

SWISS ORACLE USER GROUP

**S O U G**

[www.soug.ch](http://www.soug.ch)

**Newsletter 4/2005**

Oktober 2005



Focus on Performance

Clone-as-clone-can

Oracle auf Mac

Localize Oracle Forms

## Tony Fräfel, SOUG Editorial

Liebe Leserin, lieber Leser

Kaum zu glauben, aber Sie lesen bereits die vierte und letzte Ausgabe für 2005 des Newsletters! Die Zeit vergeht im Fluge – Geschwindigkeit (Performance?) in einem anderen Zusammenhang. Weniger emotional dafür umso hilfreicher sind die Artikel im Focus dieser Ausgabe. Die drei Artikel zum Thema Performance geben Ihnen nützliche Hinweise, wie Sie die Performance im Griff haben bzw. Wie Sie die Performance Ihrer Oracle Datenbank und Ihrer Java Anwendung steigern. Das Lesen dieser Artikel lohnt sich besonders!

Haben Sie auch "mal schnell" eine Kopie der produktiven Datenbank für Tests benötigt? Wie Sie das sicher und effizient machen, auch wenn es mehr als nur einmal ist, lesen Sie im Beitrag von Oli Sennhauser, Telekurs Services Ltd, ab Seite 62. Diesen praxisbezogenen Erfahrungsbericht erlobe ich mit dem Award dieser Ausgabe. Herzliche Gratulation an Oli und prost beim Trinken des Champagners!

Mit dem Focus der nächsten Ausgabe begeben mich etwas auf's Glatteis (ist ja auch die "Winter-Ausgabe"): SOA. Sind serviceorientierte Architekturen eine weitere Hype oder steckt mehr dahinter? Bewegen Sie sich in diese Richtung? Ist die



Ausgabe	Redaktionss	Focus
2006/1	15.	SOA

Architektur Ihrer Systeme schon lange serviceorientiert? Welche Erfahrungen haben Sie damit gemacht oder welchen Nutzen versprechen Sie sich davon? Ich bin gespannt, welche Beiträge wir zu diesem Thema erhalten!

Viel Spass beim Lesen des Newsletters!

Tony Fräfel

Chère lectrice, cher lecteur,

Difficile à croire: vous lisez déjà la quatrième et dernière édition de la Newsletter de cette année! Le temps passe tellement vite - rapidité (performance?) dans un autre contexte. Moins émotionnel mais d'autant plus utiles sont les articles du Focus de cette édition. Les trois contributions au sujet du Focus 'Performance' vous fournissent des indications pratiques; comment maîtriser la performance, respectivement, comment pouvez-vous augmenter la performance de votre base de données Oracle et de votre application Java. Le lecture de ces articles vaut vraiment la peine!

Avez-vous déjà eu <<rapidement>> besoin d'une copie de la base de données productive pour des tests? Comment vous pouvez exécuter cela d'une façon sûre et efficace - même si c'est pour plus qu'une seule fois - vous explique Oli Sennhauser, Telekurs Services Ltd., dans son article à la page 62. Je récompense ce rapport fondé sur la pratique avec l'Award de cette édition. Félicitations cordiales à Oli et: Cheers! pour son Champagne.

Je me rends un peu sur du verglas avec le choix du Focus pour la prochaine édition (bien sûr, c'est l'édition de l'hiver...): SOA. Est-ce que

des architectures orientées 'Service' sont encore un autre Hype ou est-ce qu'il y a quelque chose là-dessous? Vous orientez-vous dans cette direction? Est-ce que l'architecture de vos systèmes est depuis quelque temps déjà en direction de 'Service'? Quelles sont vos expériences respectives et quel est l'utilité que vous en attendez? Je suis curieux de lire vos contributions à ce sujet!

Bien de plaisir à la lecture de la

Edition	Délai de	Focus
2006/1	15	SOA

## Oli Sennhauser, Telekurs Services Ltd. Clone-as-clone-can

### Einführung

Es kann gelegentlich vorkommen, dass aus organisatorischen, architektonischen, applikatorischen oder sonstigen Gründen eine Datenbank täglich neu erstellt werden muss und dies über einen längeren Zeitraum...

