ZPools sollte vor der Einrichtung der ZFS Storage Appliance sorgfältig geplant werden. Oracle empfiehlt grundsätzlich, ZFS als Datenbank-Storage mit verschiedenen Pools einzurichten: Einem für die Datafiles und einem für die Redo Logs . Weiterhin ergibt sich für die Klon-DB-Shares meist aus Performance-Gesichtspunkten die Anforderung, einen MIRRORED Pool

[5] In der ZA werden hier speziell für das Schreiben geeignete SSDs verbaut.

[6] Logbias sollte für diese Shares laut Logik und Oracle-Empfehlung auf Throughput stehen, wir haben allerdings mit latency die besseren Ergebnisse erzielt.

zumindest für die Datendateien zu verwenden, während für die Archive Logs und das RMAN Backup RAIDZ-konfigurierte Pools ausreichend und hinsichtlich der Platz-Ausnutzung geeigneter wären.

Gegen das Einrichten mehrere Pools spricht allerdings der höhere Verschnitt bei den Festplatten, die nur einem Pool zugewiesen werden können. Außerdem ist die Pool-Einrichtung sehr langfristig zu betrachten, denn einmal einem Pool zugeordnete Platten können nicht mehr entfernt werden, und daher können einmal eingerichtete Pools auch weder umkonfiguriert noch verkleinert werden. Hot Spare Disks können auch nur einem Pool zur Zeit zugeordnet werden.

Aus diesen Gründen wurden in unserem Projekt pro Head nur ein Pool konfiguriert.

Verschiedene Möglichkeiten für Datenbank-Klone
Um nun einen Datenbank-Klon zu erzeugen, existieren mehrere Möglichkeiten:

- Klon auf Dateisystem-Ebene: Von existierenden Datenbanken können jederzeit Klone erstellt werden. Wenn die Datenbank zum Zeitpunkt der Erstellung lief, muss die Klon-Datenbank ein Recovery durchlaufen, bevor sie geöffnet werden kann. Hierfür ist natürlich Zugriff auf die Archive Logs nötig. Dieser Ansatz ist möglich, wenn Quell-DB und die spätere Klon-DB auf dem gleichen ZFS Server liegen, oder wenn mit „zfs send“ repliziert werden kann.
- Klon mit RMAN: Die Source-DB wird mittels RMAN als Backup Copy auf einen per NFS gemounteten Share der ZA gesichert. Dann wird von diesen Sicherungen auf der ZA ein Klon erzeugt. Nun können diese Klone auf dem Ziel-Server per NFS eingehängt und die Datenbank dort gestartet werden.
- Klon mit DataGuard: Hier wird DataGuard statt RMAN verwendet. Dies bietet erweiterte Möglichkeiten beispielsweise für ein Point In Time Klon, oder für eine Verbindung mit einem Disaster Recovery Ansatz[7].

Für die Bereitstellung des Test- und Entwicklungs-Datenbanken ist es in diesem Projekt erforderlich, diverse Anpassungen durchzuführen, bevor aus der Produktions-Datenbank Klone erzeugt werden können. Diese umfassen:

- Das Anpassen von Directory Locations
- Definition von Services
- Anlegen und Ändern von Benutzern, Gruppen und Rechten
- Anpassen von Database Links
- sowie eine Anonymisierung eines Großteils der Daten. Insbesondere alle personenbezogenen Daten mussten in unserm Fall anonymisiert werden, bevor sie in die weniger gesicherten Dev/Test-Datenbanken gespielt werden konnten.

Aus diesem Grund wird ein zweistufiger Ansatz gewählt: Die Produktionsdaten werden zunächst mittels RMAN auf die ZA restauriert. Dort werden die Anpassungen durchgeführt, und dann werden die Snapshots / Klone auf Filesystem-Ebene ausgeführt.

