STORAGETEK®

APPLICATION NOTE

SnapVantage software and Oracle databases

Database cloning is legal: Data management solutions using SnapVantage software with Oracle databases

APRIL 2004

Executive summary

StorageTek's virtual disk architecture brings several valuable and time-saving capabilities into the daily routine of an Oracle DBA. In addition to efficiency benefits realized by the DBA, the corporation also experiences cost savings by choosing virtual disk for their database storage needs. This application note explores how these benefits can maximize the corporation's return on enterprise storage investment dollars.

StorageTek's SnapVantage™ software product uses the SnapShot feature of the SVA™ Administrator (SVAA) software to clone database environments. These cloning techniques can increase the DBA's productivity by eliminating the overhead involved in creating non-production database copies for the test, development and reporting organizations. The cloning technique can also be applied to the Oracle software installation, thereby eliminating the time required to re-install the software products on multiple servers.

The CPU time saved by cloning poses an even greater benefit because that processing time can be better used for business processing instead of hot backups and single-threaded datafile copy operations. Hot backup duration is reduced to such a short timeframe that the backup process with SnapShot goes unnoticed by the database user community. In the past, long-running hot backup mode was the main reason for the high CPU consumption and the long runtime required to obtain an online backup of the database.

Single-threaded datafile copies can be replaced by simply copying the pointers of the database datafiles using the SnapShot capability of SnapVantage software. Disk consumption for non-production database copies can be reduced by a minimum of 80 percent, depending on the frequency of change to the primary database. As data warehouse databases approach one terabyte in size, making a copy for use by non-production activities may be cost prohibitive. However, with SnapShot, a one terabyte virtual clone may consume only 200 gigabytes, saving 800 gigabytes in redundant storage compared to the traditional copy method.

Media recovery can also be accomplished with cloned database datafiles, since those datafiles are online and accessible. Recovery involves making file copies with an FTP of the cloned datafiles and applying recovery using redo logs. Although media recovery is automatic with StorageTek's V-Series Shared Virtual Array® (SVA™) disk system, the capability to recover corrupt or missing files is also available at the host level. Now that the capacity required to keep a backup online is negligible, restoring datafiles from backup tapes is a thing of the past. Backup

reliability is also increased by eliminating tape backups, which may not be easily accessible and could be erroneous.

Most importantly, no database down time is required to create a virtual clone using SnapVantage software's SnapShot feature, allowing increased availability of the source database. Database backups no longer have to be disruptive to the user community using this approach. Instead of worrying about reasonable backup completion times and reliable backups of databases, the DBA can rest assured that backup scheduling conflicts and overlaps will not occur when using the SnapVantage software cloning approach. The entire hot backup process using the cloning technique takes only minutes compared to hours. Since there is no interruption in service, backups can be done several times per day if required.

1 Introduction

As our disk usage and databases grow at an increasing rate, we find that management of that data is becoming an arduous task. StorageTek's SnapVantage software can simplify the task by cloning environments that include the operating system, Oracle software installation and the full database environment. Clones may be used for many purposes such as replicating student environments or maintaining application versions. Database clones can be created and deployed in just minutes compared to many hours required for traditional backups or replication methods.

In June of 2002, StorageTek® introduced SnapVantage software as a web-based software feature of StorageTek's SVAA for VM that clones, controls and manages Linux images that run as guest operating systems under VM on an IBM S/390 or z-Series mainframe. SnapVantage software is enabled by the virtual disk technology of the Shared Virtual Array (SVA) disk system and has the ability to create 'zero-capacity' copies utilizing unique and patented SnapShot technology. StorageTek has been approved by Oracle Corporation as a qualified member of the Oracle Storage Compatibility Program (OSCP).

This paper explores the SnapVantage software virtual cloning operation as it applies to Oracle databases in a Linux/390 environment running under z/VM. This paper also presents features of the SnapVantage software product that can be used for database backup purposes to eliminate the performance impact and downtime of making a full backup of a production Oracle database. The SnapVantage software cloning operation also eliminates the excessive hot backup roll-forward activities required when setting table spaces to backup mode. SnapVantage software will revolutionize a primary role of the DBA, which is to ensure that recoverable database backups are performed in a timely fashion.

2 System clones

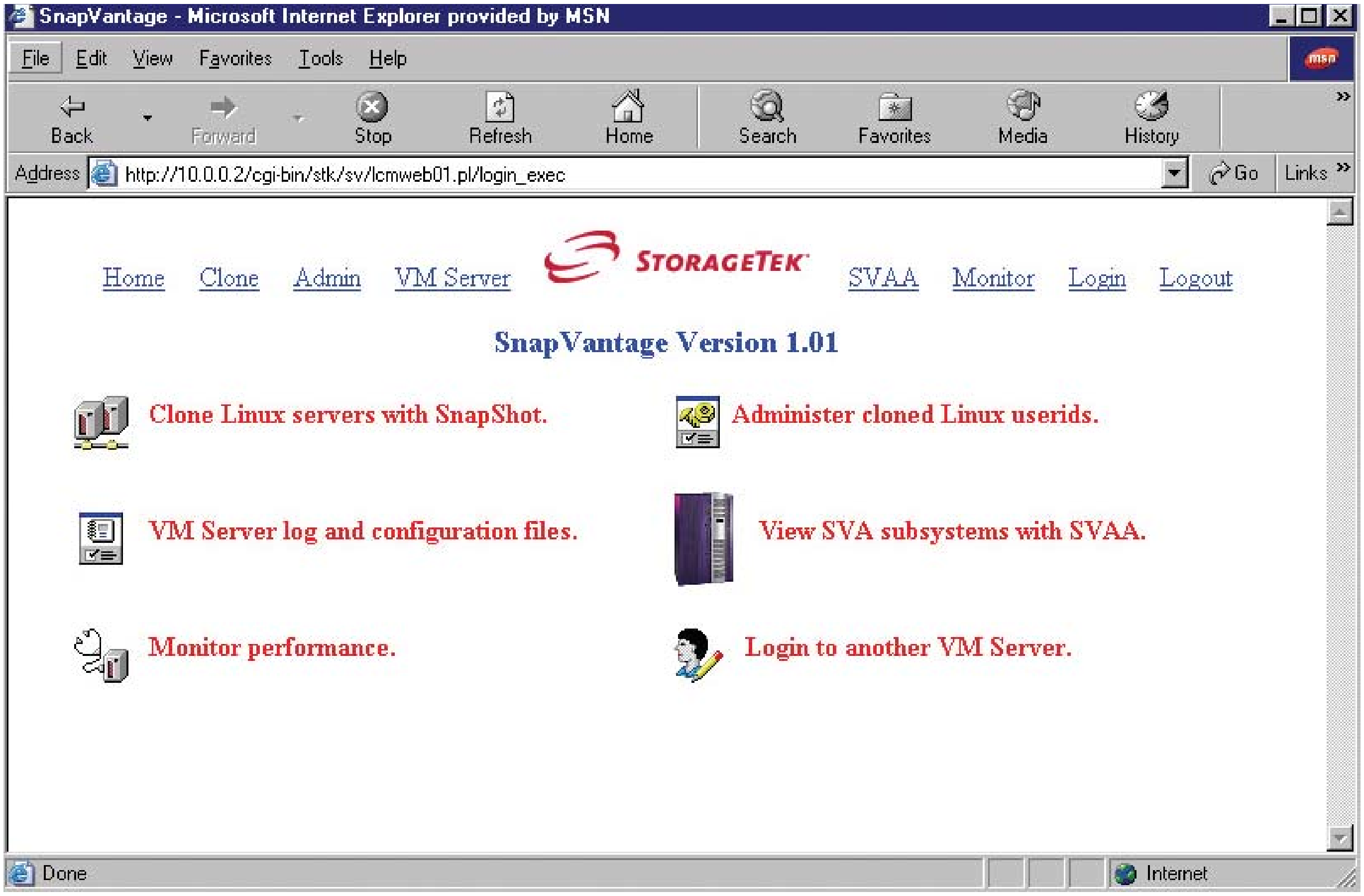
2.1 VM and Linux operating systems

The dynamic allocation, configuration and startup of new Linux server image using SnapVantage software on z/VM takes less than 60 seconds. The customer can reduce the total cost of ownership by using a single footprint to service hundreds of Linux servers. Additional cost savings are possible by requiring less disk storage, less software licenses and less server support.

Actual display screens from the SnapVantage software product are shown in Figures 2a–2b and 3a–6a.

The SnapVantage software Home Page has several options:

- Clone Linux servers with SnapShot
- View VM Server log and configuration files
- Monitor performance
- Administer cloned Linux userids
- View SVA disk systems with SVAA
- Log in to another VM userid or server


SnapVantage - Microsoft Internet Explorer provided by MSN
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History
Address http://10.0.0.2/cgi-bin/stk/sv/lcmweb01.pl/login_exec
Go Links
Home Clone Admin VM Server STORAGETEK SVAA Monitor Login Logout
Done
Internet

SnapVantage Version 1.01

Clone Linux servers with SnapShot.

Administer cloned Linux userids.

VM Server log and configuration files.

View SVA subsystems with SVAA.