Da diese Arbeit für den DBA nach ca. dem 3. Mal langweilig wird und er eigentlich besseres zu tun hätte, ist zu überlegen das ganze zu automatisieren.

Die Entwickler-Datenbanken müssen möglichst produktionsnah, schnell und ressourcenschonend erstellt werden. Hierfür bietet sich die RMAN (Oracle Recovery Manager) Funktionalität "DUPLICATE TARGET DATABASE" an.

Export/Import ist für diesen Zweck zu langsam und ein Restore der Datenbank mit anschliessendem Umbenennen der CREATE CONTROLFILE erfordert manuelle Nacharbeiten.

### Grundlagen

Wenn in diesem Context von "clonen" gesprochen wird, ist das erstellen einer Datenbank mittels der RMAN DUPLICATE Funktionalität gemeint.

Unter Mutter wird die Quell-Datenbank (= TARGET, meist eine Produktions-Datenbank, als Clone die Ziel-Datenbank (= AUXILIARY, eine Test- oder Entwicklungs-Datenbank) gemeint.

Unter "horizontalem clonen" wird das clonen vom Testsystem auf das Testsystem verstanden, während mit einem normalen ("vertikalen") Clonen ein Clone von der produktiven Mutter "runter" auf das Testsystem verstanden wird.

SAN Class A: Schnelles, hochverfügbares RAID 5 mit Remote-Copy Funktion, SAN Class C: Schnelles, hochverfügbares RAID 5 mit preiswerten Disks und Snapshot-Funktion.

Eine wichtige Voraussetzung für eine einfaches, möglichst robustes, fehlertolerantes und wartungsfreundliches System ist ein hoher Grad an Standardisierung. Es empfiehlt sich, sowohl den Server als auch die Datenbank analog aufzusetzen und generische Namen zu verwenden, da diese dann auch in Skripts jeweils aus dem aktuellen Context herleitbar sind (Beispiele: Namenskonventionen für Oracle Instanzen und Server einführen und einhalten, ORACLE\_HOME und ORACLE\_BASE haben immer denselben Pfad, Skripts sind nicht DB-spezifisch geschrieben sondern parameterisierbar, etc.).

Im vorliegenden System wurden das DB-Environment strickt nach OFA-4 (eine Verbesserung der von Oracle empfohlenen Optimal Flexible Architecture) aufgesetzt. Für das OS-Environment wurde das Framework Basenv 2.5 einer renommierten Schweizer Oracle Consulting Firma verwendet und dum das Backup/Restore/Clone-Prozedere zu

vereinfachen wurde rman\_exec (ein äusserst komfortabler Wrapper für RMAN) verwendet.

### System-Umgebung

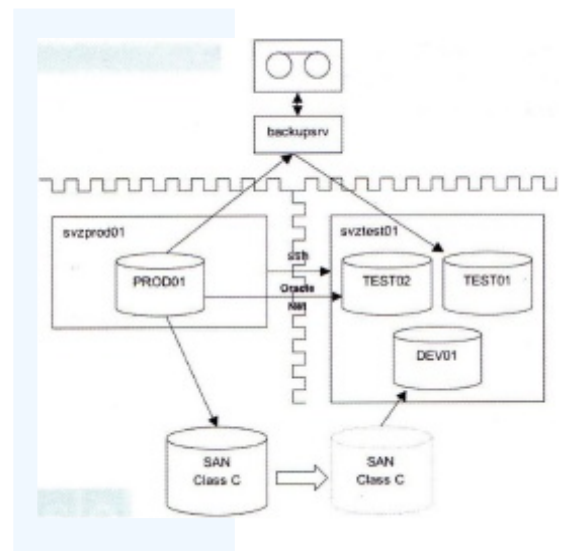
Folgende Hardware-Umgebung wird verwendet:

- \* Produktionssystem: 12 CPU à 1200 MHz, 24 GB RAM, 300 GB SAN (Classe A), 180 GB SAN (Class C) für Cloning.
- \* Testsystem: 8 CPU à 1200 MHz, 16 GB RAM, 900 GB SAN (Class A).