[7] Siehe Oracle White Paper „Oracle Database Cloning Solution Using Oracles Sun ZFS Storage Appliance and Oracle Data Guard“

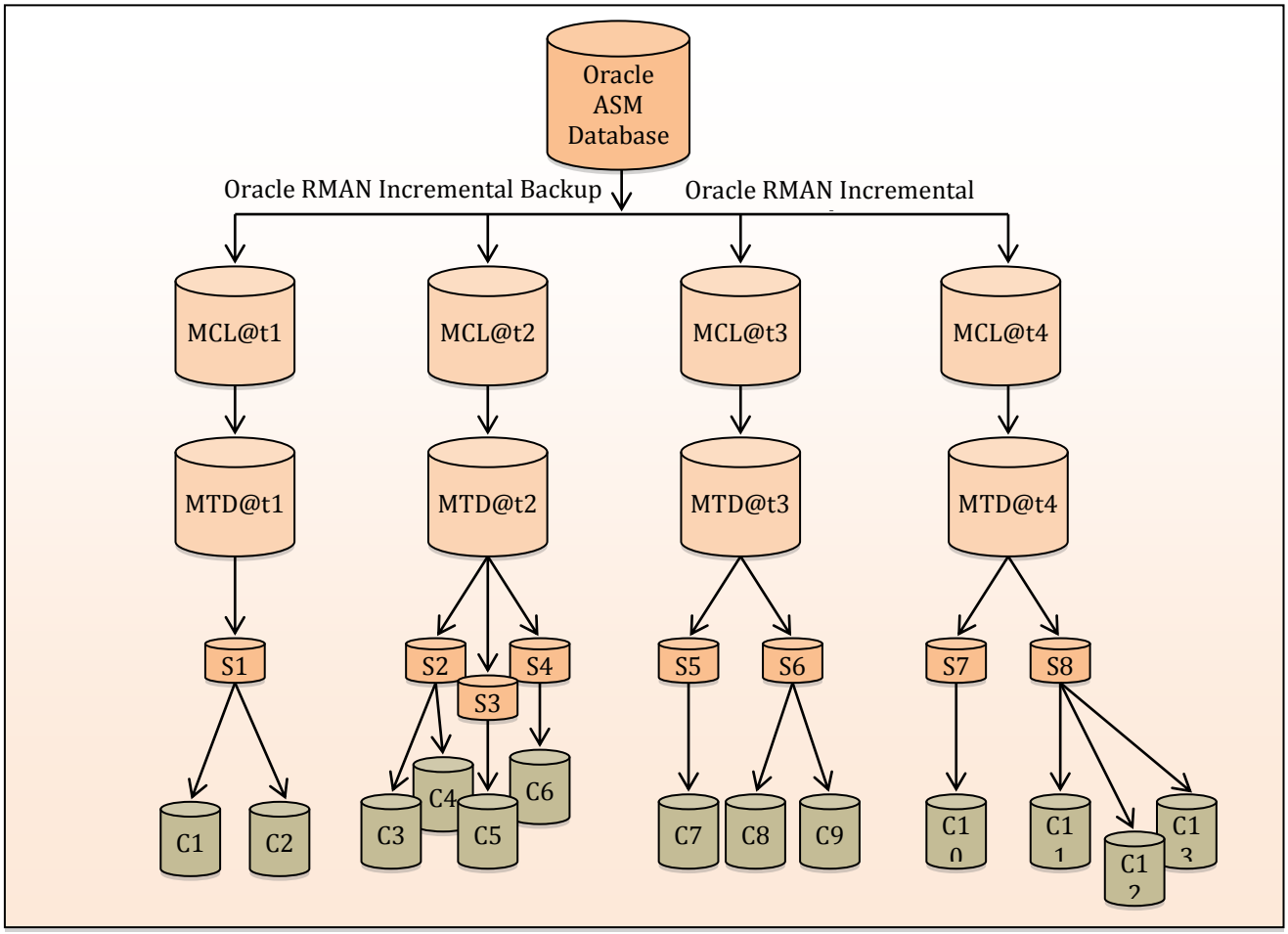

Abb. 4: Ablauf: Datenbank Klone aus inkrementellen Sicherungen erstellen

Zusammenspiel von RMAN und ZA
RMAN sichert normalerweise als Backup Sets. Es gibt allerdings auch die Möglichkeit, in eine Backup Datenbank Struktur zu sichern, indem der Modus AS COPY verwendet wird. In diesem Modus lassen sich auch verschiedene inkrementelle Sicherungen in einer Struktur zusammenführen. Dieses Feature heißt „incrementally updated backup“.