Monitor performance.

Login to another VM Server.

Figure 2a. SnapVantage software home page

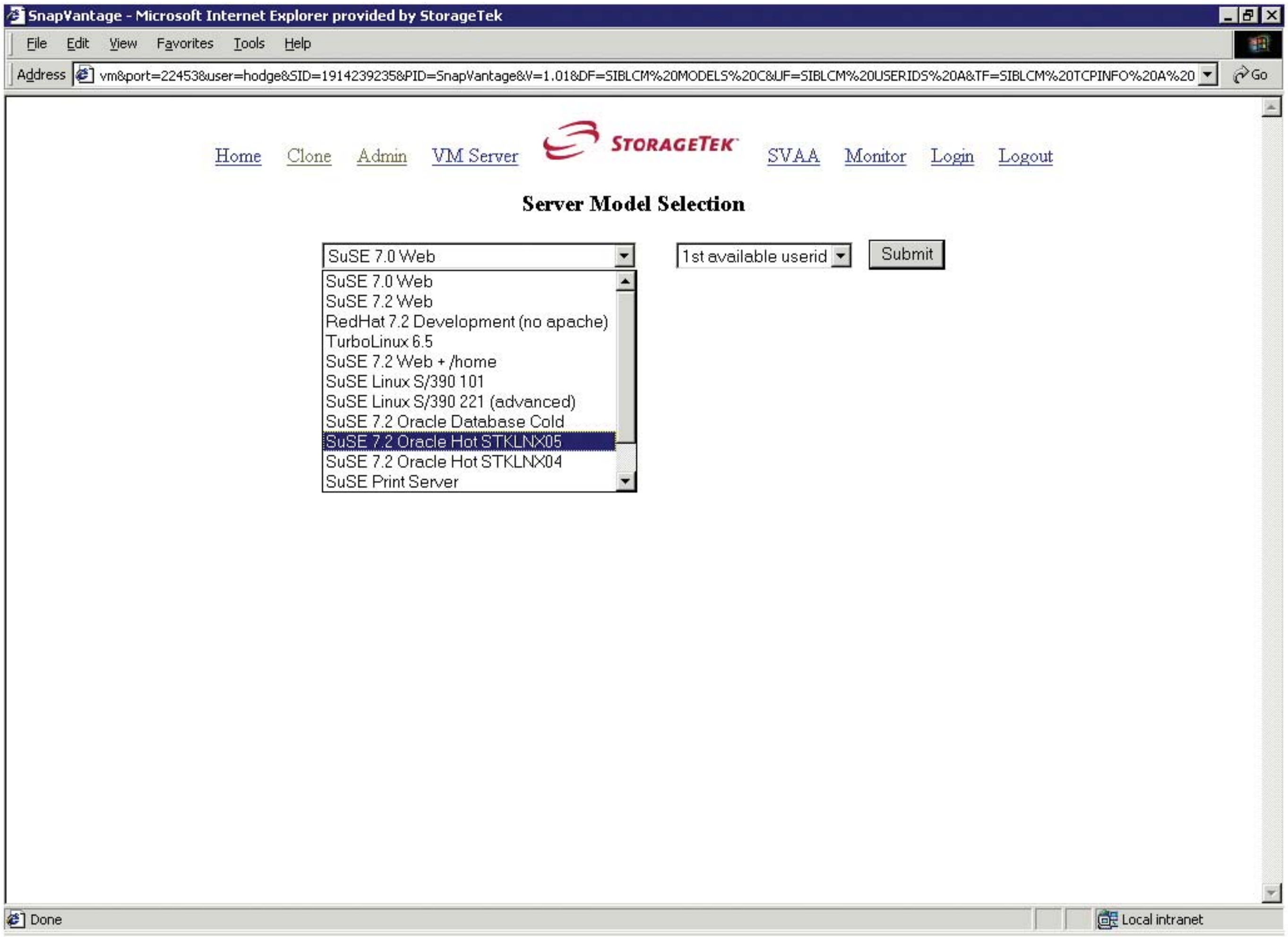
The Clone selection displays the Server Model Selection page, which is used for defining and deploying a new virtual server. From the drop-down list, select the Server Model to be used in this virtual server deployment. The second drop-down list displays the specified Linux userid.

2.2 Oracle installation clone

Once an Oracle installation or upgrade is completed on one Linux server, this entire Linux/Oracle environment can be cloned for future replication from the existing SnapVantage software clone menu. Development or test environments may be the first to receive the upgrade, followed by user acceptance test or pre-production environments. These environments may coexist on the same VM logical partition, thereby allowing shared resources across all of the clones. In the past, individual servers were allocated for each environment requiring separate resources and additional support effort. Now, customers are offered opportunities to leverage their investments and have more flexibility with workload distributions in non-production environments.

2.3 Oracle database clone

The database can be Snapped separately from the Linux and Oracle software installations by segregating the database files to separate devices (VM minidisks). The database clone can become the model



vm&port=22453&user=hodge&SID=1914239235&PID=SnapVantage&V=1.01&DF=SIBLCM%20MODELS%20C&UF=SIBLCM%20USERIDS%20A&TF=SIBLCM%20TCPINFO%20A%20

Server Model Selection

SuSE 7.0 Web

1st available userid

SuSE 7.0 Web
SuSE 7.2 Web
RedHat 7.2 Development (no apache)
TurboLinux 6.5
SuSE 7.2 Web + /home
SuSE Linux S/390 101
SuSE Linux S/390 221 (advanced)
SuSE 7.2 Oracle Database Cold
SuSE 7.2 Oracle Hot STKLNX05
SuSE 7.2 Oracle Hot STKLNX04
SuSE Print Server

Figure 2b. Server model selection

for future clones so that the production or source database is not Snapped again. The first clone would become the model after recovery is applied to the database, and then the database and operating system are shut down normally allowing cold clones to be created.

If needed, many clones can be created in seconds from the model which is an exact point-in-time copy of the production database and operating system environment. These clones will be based upon a cold database and cold Linux environment. The cold cloning process is even faster than hot cloning, because the file system check is no longer required since the source clone, which was Snapped, was quiesced.

Production database datafile clones can also be used for media recovery purposes. If the cloned datafiles are intended for backup and recovery, then instead of starting the cloned database, leave the files untouched and available for use in the event of a media failure. Due to the SVA's redundancy and fault tolerant technology, media failures are a very low concern. However, human error during processes such as database reorganizations may still occur and may cause the same effect.

2.4 Student Oracle environment replication

Since student environments are disposable, cloning these environments is an ideal application for SnapVantage software. In addition to the Linux/Oracle clone, the database may also be cloned for each student. At the end of the course, the lab assistant may simply delete the old student environments and recreate new student environments within minutes. Since most coursework is completed within a predictable timeframe, the expiration date in the model parameters of the server can be set before it is initially cloned. In this case, the expired clones will automatically be deleted on a pre-set date and all resources will be returned to their respective pools for new server or database deployments.

2.5 Development environments

IT organizations often include developers who support different aspects of the same database application. When the application is under development, there are occasions when unit and integration testing are conducted by the developer. If there are multiple developers working on different aspects of the application, integration test activities will require coordination between the developers. With SnapVantage software, each developer can work from their own development environment and a private database clone without having to be concerned about impacting any other developers. In addition, the whole development team can also have a separate integration environment by utilizing the cloning capability.

2.6 Test and pre-production environments

In addition to the development integration test environment, it is common to have a user acceptance test environment or a pre-production environment. These environments have a cyclical workload and therefore usually don't warrant a separate server; however, the additional resources are often justified for separate test environments. Due to the cyclical nature of these systems, it would be a good use of resources for them to coexist with the development environments sharing an SVA.

2.7 Reporting environments

It is often desirable to separate the reporting access to a separate database for the use of tools such as BrioQuery. Most production data warehouse systems implement stringent restrictions on database resources for use of the production database to prevent any impact to performance. Consequently, the data mining function is conducted on a copy of the production data warehouse, typically on a separate server.

By using SnapVantage software to clone the production environment on the same SVA, redundant storage is not consumed by the cloned database. Since separate server resources are allocated for the clone, the reporting activities are isolated from the production database, thereby preventing any performance impact.

The DBA can identify a short window of time to create a clone of the production database and to replicate a reporting environment. Since database naming and TCPIP/SQLnet settings would require tailoring, a few commands must be issued before the new database can be accessible.

Reporting environments are usually considered mission-critical to the business and may need to be refreshed on a daily basis in order to provide the most up-to-date data for reporting needs. Due to the speed of the cloning process, these databases can be refreshed more frequently without impact to the user community.

2.8 Non-production environment refreshes

Many information systems shops conduct a monthly or quarterly database refresh process for their non-production environments. These database copies may be required for the purpose of making current data available for testing efforts or for making available the latest changes to schemas or procedures that are stored in the database.

The production database is the source of a clone for creating a model to replicate additional database systems, as already discussed. This process is conducted by creating a hot or cold clone of the source database through automated scripts using SnapVantage software. Additional tailoring of the resulting cloned environment is required to rename the database as discussed in section 3.4 of this paper.