Für die Datenbank wird zur Zeit ORACLE Enterprise Edition Version 9.2.0.4 verwendet.

Die ganze Umgebung ist mittels Firewalls gegen Umwelteinflüsse abgeschottet. Um die Kommunikation zwischen Produktions- und Testrechner respektive Mutter- und Clone-Datenbank zu ermöglichen wurde die Firewall für OracleNet und ssh in eine Richtung geöffnet.

Für Backups resp. das Clonen stehen entweder das SAN-C Disksystem und/oder ein Tape-System zur Verfügung.



## Grundsätzlicher Ablauf

Das Clonen erfolgt täglich automatisiert, kann aber auch, in bestimmten Zeitfenstern, manuell auf Kundenwunsch von Hand bewerkstelligt werden.

Zeitgesteuert (crontab) wird um 03:25 mittels rman\_exec ein Full-Backup (inc0 + archives) auf Disk (SAN C) gemacht. Vorgängig laufen noch die Nachtverarbeitung und die Datenaggregation des letzten Tages.

Das Backup auf Disk dauert je nach Menge der zu sichernden Archive-Logs und IO-Last 40-50 Minuten für aktuell 120 GB.

Anschliessend wird auf das Mounten des Spiegels der SAN-C Disk auf dem Testsystem svwtest01 gewartet, welches ebenfalls Zeitgesteuert um 05:15 erfolgt. Dies wird durch Auslesen des erwarteten Mountpoints mittels eines ssh-Connects gemacht.

Sobald der Mountpoint mit den Backup-Daten auf dem Testsystem verfügbar ist, wird das Skript für das Clonen der jeweiligen DB generiert und diese geclonet. Anschliessend wird remote ebenfalls über ssh eine applikationsspezifisches Postprozessing gestartet.

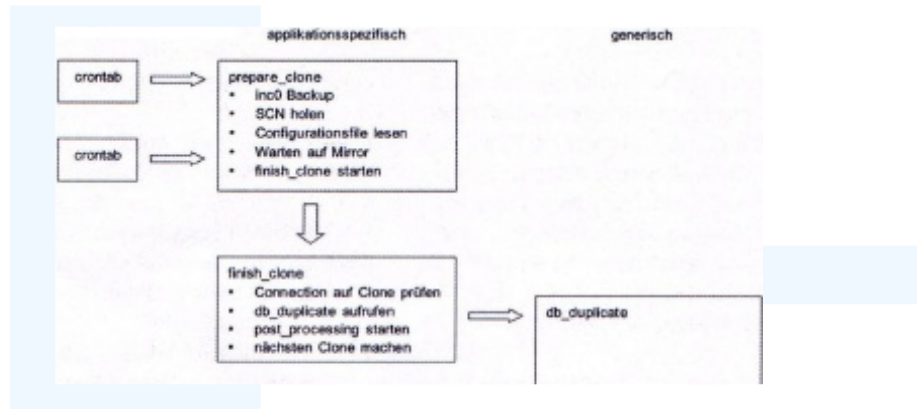
Das ganze Clonen dauert pro Datenbank jeweils ca. 1 Stunde, sodass die Datenbanken zu Arbeitsbeginn bereit stehen.

Während des Tages können auf Wunsch beliebig weitere Clones angefordert werden.

Um 17:00 wird der Spiegel des SAN-C auf dem Testsystem wieder abgehängt und die Backupsets auf Tape ausgelagert.

## Skript-Architektur

Um den Mechanismus auch in anderen Projekten verwenden zu können, wurde die benötigte Skriptlogik in einen applikationsspezifischen (prepare\_clone und finish\_clone) und in einen generischen Teil (db\_duplicate) gesplittet.



Die Skripts wurden so konzipiert, dass sie durch Parameter gesteuert werden. Dies macht den ganzen Mechanismus einerseits voll automatisierbar, andererseits ermöglicht es den Einsatz auf mehrere Arten (Clonen via Disk oder Tape, Clonen im ArchiveLog-Modus oder nicht, horizontale oder vertikale Clones, etc.).