Das Backup wird zunächst als AS COPY von der Produktions-Datenbank auf die ZA gesichert. Hierfür wird in der Datenbank ENABLE BLOCK TRACKING aktiviert. Nach einem Level 0 Backup können beliebige Level 1 Sicherungen durchgeführt werden. Anschließend werden mittels BACKUPS AS COPY SKIP INACCESSIBLE (ARCHIVELOG ALL) die Archive Logs übertragen. Ein Backup AS COPY dauerte in unseren Tests um den Faktor 1.75 länger als AS BACKUPSET.

Zusammenspiel von Kloning und ZA: SMU
Für das Zusammenspiel des Kloning-Porzesses, der ZA und der Datenbank sind einige Arbeitsschritte erforderlich:

- Alle zusammengehörigen Dateien müssen konsistent geklont werden
- Schreiboperationen der Datenbank müssen auf Redo ausgelagert werden
- Archive Logs müssen berücksichtigt werden

Für diese Operationen hat Oracle extra ein separates Tool entwickelt: Das Snap Management Utility (SMU). Dieses arbeitet sowohl mit iSCSI als auch mit (d)NFS und verfügt für die Durchführung dieser Aufgaben über ein Web-GUI sowie ein CLI, welches auch gescripted werden kann. Es unterstützt Datenbanken Oracle 10 und 11, RAC, Linux und Solaris sowie den RMAN.
Das SMU geht beim Erstellen eines Klons nach folgendem Schema vor:

1. Vom Backup Share wird ein Klon angefertigt
2. Das Klon-Share wird auf dem Target Host gemountet
3. SMU startet eine temporäre Datenbank-Instanz aus dem Backup und mountet das Controlfile, um Werte wie maxSCN und FRAU-Größe auszulesen
4. Ein neues Parameter-File wird erstellt und die Klon-Datenbank gestartet
5. Ein neues Controlfile wird erstellt
6. Die Datenbank wird recovered
7. Die Datenbank wird mit OPEN RESETLOGS gestartet
8. SMU rekompiliert alle Schema-Objekte

Client-Anbindung: iSCSI oder dNFS
Die ZA bietet verschiedene Netzwerkprotokolle auf verschiedenen Schichten an. Auf der Hardware-Ebene stehen neben FC IB und (10G) Ethernet zur Verfügung.
Über IB und Ethernet wird TCP/IP verwendet. Die ZA bietet eine Reihe von IP-basierten Protokollen an: ftp, smb, nfs, iSCSI.
iSCSI stellt Volumes (ZFS datasets) als raw devices zur Verfügung. Dieses können als ASM-Volumes verwendet werden.
Oracle empfiehlt dNFS als Zugriffsprotokoll auch für die Datenbank-Dateien. dNFS ist ein NFS-Client, der direkt in die Oracle-Datenbank-Binaries einkompiliert wird. Kurze Wege sollen eine erhebliche Performance-Verbesserung im Vergleich zu Kernel-NFS bieten.
Für das Arbeiten mit Snapshots stellt sich der Einsatz von iSCSI-Volumes mit ASM als unhandlich dar, da diese nur im Ganzen erstellt werden können. Es ist auch nicht möglich, mehr als eine ASM-Instanz auf einem System zu betreiben, daher schied diese Variante aus.

Weitere Best Practices für den Einsatz
Im folgenden werden die von Oracle empfohlenen und in unseren Tests erfolgreichsten Konfigurationen zusammen mit einigen Anmerkungen zusammengestellt.

- Infiniband-Konfiguration
  - LACP Aggregation sollte aktiviert werden
  - Die Active/Active Konfiguration für IPMP wird nicht empfohlen, obwohl es die Schreibrate erhöht[8]
  - Linux ifcfg-ibx: MTU=65520
  - Connected Mode verwenden
- Exadata / Backup Konfiguration
  - Der RMAN Channel für SBT_TAPE sollte ohne compression verwendet werden, wenn die verwendete Software selbst Deduplizierung oder Kompression anwendet. In unserem Fall wurde EMC DBboost verwendet
  - RMAN Kompression ist sehr CPU intensiv – hier sind vor dem Einsatz Tests erforderlich
  - Wenn möglich, mehrere RMAN Channels verwenden
  - Zum Mounten der Backup-Shares kann ein Init.d-Skript geschrieben oder besser der Automounter verwendet werden
- ZFS Pool Konfiguration
  - In einem ZA Cluster sollte nur ein Pool pro Kopf eingerichtet werden, wenn es keine zu unterschiedlichen Anforderungen an die Storage-Charakteristik gibt, sonst ist der Verschnitt größer als notwendig