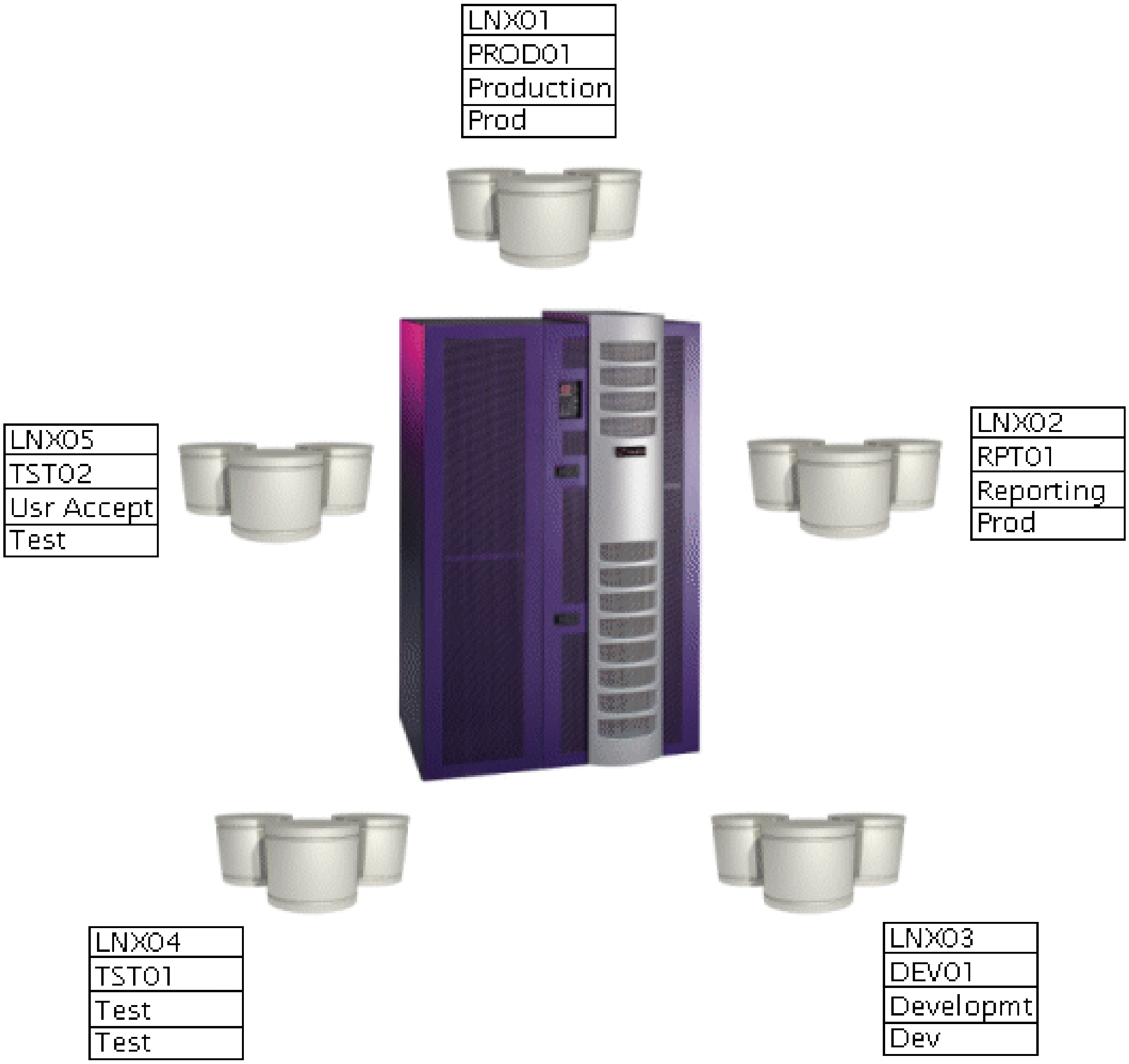


Figure 2c. Production environment clones

There are also information systems shops that may decide to have the development environment become the source of a clone process for test environments for validating changes to the database application. Illustrated below are examples of different test environments for test organizations such as Early Validation Test, Integration Test, Solutions Test and User Acceptance Test. As bugs are found and the development environment is corrected, many iterations of the cloning process can be conducted to keep the test environments as current as possible to maximize test effectiveness.

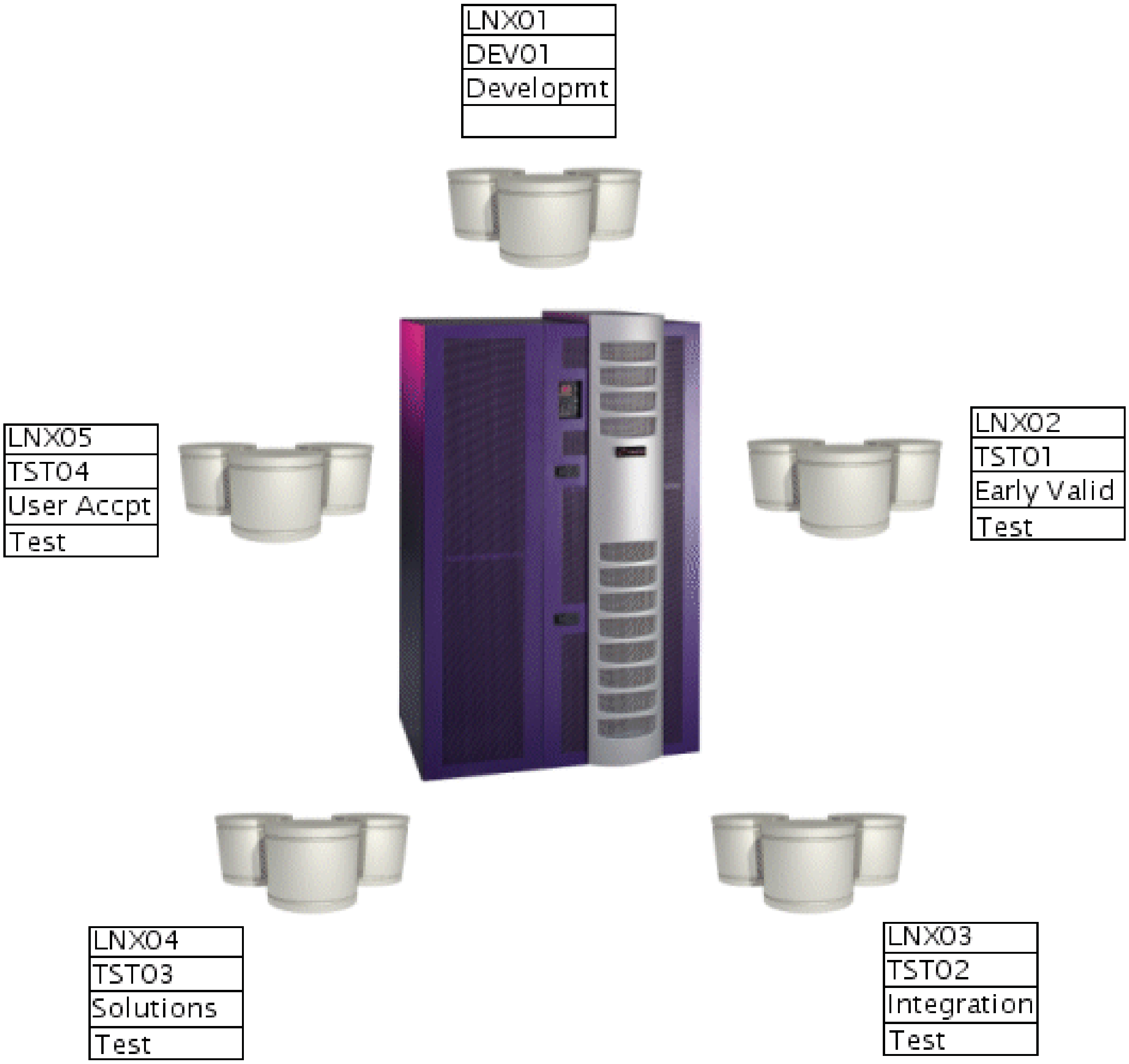


Figure 2d. Non-production environment refreshes

3 SnapVantage software cloning process

The cloning process referenced in this paper outlines the manual steps necessary to clone the server, the Oracle software and the Oracle database. The Oracle 9i DBA Handbook illustrates the three-part process for a database hot backup:

1. A tablespace-by-tablespace backup of the datafiles, which in turn consists of
   a. Setting the table space into backup state
   b. Backing up the table spaces datafiles
   c. Restoring the table space to its normal state
2. Backup of the archived redo log files, which consists of
   a. Recording which files are in the archived redo log destination directory
   b. Backing up the archived redo log files, then (optionally) deleting or compressing them
3. Backup of the control file via the alter database backup control file command.

The following process can be used with SnapVantage software and Oracle Sqlplus database access and is detailed in section 3.2 through 3.4 of this paper. A tablespace-by-tablespace backup of the datafiles is no longer required. Step #1 of the SnapVantage Software SnapShot Clone Creation process replaces all three steps described in the hot backup process outlined above.