## Skripts im Detail

### prepare\_clone

Dieses Skript wird um 03:25 aus der Crontab gestartet. Da es sich um applikationsspezifische Vorbereitungen handelt, ist eingiges hart codiert (Name der Mutter, e-Mail-Adresse für Alarmierung, rman\_exec Parameter, Name des Testservers, etc.) und das Skript wird ohne Parameter aufgerufen.

Zuerst wird ein Backup auf Disk (auch auf Tape möglich) gemacht. Anschliessend wird die letzte SCN des letzten gesicherten archivierten Redologs benötigt. Dies ist der Zeitpunkt auf welchen die Clones bereitgestellt werden (ca. 04:25). Diese SCN wird in einem File zwischengespeichert und in den Folgeskripts benötigt.

Anschliessend wird aus einem Konfigurationsfile ausgelesen, welche Clones am heutigen Tag benötigt werden. Dieses Konfigurationsfile hat folgendes Format:

```
Mon: TEST01, TEST02, DEVELO1, DEVELO2
Tue: TEST01, TEST02, DEVELO1
Sat: none
Sun: none
*: TEST01
```

Nachdem das Backup fertig ist und klar ist, welche Datenbank auf welchen Zeitpunkt geclont werden soll, wird gewartet, bis die Disks, auf welchen das Backup liegt, von der Produktion auf das Testsystem gespiegelt und gemounted sind.

```
# Waiting in minutes
waitTime=120
ret=1
i=0
while [ ${i} -lt ${waitTime} ] ; do
    sleep 60
    let c="$(ssh oracle@${server} "ls -l ${ORACLE_BASE}/admin/${Mother}/backup/* 2>/dev/null | wc
-l")"

    if [ ${c} -gt 0 ] ; then
        ret=0
        break
    fi
    let i=${i}+1
done
```

Sobald die Disks mit den Backups gemounted sind, wird das Skript finish\_clone aufgerufen. Sollten die Disks einmal nicht gemounted werden, bricht das Skript ab und schickt ein Mail an das DBA-Team, welche das Problem zu Arbeitsbeginn beheben kann.

### finish\_clone

Dieses Skript ist zuständig für das Erstellen der einzelnen Clones. Es ist sowohl im automatischen Prozess, welcher am Morgen früh abläuft eingebunden als auch von Hand unter Tags aufrufbar, sofern die Backups vorhanden sind.

Da auch dieses Skript applikationsspezifisch ist, sind einige Informationen hart codiert (Name der Mutter).

Als Parameter können übergeben werden:

- \* Liste der Clones, welche erstellt werden sollen.
- \* ARCHIVELOG-Modus oder den NOARCHIVELOG-Modus in welchen die Clones gestellt werden sollen.
- \* e-Mail-Empfängerliste im Fehlerfall

Ein weiterer Parameter, die SCN, liegt im entsprechenden File vor (siehe oben). Die Übergabe diese Parameters wurde mittels File gelöst, damit sich der DBA unter Tags beim manuellen Clonen nicht um den Zeitpunkt kümmern muss, auf welche der Clone erstellt werden soll.

Nun wird über die Cloneliste iteriert. Als erstes wird der Connectstring für die Clone-Datenbank benötigt. Mit

diesem Connect-String wird geprüft, ob die zu überclonenden Datenbank überhaupt erreichbar ist (diese muss mindestens im NOMOUNT Status sein).

Ist die zu überclonende Datenbank erreichbar, wird das applikationsneutrale Skript db\_duplicate gestartet. Wenn nicht wird versucht, den nächsten Clone in der Cloneliste zu erstellen.

Ist das Skript db\_duplicate erfolgreich durchlaufen, wird remote ein applikatorisches Postprocessing gestartet, siehe unten.

```
.../db_duplicate.ksh -d ${Mother} -c ${Clone} -m disk ${ArchiveParameter} -s ${scn} >>
db_duplicate.log 2>&1
r=${?}

if [ ${r} -ne 0 ] ; then
    echo "ERR: DB duplicate failed."
    let ret=${ret}+${r}
else

    # do Postprocessing

    cmd=".../clone_post_processing.ksh ${Clone} >> .../clone_post_processing.log 2>&1 &"

    echo "Starting post processing in background."
    cloneServerName="$(tnsping ${Clone} | grep 'Attempting to contact' | tr '(' '\n' | grep
Host | cut -d '=' -f 2 | sed s/\)//g)"
    echo "    Clone ${Clone} is on server ${cloneServerName}."

    cmd="nohup ssh oracle@${cloneServerName} '${cmd}' &"
    echo "    Logfile is remote on ${cloneServerName}."
    eval ${cmd}
fi
```