[8] Siehe MOS Note: 283107.1

    - Die Konfiguration der SSDs als Writezilla kann man, einmal eingefügt, in der ZA nicht mehr ändern, wenn man sie einmal aktiviert hat, da sie dem Pool hinzugefügt werden, obgleich dies in normalen ZFS funktioniert
    - MIRRORED statt RAIDZ2, wenn Performance erwünscht ist
    - Das Cache Device wird im ZA Cluster beim Schwenk nicht auf den zweiten Kopf übernommen
    - Die Cache Devices benötigt nach dem Neustart eines Kopfes eine Aufwärm-Phase, bis der Cache gefüllt ist
    - Oracle empfiehlt 4 Schreib-SSDs für Klone und inkrementelle Backups
    - Checksum Fletcher4
    - Aller Services außer NFS und SFTP können deaktiviert werden
- ZFS Share Einstellungen
    - Ein ZA Projekt beinhaltet Voreinstellungen für Shares
    - Ein ZA Share entspricht einem ZFS dataset
    - iSCSI block devices werden immer synchron geschrieben, hier sollte kein latency Modus konfiguriert werden
    - Oracle empfiehlt, Deduplikation auf keinen Fall zu verwenden, vor allem nicht für inkrementelle RMAN Backups, da der Speicherverbrauch für die Deduplikations-Tabellen in der Regel entweder nicht in den RAM passt, oder hier den ZFS Cache dezimiert
    - ZA-Metadaten wie Benutzer werden zwischen den ZA Köpfen synchronisiert, aber Shares werden nicht beidseitig angelegt
    - User und Group IDs der Share-Voreinstellungen sollten auf 1001 gesetzt werden
    - Es empfiehlt sich, ein eindeutiges Namens-Schema für Shares und Snapshots zu erstellen und einzuhalten
- SMU-Konfiguration
    - Der Einsatz des SMU setzt voraus, dass die Archive Logs in einem von der Datenbank getrennten Share / einer separaten LUN gesichert werden. Dies entspricht aber auch dem Best Practice Konzept für die Einteilung der Shares.
    - Alle Shares für eine Datenbank müssen auf dem gleichen Kopf einer Cluster-ZA liegen. Dies unterminiert das Ansinnen, die Last einer Datenbank auf zwei Köpfe zu verteilen.
    - Wird von einem RMAN Backup geklont, darf auf dem RMAN Share nicht außer diesem Backup liegen. Dieses muss im Image Copy Format vorliegen und das Controlfile umfassen.
- Manuelles Klonen
    - Das Löschen der obsoleten Snapshots automatisieren
    - Zum Aktualisieren eines Klones (refresh), muss der bestehende Klon gelöscht und ein neuer Snapshot erzeugt werden. Wenn mehrere Klone, die auf einem Snapshot basieren, erneuert werden müssen, sollte zuerst der Snapshot gelöscht werden – alle darauf basierenden Klone werden dann ebenfalls gelöscht
- Linux-Client-Konfiguration
    - Laut Aussage eines Oracle-Beraters ist Solaris Kernel NFS durch spezielle Optimierungen schneller als dNFS unter Linux
    - Oracle empfiehlt folgende Mount-Optionen[9]:
        - Linux:
          rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,vers=3,timeo=600

[9] Oracle White Paper „Protecting Oracle Exadata with the Sun ZFS Storage Appliance“, 2013

        - Solaris:
          rw,bg,hard,nointr,rsize=1048576,wsize=1048576,proto=tcp,vers=3,forcedirectio
    - Linux native NFS kann abgeschaltet werden;
      ```
      # chkconfig portmap on
      # service portmap start
      # chkconfig nfs on
      # service nfs start
      # chkconfig nfslock on
      # service nfslock start
      ```
    - Empfohlene[10] Linux /etc/sysctl.conf- Settings:
      ```
      net.ipv4.tcp_timestamps=0
      net.ipv4.tcp_sack=0
      net.core.netdev_max_backlog=250000
      net.core.rmem_max=16777216
      net.core.wmem_max=16777216
      net.core.rmem_default=16777216
      net.core.wmem_default=16777216
      net.core.optmem_max=16777216
      net.ipv4.tcp_mem="16777216 16777216 16777216"
      net.ipv4.tcp_rmem="4096 87380 16777216"
      net.ipv4.tcp_wmem="4096 65536 16777216"
      ```
    - NFS Version 3 wird vorausgesetzt