1. SnapVantage Software SnapShot Clone Creation
   a. Backup control file to generate control file creation script
   b. Set all of the table spaces into backup state
   c. Clone Environment using SnapVantage software
   d. Restore the table spaces to their normal state
2. Recovery of Non-Production Clone
   e. Restore the table spaces to their normal state
   f. Optional Archive Log Recovery of Non-Production Clone
3. Cloned Database Rename
   g. Modify/Invoke supporting files, e.g., .profile ORACLE_SID value
   h. Tailor/Invoke create control file script

The Hot Backup section in *Oracle 7x24 Tips and Techniques* cites an example where fifteen hours are required to backup a 250 gigabytes database. By using the SnapVantage software Cloning Process, the backup and recovery can be completed in less than one hour. In our example, the size of the database, software and operating system Snapped is approximately 20 gigabytes. The cloning process and operating system startup are conducted during this process, which took roughly two minutes.

3.1 Linux server setup

After the initial Linux server environment is established manually through VM, additional Linux servers can be created during the cloning process. If the Linux server already exists and the environment needs to be refreshed from a model system, the existing server will need to be logged off of VM, then deleted using the SnapVantage software Admin panel.

3.2 Pre-clone Oracle database preparation

When cloning an existing database, it is important to generate a control file creation script for use in establishing control files for the newly cloned database. This script will indicate all of the datafiles and their locations in the source model environment. Since the clone’s file system structure will be exactly the same as the source, these datafile location descriptions are directly transferable to the clone. Generate the control file creation script by backing up the control file to the ccf.sql filename into a path where it can easily be found on the new clone.

Use the following commands from sqlplus to generate the script:

```
SQL> connect/as sysdba
Connected.
SQL> alter database backup controlfile to trace as
'/oracle/admin/stkdb/arch/ccf.sql' reuse resetlogs;
Database altered.
```

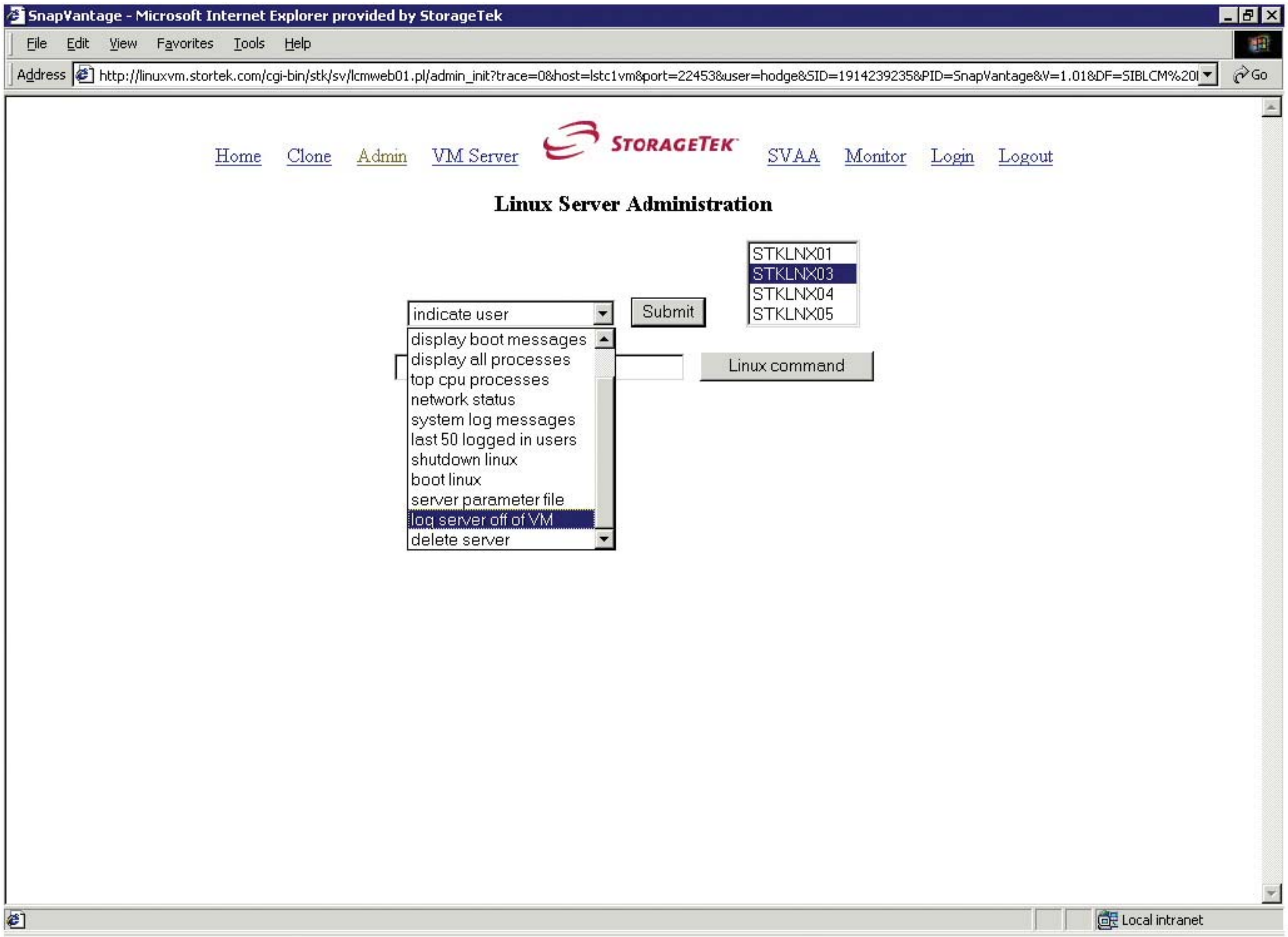
The create controlfile script will be automatically copied to the target system during the clone process. The contents of the ccf.sql file must be modified to indicate the cloned database name and set command, which will cause the database name change.

Edit the "ccf.sql" file:

```
FROM: CREATE CONTROLFILE REUSE DATABASE "olddbname"
RESETLOGS ...
TO: CREATE CONTROLFILE REUSE set DATABASE "newdbname"
RESETLOGS ...
```

This syntax will allow the existing controlfiles to be overwritten without giving an error. The `ARCHIVELOG` parameter can also be changed to `NOARCHIVELOG` since archive logging is not usually conducted in non-production environments. Remove the comments and commands following the `CREATE CONTROLFILE` statement since further recovery will not be necessary for the non-production clone.

Since datafiles are added dynamically throughout the life of the database, the script to alter the tablespaces to backup mode will need to be created at the time of the backup process to ensure all tablespaces and associated datafiles are included in the backup set.



Address http://linuxvm.stortek.com/cgi-bin/stk/sv/lcmweb01.pl/admin_init?trace=0&host=lstc1vm&port=22453&user=hodge&SID=1914239235&PID=SnapVantage&V=1.01&DF=SIBLCM%20

Figure 3a. Linux server administration

The following script generates the alter tablespace commands needed to set the tablespaces to backup mode. The resulting script, spooled to alter_begin.sql, will switch the tablespaces into backup mode one at a time.

```
REM gen_alter_backup.sql
REM
set pagesize 0 feedback off

select
 'alter tablespace '||tablespace_name||' begin
backup;'
from dba_tablespaces
where status <> 'INVALID'

spool alter_begin.sql
/
spool off
```

Following are the results of the gen_alter_backup.sql script in spooled file alter_begin.sql:

```
REM alter_begin.sql
REM
alter tablespace SYSTEM begin backup;
alter tablespace TOOLS begin backup;
alter tablespace RBS begin backup;
alter tablespace TEMP begin backup;
alter tablespace USERS begin backup;
alter tablespace STK_1998 begin backup;
alter tablespace STK_1999 begin backup;
alter tablespace STK_2000 begin backup;
alter tablespace STK_2001 begin backup;
alter tablespace STK_2002 begin backup;
```

The following script generates the end backup commands to be invoked after the clone is created.

```
REM gen_alter_backup_end.sql
REM
set pagesize 0 feedback off

select
 'alter tablespace '||tablespace_name||' end backup;'
from dba_tablespaces
where status <> 'INVALID'

spool alter_end.sql
/
spool off
```

Following are the results of the gen_alter_backup_end.sql script in the spooled file alter_end.sql. These commands will need to be run against the source database after the clone is created and also against the cloned database since the cloned database state will still be in backup mode upon creation.

```
REM alter_end.sql
alter tablespace SYSTEM end backup;
alter tablespace TOOLS end backup;
alter tablespace RBS end backup;
alter tablespace TEMP end backup;
alter tablespace USERS end backup;
alter tablespace STK_1998 end backup;
alter tablespace STK_1999 end backup;
alter tablespace STK_2000 end backup;
alter tablespace STK_2001 end backup;
alter tablespace STK_2002 end backup;
```

3.3 SnapVantage software clone of production environment

After the alter_begin.sql script is complete and all tablespaces are safely switched to backup mode, invoke the Clone option from SnapVantage software to clone the entire environment including the Linux server, Oracle software and Oracle database. The userid can be selected from the pull-down list, or the first available userid can be used to create the clone account.

In this example, three Oracle Database models are presented on the selection list. A model is available for a cold source environment, meaning all processes are shut down when the model was cloned. There are also two hot source environments, meaning that the server and database processes are up and running on the source model.