## db\_duplicate

Dies ist das eigentliche Kernstück des ganzen Clone-Prozesses. Damit das Skript auch für andere Applikationen verwendet werden kann wurde es voll parametrisierbar und applikationsunabhängig geschrieben. Es wird auch bereits in weiteren Projekten verwendet.

Folgende Parameter stehen zur Verfügung:

- \* Parameter zur Steuerung des ARCHIVELOG-Modus
- \* Parameter für Name der Mutter
- \* Parameter für die Clone-Liste

- \* Möglichkeit zu forcieren eines Clones mit Produktiv-Sattus (DB-Name: P...).
- Produktive Datenbanken lassen sich sonst durch dieses Skript nicht überclonen. Produktive Datenbanken fangen in dieser Umgebung zwingend mit dem Buchstaben P an.
- \* Parameter für verwendetes Backup-Medium (Disk oder Tape).
- \* Parameter für SCN.

Zuerst werden die Server für Mutter und Clone anhand des Servicenamens bestimmt. Dann wird die Verbindung auf die zu clonende DB (AUXILIARY) geprüft. Der

Connect-String wird wiederum über ein Konfigurationsfile ermittelt.

Danach wird das RMAN-Skript zusammengestellt, mit welchem geclont werden soll. Es besteht zudem die Möglichkeit anhand einer Mountpoint Liste, Datafiles für den Clone von einem Mountpoint auf deinen anderen zu legen. Dies ist jedoch nur für alle Datafiles eines Mountpoints gemeinsam möglich. Beispiel: /u00/oradata/SID/\* -> /u04/oradata/SID/ etc.

```
connect target /
connect auxiliary sys/manager@TEST01

run
{
  startup clone force nomount;

  ALLOCATE AUXILIARY CHANNEL ch1 DEVICE TYPE disk;
  ALLOCATE AUXILIARY CHANNEL ch2 DEVICE TYPE disk;
  ALLOCATE AUXILIARY CHANNEL ch3 DEVICE TYPE disk;

  SET UNTIL SCN 14120181272;

  SET NEWNAME FOR DATAFILE 1 TO '/u01/oradata/TEST01/system01TEST01.dbf';
  SET NEWNAME FOR DATAFILE 2 TO '/u01/oradata/TEST01/undo01TEST01.dbf';
  ...
  SET NEWNAME FOR DATAFILE 70 TO '/u01/oradata/TEST01/i21401TEST01.dbf';

  DUPLICATE TARGET DATABASE TO 'TEST01'
  LOGFILE
  GROUP 1 (
    '/u00/oradata/TEST01/redog1m1TEST01.dbf',
    '/u02/oradata/TEST01/redog1m2TEST01.dbf'
  ) SIZE 128M,
  GROUP 2 (
    '/u00/oradata/TEST01/redog2m1TEST01.dbf',
    '/u02/oradata/TEST01/redog2m2TEST01.dbf'
  ) SIZE 128M,
  GROUP 3 (
    '/u00/oradata/TEST01/redog3m1TEST01.dbf',
    '/u02/oradata/TEST01/redog3m2TEST01.dbf'
  ) SIZE 128M,
  GROUP 4 (
    '/u00/oradata/TEST01/redog4m1TEST01.dbf',
    '/u02/oradata/TEST01/redog4m2TEST01.dbf'
  ) SIZE 128M
  ;
}

sql clone "ALTER TABLESPACE TEMP ADD TEMPFILE ''/u00/oradata/TEST01/temp01TEST01.dbf'' SIZE 1250M REUSE;
sql clone "ALTER TABLESPACE TEMP ADD TEMPFILE ''/u01/oradata/TEST01/temp02TEST01.dbf'' SIZE 1000M REUSE;
sql clone "ALTER TABLESPACE TEMP ADD TEMPFILE ''/u02/oradata/TEST01/temp03TEST01.dbf'' SIZE 1250M REUSE;

sql clone "ALTER DATABASE RENAME GLOBAL_NAME TO TEST01";

startup clone force dba;
shutdown clone immediate;
startup clone mount;
sql clone "ALTER DATABASE NOARCHIVELOG";
sql clone "ALTER DATABASE OPEN";
```

Zum Schluss wird das Skript RMAN übergeben und damit der Clone-Prozess eingeleitet.