Klonen der Oracle-Homes
Werden separate ORACLE_HOMEs für jede Klon-Instanz verwendet (was den allgemeinen Datenbank Best Practices entspricht und gerade beim patching viele Vorteile bietet), bietet sich an, auch diese zu auf dem ZFS-Storage zu lagern und dort für jede neue Test-Instanz zu klonen, um Speicherplatz zu sparen.

Hierfür bietet sich folgende Vorgehensweise an:

1. Erstellen eines Masters der ORACLE_HOME Installation auf einem ZA Share
2. Erstellen eines ZFS Klones mit den Bordmitteln der ZA
3. Mounten des Klon-Shares auf dem Datenbank-Server
4. Konfigurieren des neuen ORACLE_HOMEs mit dem „clone“ Perl Skript[11]

Performance-Probleme bei der Implementierung
Als anfänglich größtes Problem hat sich die Performance herausgestellt. Insbesondere beim Schreiben aus der Datenbank auf die ZA waren die erreichten Werte anfänglich vollkommen unbefriedigend.

Folgende Werte wurden erreicht:

Konfiguration / Test	*Durchsatz lesend*	*Durchsatz schreibend*
Initiale Konfig, RMAN/OS		260 MB/s

[10] Infiniband IP performance settings, http://serverfault.com/questions/327947/set-up-simple-infiniband-block-storage-srp-or-iser
[11] Oracle-Tool zum Klonen der ORACLE_HOME, siehe MOS Dokument ID300062.1

CTAS, INSERT APPEND		100 MB/s
Kernel-NFS	400 MB/s	
dNFS	1,9 GB/s	
Adaptive Direct IO, NOLOG, throughput bias, rs=128k, IPoIB	1,6 GB/s	
NFS over RDMA	3,5 GB/s	400 MB/s pro Pool
Bündel-Test, Latency Mode	5,5 GB/s mit beiden Köpfen	1 GB/s
ORION Test		600 MB/s (ein Kopf)
_adaptive_direct_read=TRUE		55 MB/s (DOP=10)
Mit Patch BUG 19339320		115 MB/s
_direct_io_wslots=32, parallel_execution_message_size=65536		315 MB/s
Schreiben mit vielen parallelen Sessions		1,2 GB/s
Mit allen SSDs		1,7 GB/s
NFS/IPoIB Durchsatz X4-2 / ZA	1,25GB/s pro Interface	
Ziel mit 2x 4G Wirespeed	*7 GB/s*	*4 GB/s (sustained IO)*
Oracle-Angabe	*27TB/h, 17,3GB/s on ZA*	

Die Tests wurden in der Datenbank mit DBMS_RESOURCE_MANAGER.CALIBRATE_IO, mit RMAN sowie im OS mit dd und IOZone durchgeführt.

Der X4-2 Test-Host war mit Oracle Linux 6.5 UEK3 3.8.13-35.el6uek.x86_64 installiert, die ZA über IB über NFSv3 und IPoIB angebunden. IB war normal auf 40Gbps konfiguriert. Bereits über diese Verbindung wurde nur ein Durchsatz von 2,5Gb/s / 1,25GB/s erreicht.

Die Suche nach dem Performance Bottleneck erstreckte sich über mehrere Oracle Service Requests und schien mehrere Support Engineers verschlissen zu haben. Auch der Einsatz bekannter Oracle-Consultants vor Ort konnte keine besseren Ergebnisse erzielen oder die Diskrepanz zwischen erwarteten und erhaltenen Werten zumindest erklären. Zuletzt wurden die SRs durch Oracle geschlossen und in einen Bug umgewandelt.