After the server model selection is picked, the Model Parameters page is displayed. The parameters are associated with the source model and defined on the VM server account as needed to build the new clone. This example includes the swap disk, the Linux boot disk, the Oracle software installation disk and the Oracle database disk.

3.3.1 Cloning log

After the model parameters are chosen, the submit button starts the cloning process resulting in the following online log displayed from SnapVantage software. The alter_end.sql script can be invoked as soon as the SIBLCMBL: SnapShot messages are complete for every volume specified in the model.

```
DMSACP723I A (191) R/O
DMSACP723I D (192) R/O
DMSACP723I E (194) R/O
Ready;
10:25:58 SIBLCMBL: Linux boot process for STKLNX04 on
15 Oct 2002
10:25:58 SIBLCMBL: S : DONJ : 1200 : DJPOOL : 0150 :
200 : 3390 : swap
10:25:58 SIBLCMBL: CP LINK * 0150 0150 M
10:25:58 SIBLCMBL: CP LINK DONJ 1200 AS 1FFF RR
10:25:58 SIBLCMBL: SnapShot took 0.343 seconds to
copy 200 cylinders
10:25:58 SIBLCMBL: S : STKLNX05 : 1151 : DJPOOL :
0151 : 3000 : 3390 : boot
10:25:58 SIBLCMBL: CP LINK * 0151 0151 M
10:25:58 SIBLCMBL: CP LINK STKLNX05 1151 AS 1FFF RR
10:25:58 SIBLCM93: DASD 1FFF LINKED R/O; R/W BY
STKLNX05
10:25:59 SIBLCMBL: SnapShot took 0.384 seconds to
copy 3000 cylinders
10:25:59 SIBLCMBL: S : STKLNX05 : 0152 : DJPOOL :
0152 : 5000 : 3390 : oracle
10:25:59 SIBLCMBL: CP LINK * 0152 0152 M
10:25:59 SIBLCMBL: CP LINK STKLNX05 0152 AS 1FFF RR
10:25:59 SIBLCM93: DASD 1FFF LINKED R/O; R/W BY
STKLNX05
10:26:00 SIBLCMBL: SnapShot took 0.867 seconds to
copy 5000 cylinders
10:26:00 SIBLCMBL: S : STKLNX05 : 0153 : DJPOOL :
0153 : 7000 : 3390 : db1
10:26:00 SIBLCMBL: CP LINK * 0153 0153 M
10:26:00 SIBLCMBL: CP LINK STKLNX05 0153 AS 1FFF RR
```

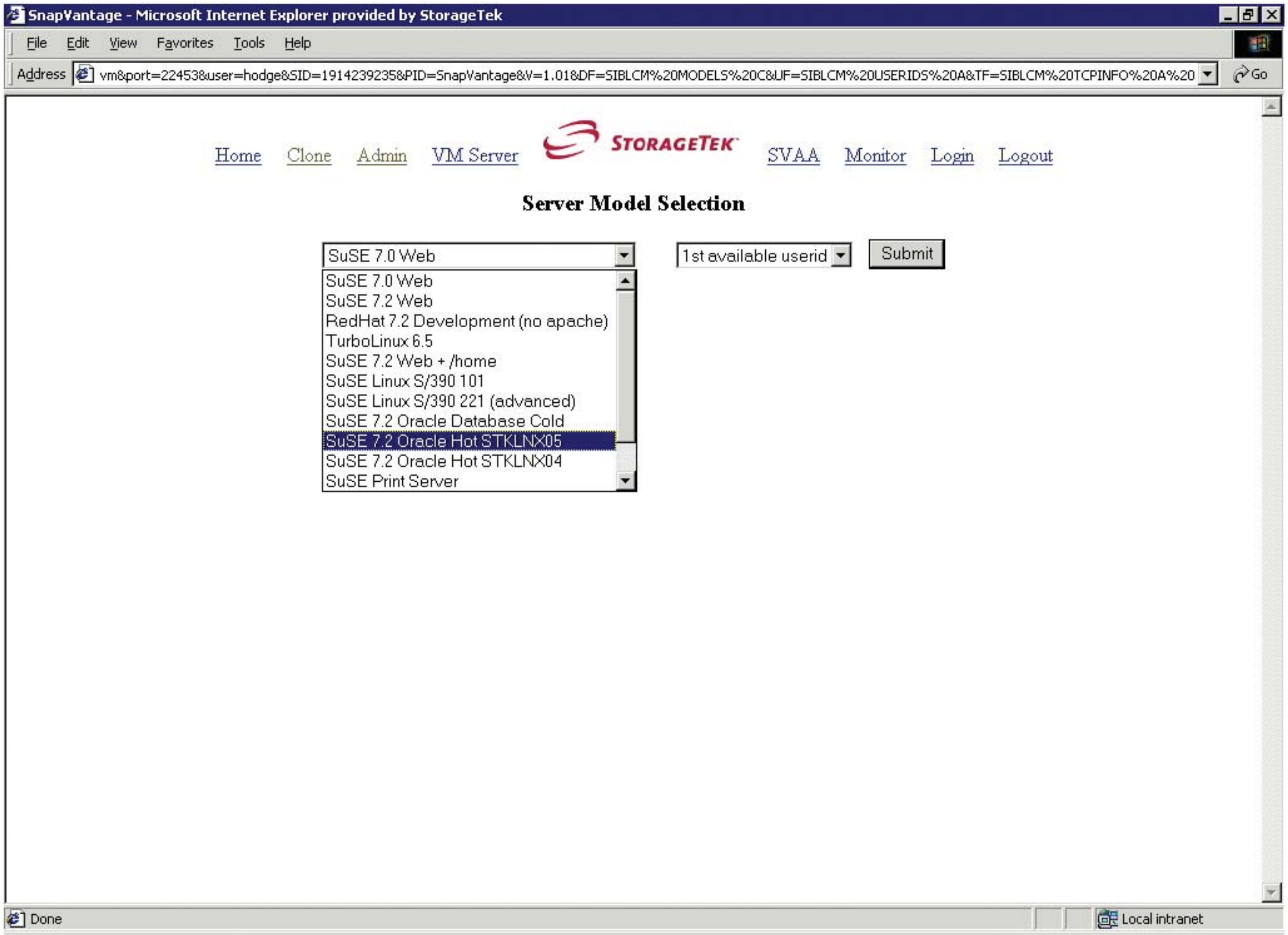
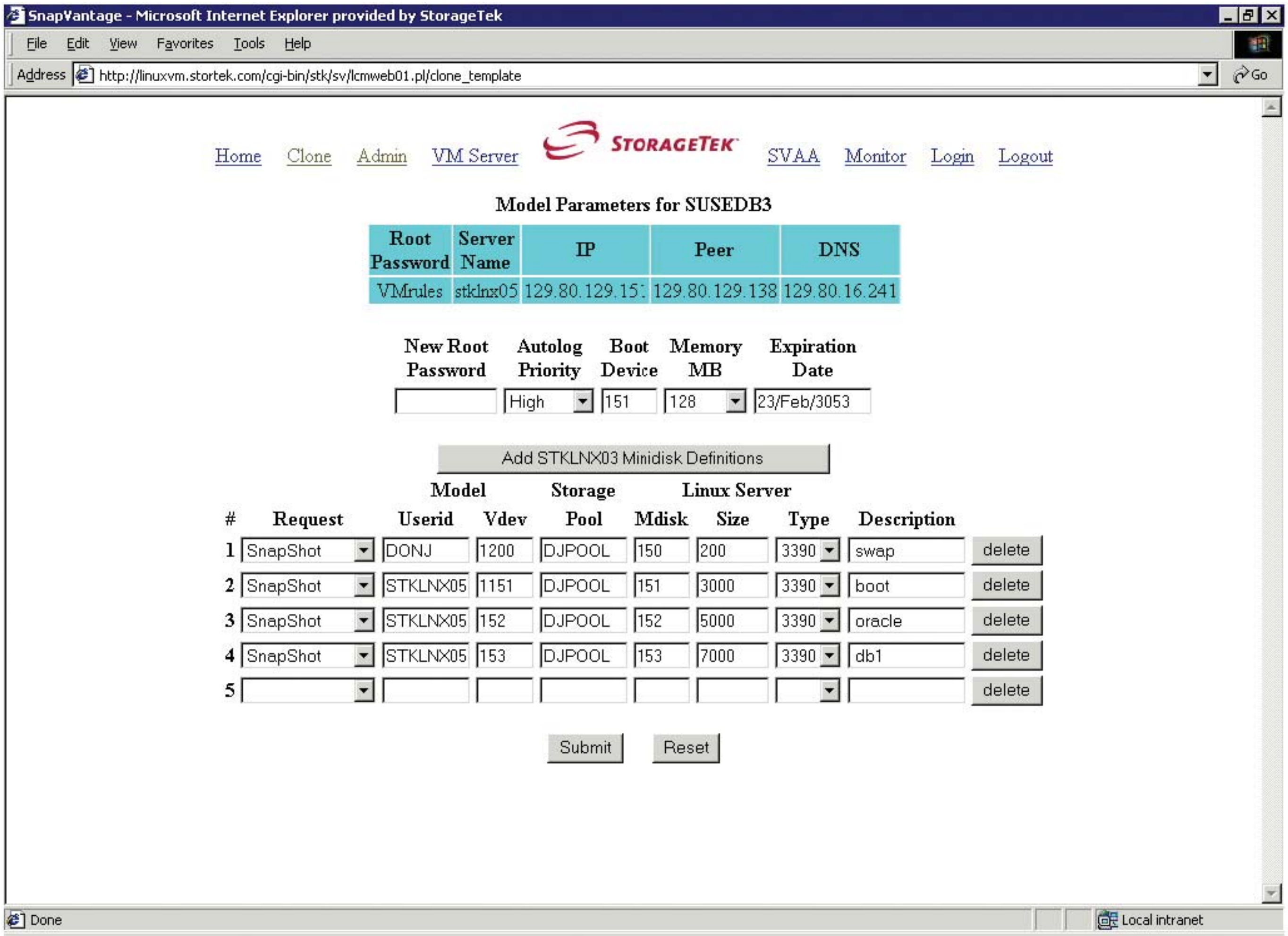