## post\_processing

Dieses Skript, welches auf dem Rechner des Clones liegt, wird nach dem Clonen remote via ssh von der Produktion aus gestartet. Als Parameter dient die SID der jeweiligen Clone-Datenbank.

In ihm werden applikationsspezifische Anpassungen an der Clone-Datenbank vorgenommen. Diese können z.B. sein:

- \* Passwörter um- oder zurücksetzen.
- \* Funktionalitäten welche nur auf der Produktion laufen dürfen ausschalten oder umbiegen.
- \* Daten scrambeln, etc.

Um mögliche Schäden auf der Produktion zu verhindern, ist dieses Skript nur auf der Testumgebung vorhanden.

Codefragment:

```
...
echo "Reset appli cation
password"

case "${Cl one}" i n
"TEST02")
    passphrase="appl /bubu" ;;
"DEVELO2")
    passphrase="appl /ti mpp" ;;
*)
    passphrase="appl /appl " ;;
esac
...
```

## Leistungsgrenzen

Die hier vorgestellte Methode für das Clonen von Datenbanken ist ressourcenintensiv. Sowohl das Disksystem/Tapesystem als auch das Netzwerk müssen in der Lage sein, nebst dem produktiven Betrieb, in kurzer Zeit 1/4 Terabyte zu schlucken.

In der beschriebenen Umgebung stellte vor allem das IO-System (SAN) den Flaschenhals dar. Es wird daher nicht mehr als 1 Clone parallel erstellt, da sonst auf dem Testsystem nicht mehr richtig gearbeitet werden konnte (> 60% IO-Waits).

Da auf das SAN zusätzlich andere Applikationen zugreifen und von den nicht überclonten Datenbanken ebenfalls ein uneinheitlicher Workload verursacht wurde, können die Clone-Zeiten nur ungefähr vorausgesagt werden (es wird zurzeit mit ca. 1 h für einen 120 GB Datenbank gerechnet). Zu Lastspitzen kann es aber durchaus bis zum 2.5 fachen dieser Zeit dauern. Es ist daher mit dieser Umgebung nicht möglich, mehr als 8-10 Clones pro Tag zu erzeugen.

## Zahlen

Seit Anfangs Juni 2004 wird auf diese Art und Weise geclont. Während dieses Jahres wurden bis jetzt (Stand Mitte Juni 2005) ca. 1000 Clones

erzeugt. Bei einer durchschnittlichen Datenbankgrösse von 80 GB (40 GB zu Beginn, heute 120 GB) macht dies 80 TB Daten welche "rumkopiert" wurden.

## Zusammenfassung

Dieser Clone-Mechanismus über Disk has sich als äusserst stabil erwiesen.

Dank der guten Erfahrungen, welche mit diesem System gemacht wurden und der hohen Standardisierung und Flexibilität der verwendeten Skripts wird dieses Verfahren mittlerweile auch für weitere Applikationen eingesetzt.

Mit der beschriebenen Methode steht ein komfortables Setup für das Bereitstellen von Testumgebungen für Softwareentwickler zur Verfügung.

Das Verfahren ist ziemlich ressourcenintensiv und verleitet zu flüchtigerem Arbeiten (der DBA kann ja schnell neu clonen...)

Organisatorische Massnahmen würden, wie in der IT üblich, wesentlich mehr bringen, als technische Ausreizungen der Systeme...

Mein Dank geht an Petra Knöbl (Firma Trivadis) für Ihren Input.

## Contact

Telekurs Ltd.

Oli Sennhauser

E-Mail:  
oli.sennhauser@bluewin.ch