Untersucht wurden alle nur erdenklichen Ausgestaltungen der sich aus den möglichen Konfigurationsvarianten auf Storage- und Client-Seite ergebenden Matrix[12]. Hier wurden über ein halbes Jahr hinweg vom Oracle-Support teilweise täglich andere Einstellungsversuche vorgegeben, die oft einen Neustart von ZA und / oder dem X4-2-Client erforderten, und im Anschluss mussten dann die Ergebnisse ausgewertet werden.

Weder ZFS IOPS Inflation noch Block Ganging (Fragmentierung), Fehler bei der IP-Packetierung, im Multipathing, durch den Mellanox-Treiber in Linux oder beim synchronen Schreiben konnte den fehlenden Durchsatz letztendlich erklären. Die erfolgreichsten Versuche wurden mit RDMA[13] erzielt, es konnte allerdings nicht abschließend geklärt werden, ob diese Variante durch Oracle unterstützt wird. RDMA wird in Metalink nicht empfohlen, lässt sich aber im OEL Kernel als Modul nachladen.

[12] Die Ausformulierung dieser Matrix ergibt leider ein Kartesisches Produkt…

[13] Remote Direct Memory Access (RDMA) kann Daten direkt und ohne zusätzlichen CPU-Overhead zwischen Hauptspeichersegmenten übertragen und ist in Solaris 11 die Standard-NFS-Variante.

Am Ende blieb nur der Workaround, die ZA bis zum Vollausbau mit SSD-Cache-Devices auszurüsten, so dass das Endergebnis schließlich hinnehmbar war.