SnapVantage - Microsoft Internet Explorer provided by StorageTek
File Edit View Favorites Tools Help
Address vm&port=22453&user=hodge&SID=1914239235&PID=SnapVantage&V=1.01&DF=SIBLCM%20MODELS%20C&UF=SIBLCM%20USERIDS%20A&TF=SIBLCM%20TCPINFO%20A%20
Go
Home Clone Admin VM Server STORAGETEK SVAA Monitor Login Logout
Server Model Selection
SuSE 7.0 Web
1st available userid
Submit
SuSE 7.0 Web
SuSE 7.2 Web
RedHat 7.2 Development (no apache)
TurboLinux 6.5
SuSE 7.2 Web + /home
SuSE Linux S/390 101
SuSE Linux S/390 221 (advanced)
SuSE 7.2 Oracle Database Cold
SuSE 7.2 Oracle Hot STKLNX05
SuSE 7.2 Oracle Hot STKLNX04
SuSE Print Server
Done
Local intranet

Figure 3b. Server model selection

```
10:26:00 SIBLCM93: DASD 1FFF LINKED R/O; R/W BY
STKLNX05
10:26:00 SIBLCMBL: SnapShot took 0.639 seconds to copy
7000 cylinders
10:26:00 SIBLCM78: CP DEFINE CTCA 0EB0 USER TCPIP
10:26:00 SIBLCM78: CP DEFINE CTCA 0EB1 USER TCPIP
10:26:00 SIBLCM78: CP COUPLE 0EB0 TCPIP 0F74
10:26:00 SIBLCM78: CP COUPLE 0EB1 TCPIP 0F75
10:26:00 SIBLCM78: Virtual CTCs coupled with TCPIP
10:26:00 SIBLCMBL: CP QUERY VIRTUAL STORAGE
10:26:00 SIBLCMBL: STORAGE = 128M
10:26:00 SIBLCMBL: CP IPL 151 CLEAR
hwc low level driver: can write messages
hwc low level driver: can not read state change noti-
fications
hwc low level driver: can read commands
hwc low level driver: can read priority commands
Linux version 2.4.7-SuSE-SMP (root@k_deflt.suse.de)
(gcc version 2.95.3 20010315 (SuSE)) #1 SMP Wed Oct 17
15:31:03 GMT 2001
We are running under VM
This machine has an IEEE fpu
On node 0 totalpages: 32768
zone(0): 32768 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line:
dasd=0150,0151,0152,0153,0154,0155,0156
root=/dev/dasdb1 noinitrd iucv=tcpip
Highest subchannel number detected (hex) : 0011
Calibrating delay loop...
517.73 BogoMIPS
```


SnapVantage - Microsoft Internet Explorer provided by StorageTek
File Edit View Favorites Tools Help
Address
Go
Home Clone Admin VM Server
STORAGETEK
SVAA Monitor Login Logout
Done
Local intranet

http://linuxvm.stortek.com/cgi-bin/stk/sv/lcmweb01.pl/clone_template

Model Parameters for SUSEDB3

Root Password	Server Name	IP	Peer	DNS
VMrules	stklnx05	129.80.129.15	129.80.129.138	129.80.16.241

New Root Password	Autolog Priority	Boot Device	Memory MB	Expiration Date
	High	151	128	23/Feb/3053

Add STKLNX03 Minidisk Definitions

		Model		Storage	Linux Server				
#	Request	Userid	Vdev	Pool	Mdisk	Size	Type	Description	
1	SnapShot	DONJ	1200	DJPOOL	150	200	3390	swap	delete
2	SnapShot	STKLNX05	1151	DJPOOL	151	3000	3390	boot	delete
3	SnapShot	STKLNX05	152	DJPOOL	152	5000	3390	oracle	delete
4	SnapShot	STKLNX05	153	DJPOOL	153	7000	3390	db1	delete
5									delete

Submit Reset

Figure 3c. Model parameters

```
Memory: 125316k/131072k available (2017k kernel code,
0k reserved, 622k data, 52k init)
Dentry-cache hash table entries: 16384 (order: 5,
131072 bytes)
Inode-cache hash table entries: 8192 (order: 4, 65536
bytes)
Mount-cache hash table entries: 2048 (order: 2, 16384
bytes)
Buffer-cache hash table entries: 8192 (order: 3,
32768 bytes)
Page-cache hash table entries: 32768 (order: 5,
131072 bytes)
debug: Initialization complete
POSIX conformance testing by UNIFIX
Detected 1 CPU's
Boot cpu address 0
cpu 0 phys_idx=0 vers=FF ident=0C0F69 machine=2064
unused=0000
init_mach : starting machine check handler
init_mach : machine check buffer : head = 00266404
init_mach : machine check buffer : tail = 00266408
init_mach : machine check buffer : free = 0026640C
init_mach : CRW entry buffer anchor = 00266410
init_mach : machine check handler ready
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society
NET3.039
Initializing RT netlink socket
mach_handler : ready
mach_handler : waiting for wakeup
Starting kswapd v1.8
VFS: Diskquotas version dquot_6.4.0 initialized
pty: 256 Unix98 ptys configured
block: queued sectors max/low 83069kB/27689kB, 256
slots per queue
RAMDISK driver initialized: 16 RAM disks of 32768K
size 1024 blocksize
dasd:initializing...
dasd:Registered successfully to major no 94
dasd(eckd):ECKD discipline initializing
dasd(eckd):0150 on sch 4: 3390/0C(CU:3990/04) Cyl:200
Head:15 Sec:224
dasd(eckd):0150 on sch 4: 3390/0C(CU:3990/04):
Configuration data read
SPID - Device 0150 on Subchannel 0004 became 'not
operational'
dasd(eckd):0151 on sch 5: 3390/0C(CU:3990/04) Cyl:3000
Head:15 Sec:224
dasd(eckd):/dev/dasda(94:0),0150@0x4:(4kB blks):
144000kB at 48kB/trk classic disk layout
dasd(eckd):0151 on sch 5: 3390/0C(CU:3990/04):
Configuration data read
SPID - Device 0151 on Subchannel 0005 became 'not
operational'
dasd(eckd):0152 on sch 6: 3390/0C(CU:3990/04) Cyl:5000
Head:15 Sec:224
dasd(eckd):0152 on sch 6: 3390/0C(CU:3990/04):
Configuration data read
SPID - Device 0152 on Subchannel 0006 became 'not
operational'
dasd(eckd):0153 on sch 7: 3390/0C(CU:3990/04) Cyl:7000
Head:15 Sec:224
dasd(eckd):0153 on sch 7: 3390/0C(CU:3990/04):
Configuration data read
SPID - Device 0153 on Subchannel 0007 became 'not
operational'
dasd:waiting for responses...
dasd(eckd):/dev/dasdb(94:4),0151@0x5:(4kB blks):
2160000kB at 48kB/trk classic disk layout
dasd(eckd):/dev/dasdc(94:8),0152@0x6:(4kB blks):
3600000kB at 48kB/trk classic disk layout
dasd(eckd):/dev/dasdd(94:12),0153@0x7:(4kB blks):
5040000kB at 48kB/trk classic disk layout
Partition check:
 dasda:(nonl)/    : dasda dasda1
 dasdb:LNX1/ 0X0151: dasdb dasdb1
 dasdc:(nonl)/    : dasdc dasdc1
 dasdd:(nonl)/    : dasdd dasdd1
dasd(eckd):We are interested in: CU 3880/00
dasd(eckd):We are interested in: CU 3990/00
dasd(eckd):We are interested in: CU 2105/00
dasd(eckd):We are interested in: CU 9343/00
dasd:Registered ECKD discipline successfully
dasd(fba):FBA discipline initializing
dasd(fba):We are interested in: Dev 9336/00 @ CU
6310/00
dasd(fba):We are interested in: Dev 3370/00 @ CU
3880/00
dasd:Registered FBA discipline successfully
dasd:initialization finished
loop: loaded (max 8 devices)
debug: cio_msg: new level 2
debug: cio_trace: new level 2
debug: cio_crw: new level 6
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 8Kbytes
TCP: Hash tables configured (established 4096 bind
8192)
Linux IP multicast router 0.06 plus PIM-SM
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 13k freed
INIT: version 2.78 booting
Running /etc/init.d/boot
Mounting /proc device..done
Mounting /dev/pts..done
blogd: boot logging disabled
Activating swap-devices in /etc/fstab...
Adding Swap: 143980k swap-space (priority -1)
..done
Checking file systems...
Parallelizing fsck version 1.19a (13-Jul-2000)
/dev/dasdb1: clean, 54455/557056 files, 362188/539997
blocks
/dev/dasdd1 was not cleanly unmounted, check forced.
/dev/dasdc1 was not cleanly unmounted, check forced.
10:26:18 Waiting for messages from STKLNX04
/dev/dasdc1: Inode 177070, i_blocks is 16, should be
8. FIXED.
/dev/dasdc1: Inode 177071, i_blocks is 56, should be
8. FIXED.
```

```
/dev/dasdc1: Inode 321929, i_blocks is 75912, should
be 102504. FIXED.
10:26:48 Waiting for messages from STKLNX04
/dev/dasdc1: 81/450688 files (21.0% non-contiguous),
844420/899997 blocks
/dev/dasdd1: Inode 372300, i_blocks is 64, should be
8. FIXED.
/dev/dasdd1: Inode 372305, i_blocks is 64, should be
8. FIXED.
10:27:18 Waiting for messages from STKLNX04
10:27:38 Waiting for messages from STKLNX04
/dev/dasdd1: 22548/630240 files (0.7% non-contiguous),
762187/1259997 blocks
..done
Mounting local file systems...
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/dasdc1 on /oracle type ext2 (rw)
/dev/dasdd1 on /db1 type ext2 (rw)
..done
Activating remaining swap-devices in /etc/fstab...
..done
Setting up timezone data
..done
Setting up loopback device..done
Setting up hostname..done
Mount SHM FS on /dev/shm..done
Running /etc/init.d/boot.local
..done
Creating /var/log/boot.msg
..doneEnabling syn flood protection..done
Disabling IP forwarding..done
INIT: Entering runlevel: 1
blogd: boot logging disabled
Master Resource Control: previous runlevel: N,
switching to runlevel: 1
Sending all processes the TERM signal...
..done
Sending all processes the KILL signal...
..done
INIT: Going single user
INIT: Sending processes the TERM signal
Give root password to login:
#*#:
/root/stk/chgident.sh stklnx05 stklnx04 129.80.129.151
129.80.129.153 12
<stklnx05 stklnx04 129.80.129.151 129.80.129.153 129
129.80.129.138 129.80.129.13
<29.151 129.80.129.153 129.80.129.138 129.80.129.138
129.80.16.241 129.80.16.
<80.129.138 129.80.129.138 129.80.16.241 129.80.16.241
Old Hostname = stklnx05
New Hostname = stklnx04
Old Host IP = 129.80.129.151
New Host IP = 129.80.129.153
Old Peer IP = 129.80.129.138
New Peer IP = 129.80.129.138
Old DNS IP = 129.80.16.241
New DNS IP = 129.80.16.241
Distribution = suse
SED CMD = /bin/sed -e
's/129.80.129.151/129.80.129.153/g' -e
's/stklnx05/stklnx04/g' -e
's/129.80.129.138/129.80.129.138/g' -e
's/129.80.16.241/129.80.16.241/g'
#*#:
perl /root/stk/init_check.pl
INIT: Switching to runlevel: 3
INIT: Sending processes the TERM signal
#*#:

STKLNX04 initialized with IP address 129.80.129.153
and root password VMrules
SUSE72 clone operation completed in 2 minutes 5 seconds
```