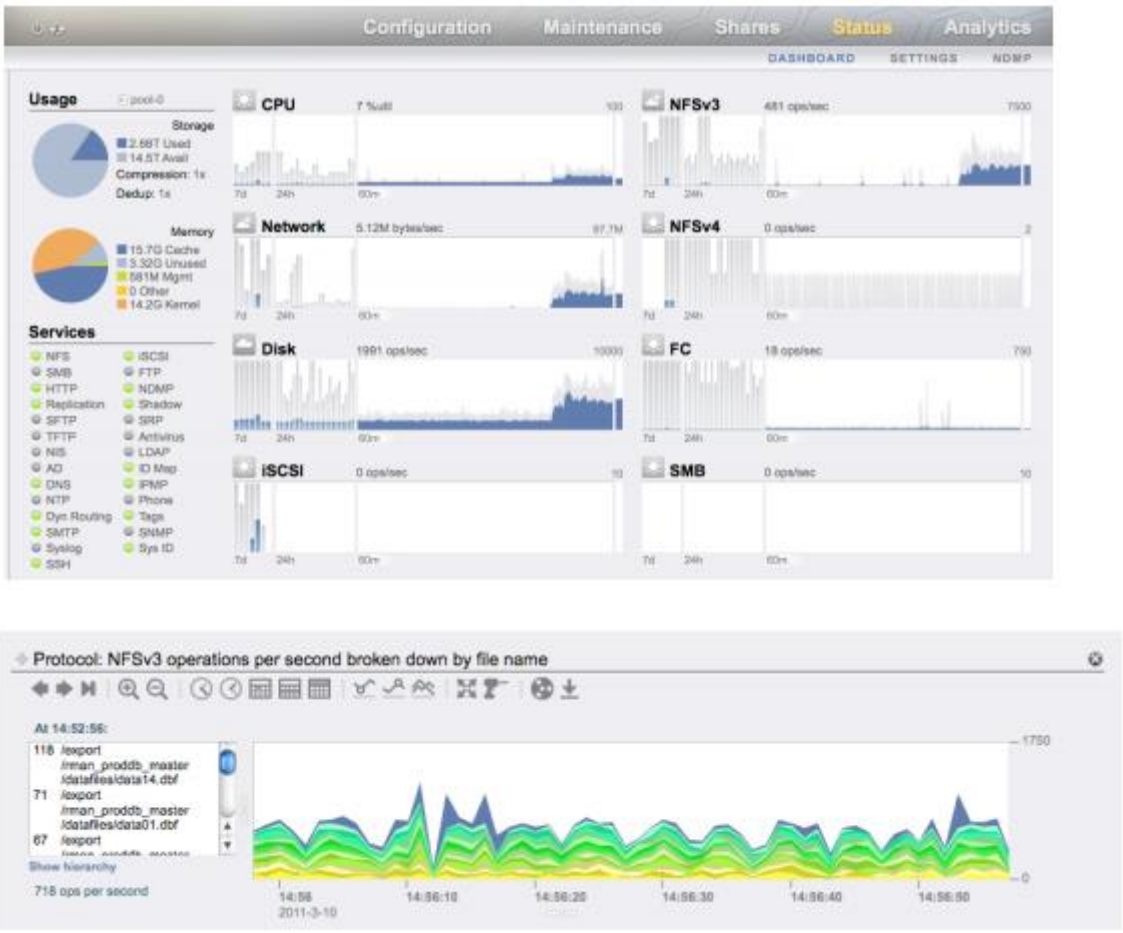

Abb. 5: Lastanalyse mit dem dTrace-basierten ZA Dashboard

Fazit

Das Zusammenspiel von Exadata und ZFS Storage Appliance verspricht durch den kohärenten Ansatz erhebliche Synergie-Effekte. Es ist möglich, Exadata-Racks mit dem ZFS Storage über Infiniband zu verschmelzen und so nicht nur den effizienten nativen Storage-Bus der Exadata direkt an das Storage anzuschließen, sondern auch das ZFS-Dateisystem mit allen Features eines modernen Storage Systems zu verwenden. Diese zeigen sich insbesondere für das Anfertigen von schnellen Klonen aus dem Backup. Die Lösung bietet Technologie im Sinne von Engineered Systems aus einem Guss – bis hin zur durchgehenden Verwendbarkeit von Exadata-Features wie Hybrid Columnar Compresion.
Performance-wise könnte diese Lösung allerdings in unserem Setup nicht mit direct attached Storage mithalten. Es ist schlussendlich während des gesamten Aufbau-Zeitraumes nicht gelungen, die Durchsatzwerte zu erreichen, die man sich erhofft hatte.
Insgesamt scheint die Technologie noch sehr jung und der Oracle-Support hinterließ nicht den Eindruck, selbst auf einen großen Erfahrungsschatz realer Installation zurückgreifen zu können.

Weiterführende Literatur

- „Protecting Oracle Exadata with the Sun ZFS Storage Appliance: Configuration Best Practices“, Oracle White Paper, 2013
- „Database Cloning Using Oracle Sun ZFS Storage Appliance and Oracle Recovery Manager“, Oracle MAA White Paper, 2011
- „Best Practices for Implementing a Data Warehouse on the Oracle Exadata Database Machine“, Oracle White Paper, November 2010
- „Optimizing Storage for Oracle Database 11*g* Release 2 with the Oracle ZFS Storage Appliance“, Oracle White Paper, 2014
- „Configuring Oracle Solaris ZFS for an Oracle Database“, Oracle Technical White Paper, 2014
- „Networking Best Practices with the Oracle ZFS Storage Appliance“, Oracle Technical White Paper, 2014
- „Configuring a Single Oracle ZFS Storage Appliance into an InfiniBand Fabric with Multiple Oracle Exadata Machines“, Oracle Technical White Paper, 2013
- „How to Manage the ZFS Storage Appliance with JavaScript“, Peter Brouwer, OTN 2012
- „Database Thin Cloning: Allocate on Write (ZFS)“, Kyle Hailey on Oracleworld.Com, 2014
- „Clone your dNFS Production Database for Testing“, MOS Note 1210656.1, 2013
- „Oracle Snap Management Utility for Oracle Database“, Oracle White Paper, 2013
- „Sun ZFS Storage Appliance and Oracle IT: Use Cases and Benefits“ Oracle White Paper 2012
- „NFS with native Infiniband“, http://www.linux-mag.com/id/7163/
- „Performance Tuning Guidelines for Mellanox Network Adapters“, http://www.mellanox.com/related-docs/prod_software/Performance_Tuning_Guide_for_Mellanox_Network_Adapters_v1.6.pdf
- „Snap Management Utility for the Oracle Database - Information and Troubleshooting“ (Doc ID 1522925.1)

Kontaktadresse:

Jan Schreiber
Loopback.ORG GmbH
An der Alster 83
D-20099 Hamburg

Telefon:	+49 (0) 40 2263236 0
Fax:	+49 (0) 40 2263236 99
E-Mail	info@loopback.org
Web:	http://www.loopback.org