3.4 Oracle clone recovery

Upon completion of the virtual cloning process, log in to the cloned server and start up the database to allow automatic recovery to occur. First, startup mount the database so that the alter_end.sql script can be invoked to end backup mode on the tablespaces.

```
SQL> startup mount
ORACLE instance started.

Total System Global Area 169779280 bytes
Fixed Size               450640 bytes
Variable Size            46137344 bytes
Database Buffers         122880000 bytes
Redo Buffers             311296 bytes
Database mounted.

SQL> @alter_end.sql

alter tablespace SYSTEM end backup
/
alter tablespace TOOLS end backup
/
alter tablespace RBS end backup
/
alter tablespace TEMP end backup
/
alter tablespace USERS end backup
/
alter tablespace STK_1998 end backup
/
alter tablespace STK_1998_IDX end backup
/
alter tablespace STK_1999 end backup
/
alter tablespace STK_2000 end backup
/
alter tablespace STK_2001 end backup
/
alter tablespace STK_2002 end backup
/
Tablespace altered.
```

```
SQL>
Tablespace altered.

SQL>
Tablespace altered.

SQL>
Tablespace altered.


Tablespace altered.

SQL>
Tablespace altered.

SQL>
Tablespace altered.

SQL>
Tablespace altered.

SQL>
Tablespace altered.

SQL>
Tablespace altered.

SQL>
Tablespace altered.
```

Verify the backup status of 'NOT ACTIVE' by selecting from the v$backup table before attempting to open the database.

```
SQL> describe v$backup
 Name                    Null?   Type
 ----------------------- -------- --------
 FILE#                            NUMBER
 STATUS                           VARCHAR2(18)
 CHANGE#                          NUMBER
 TIME                DATE

SQL> select status from v$backup;

STATUS
------------------
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
NOT ACTIVE
```

```
SQL> alter database open;
Database altered.
```

Oracle's automatic crash recovery occurs after the ALTER DATABASE OPEN command is issued, resulting in a successful recovery of online redo logs. See Appendix 2 for the cloned database alert log.

After startup is complete, shut down the database and start it up again to ensure database health before imposing any changes to the database. The database must be shut down with SHUTDOWN NORMAL or SHUTDOWN IMMEDIATE commands. It must not be shut down abnormally using SHUTDOWN ABORT or by using an unrecovered clone.

When using a virtual clone to create a database with a different name, be careful to modify all of the supporting files in the cloned ORACLE_HOME path in preparation for the database name change. To ensure all files referencing the source model database name are changed, use the find command to identify the files that must be changed. The most important files include the .profile, which references the ORACLE_SID; the init.ora file, which is the database initialization file specifying the DB_NAME, INSTANCE_NAME, and SERVICE_NAMES; and the tnsnames.ora and listener.ora parameter files for Oracle Net.

Modify and invoke the .profile to set the new ORACLE_SID:

```
stklnx03:/opt/oracle:stkdb$ vi .profile

export ORACLE_SID=stkdb2


stklnx03:/opt/oracle:stkdb$ . .profile

ORACLE_SID = stkdb2
ORACLE_BASE = /opt/oracle
ORACLE_HOME = /db1/oracle1/product/9.2
```

Rename the init.ora file to include the new database name and modify the database name references:

```
stklnx03:/oracle/admin/pfile:stkdb2$ mv initstkdb.ora
initstkdb2.ora
stklnx03:/oracle/admin/pfile:stkdb2$ vi initstkdb2.ora
db_name = "stkdb2"
instance_name = stkdb2


service_names = stkdb2
```

Create a new symbolic link in the dbs directory to the newly named init.ora file:

```
stklnx03:/db1/oracle1/product/9.2.0/dbs:stkdb2$
ln -s /oracle/admin/pfile/initstkdb2.ora
initstkdb2.ora
```

Following is a sample create controlfile script including the previously described changes:

```
REM ccf.sql
REM
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE SET DATABASE "STKDB2" RESET-
LOGS NOARCHIVELOG
  MAXLOGFILES 100
  MAXLOGMEMBERS 5
  MAXDATAFILES 1022
  MAXINSTANCES 1
  MAXLOGHISTORY 1134
LOGFILE
 GROUP 1 (
  '/oracle/ora03/oradata/stkdb/redo1_01.log',
  '/oracle/ora04/oradata/stkdb/redo1_02.log'
 ) SIZE 50M,
 GROUP 2 (
  '/oracle/ora03/oradata/stkdb/redo2_01.log',
  '/oracle/ora04/oradata/stkdb/redo2_02.log'
 ) SIZE 50M,
 GROUP 3 (
  '/oracle/ora03/oradata/stkdb/redo3_01.log',
  '/oracle/ora04/oradata/stkdb/redo3_02.log'
 ) SIZE 50M,
 GROUP 4 (
  '/oracle/ora03/oradata/stkdb/redo4_01.log',
  '/oracle/ora04/oradata/stkdb/redo4_02.log'
 ) SIZE 50M,
 GROUP 5 (
  '/oracle/ora03/oradata/stkdb/redo5_01.log',
  '/oracle/ora04/oradata/stkdb/redo5_02.log'
 ) SIZE 50M,
 GROUP 6 (
  '/oracle/ora03/oradata/stkdb/redo6_01.log',
  '/oracle/ora04/oradata/stkdb/redo6_02.log'
 ) SIZE 50M
DATAFILE
 '/oracle/ora02/oradata/stkdb/system01.dbf',
 '/oracle/ora02/oradata/stkdb/tools01.dbf',
 '/oracle/ora02/oradata/stkdb/rbs01.dbf',
 '/oracle/ora03/oradata/stkdb/temp01.dbf',
 '/oracle/ora03/oradata/stkdb/users01.dbf',
 '/oracle/ora02/oradata/stkdb/stk_1998_01.dbf',
 '/oracle/ora03/oradata/stkdb/stk_1999_01.dbf',
 '/db1/oracle1/oradata/stkdb/stk_1998_02.dbf',
 '/db1/oracle1/oradata/stkdb/stk_1998_idx_01.dbf',
 '/oracle/ora04/oradata/stkdb/stk_2000_01.dbf',
 '/oracle/ora05/oradata/stkdb/stk_2001_01.dbf',
 '/oracle/ora06/oradata/stkdb/stk_2002_01.dbf',
 '/db1/oracle1/oradata/stkdb/stk_1998_idx_02.dbf',
 '/db1/oracle1/oradata/stkdb/users_idx_01.dbf'
CHARACTER SET UTF8
;
```

Invoke this script from sqlplus to create the new control files for the cloned database:

```
stklnx03:/oracle/admin/stkdb/arch:stkdb2$ sqlplus
/nolog

SQL*Plus: Release 9.2.0.1.0 - Production on Tue Nov 5
09:25:09 2002

Copyright (c) 1982, 2002, Oracle Corporation.
All rights reserved.

SQL> connect/as sysdba
Connected to an idle instance.
SQL> @ccf.sql
ORACLE instance started.

Total System Global Area 169779280 bytes
Fixed Size           450640 bytes
Variable Size      46137344 bytes
Database Buffers     122880000 bytes
Redo Buffers          311296 bytes

Control file created.
```

After the new control files are successfully created, open the new database with the resetlogs option:

```
SQL> alter database open resetlogs;

Database altered.
```

The global database name should also be changed:

```
SQL> alter database rename global_name to
stkdb2.world;

Database altered.
```

The database name changes can be verified using the following select statements:

```
SQL> select instance from v$thread;

INSTANCE
----------------
stkdb2

SQL> select name from v$database;

NAME
---------
STKDB2
```

Again, to ensure database health, shut down the cloned database and start it up again. If it starts up normally, the virtual cloning process and database rename is complete!

```
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 169779280 bytes
Fixed Size           450640 bytes
Variable Size       46137344 bytes
Database Buffers     122880000 bytes
Redo Buffers          311296 bytes
Database mounted.
Database opened.
SQL>
```

4 Traditional hot backup recovery

If the cloned database is intended to be used for backup/archival, disaster recovery purposes or for creating non-production clones, the traditional hot backup procedure may be used. When recovering from a virtual clone, the archived redo log recovery process is required for rolling forward transactions.

Follow the process outlined in sections 3.2 and 3.3 before continuing with the archive log recovery of the cloned database.

4.1 Archive log preparation

Check the source system for the set of archive logs associated with the backup set in progress. Since there is a possibility that the timing of the SnapShot operation may not coincide with the archive log file write, it is recommended to FTP the archive logs to the clone system to ensure file consistency.

If a data load has completed against a data warehouse during the time-frame of the cloning process, the last archive log may be needed for the clone to include the complete set of data. Switching the logfile will force a checkpoint and generate the next archived redo log file for use in recovery of the clone.

After the successful SnapSnot operation, switch the logfile on the source database and list the archive logs to find the current log sequence number. The last redo log is the prior log in this sequence. Therefore, archive logs 37 through 42 are needed for this example.

```
Welcome to SuSE Linux 7.2 (S390) - Kernel 2.4.7-SuSE-
SMP (0).

stklnx05 login: oracle
Password:
Last login: Tue Oct 15 13:12:46 from
C115473.stortek.com
Have a lot of fun...

ORACLE_SID = stkdb
ORACLE_BASE = /opt/oracle
ORACLE_HOME = /db1/oracle1/product/9.2.0

stklnx05:/opt/oracle:stkdb$ sqlplus /nolog

SQL*Plus: Release 9.2.0.1.0 - Production on Tue Oct
15 18:55:02 2002

Copyright (c) 1982, 2002, Oracle Corporation. All
rights reserved.

SQL> connect/as sysdba
Connected.
SQL>
SQL> alter system switch logfile;

System altered.

SQL> archive log list;
Database log mode       Archive Mode
Automatic archival       Enabled
Archive destination      /oracle/admin/stkdb/arch
Oldest online log sequence   38
Next log sequence to archive  43
Current log sequence      43
```

```
SQL> exit
Disconnected from Oracle9i Enterprise Edition Release
9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining
options
JServer Release 9.2.0.1.0 - Production

stklnx05:/oracle/admin/stkdb/arch:stkdb$ ls -al
total 286960
drwxr-xr-x  2 oracle  oinstall   4096 Oct 15 18:56 ./
drwxr-xr-x  6 oracle  oinstall   4096 Oct 11 16:02
../
-rw-r-----  1 oracle  oinstall 52424704 Oct 15 16:54
arch_1_37.arc
-rw-r-----  1 oracle  oinstall 52424704 Oct 15 16:58
arch_1_38.arc
-rw-r-----  1 oracle  oinstall 52424704 Oct 15 17:02
arch_1_39.arc
-rw-r-----  1 oracle  oinstall 52424704 Oct 15 17:06
arch_1_40.arc
-rw-r-----  1 oracle  oinstall 52424704 Oct 15 17:11
arch_1_41.arc
-rw-r-----  1 oracle  oinstall 31391744 Oct 15 18:56
arch_1_42.arc
```

4.2 Archive log database recovery

FTP the archive log files from source server to target server into the cloned database archive log directory. Depending on the timing of the logfile switch and the FTP, there may be one or more archive log files generated from database transactions. These log files are used as input to the recovery process conducted on the clone.

```
Welcome to SuSE Linux 7.2 (S390) - Kernel 2.4.7-SuSE-
SMP (0).

stklnx04 login: oracle
Password:
Last login: Fri Oct 11 08:35:08 from
C115473.stortek.com
Have a lot of fun...

ORACLE_SID = stkdb
ORACLE_BASE = /opt/oracle
ORACLE_HOME = /db1/oracle1/product/9.2.0

stklnx04:/opt/oracle:stkdb$ cd
/oracle/admin/stkdb/arch
stklnx04:/oracle/admin/stkdb/arch:stkdb$ ls
stklnx04:/oracle/admin/stkdb/arch:stkdb$ ftp -i falcon
Trying 129.80.32.77...
Connected to falcon.stortek.com.
220 falcon FTP server (SunOS 5.6) ready.
Name (falcon:oracle):
331 Password required for oracle.
Password:
```

```
230 User oracle logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd arch/stkdb
250 CWD command successful.
ftp> ls
500 'EPSV': command not understood.
227 Entering Passive Mode (129,80,32,77,158,193)
150 ASCII data connection for /bin/ls
(129.80.129.153,1027) (0 bytes).
total 573764
drwxr-xr-x  2 oracle  oinstall   512 Oct 15 19:14 .
drwxr-xr-x  3 oracle  oinstall   512 Oct 15 13:42 ..
-rw-r--r--  1 oracle  oinstall 52424704 Oct 15 19:10
arch_1_37.arc
-rw-r--r--  1 oracle  oinstall 52424704 Oct 15 19:11
arch_1_38.arc
-rw-r--r--  1 oracle  oinstall 52424704 Oct 15 19:12
arch_1_39.arc
-rw-r--r--  1 oracle  oinstall 52424704 Oct 15 19:13
arch_1_40.arc
-rw-r--r--  1 oracle  oinstall 52424704 Oct 15 19:14
arch_1_41.arc
-rw-r--r--  1 oracle  oinstall 31391744 Oct 15 19:15
arch_1_42.arc
226 ASCII Transfer complete.
ftp> bin
200 Type set to I.
ftp> mget *
```

The create controlfile script is generated during the Pre-Clone preparation step when issuing the backup controlfile command. The resulting file ccf.sql is stored in the archive log directory along with the archive logs for use in the recovery process. The Resetlogs parameter was specified so that the recover process will call for archived redo logs.

```
stklnx04:/oracle/admin/stkdb/arch:stkdb$ more ccf.sql
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "STKDB" RESETLOGS
NOARCHIVELOG
  MAXLOGFILES 100
  MAXLOGMEMBERS 5
  MAXDATAFILES 1022
  MAXINSTANCES 1
  MAXLOGHISTORY 1134
LOGFILE
 GROUP 1 (
  '/oracle/ora03/oradata/stkdb/redo1_01.log',
  '/oracle/ora04/oradata/stkdb/redo1_02.log'
 ) SIZE 50M,
 GROUP 2 (
  '/oracle/ora03/oradata/stkdb/redo2_01.log',
  '/oracle/ora04/oradata/stkdb/redo2_02.log'
 ) SIZE 50M,
 GROUP 3 (
  '/oracle/ora03/oradata/stkdb/redo3_01.log',
  '/oracle/ora04/oradata/stkdb/redo3_02.log'
 ) SIZE 50M,
 GROUP 4 (
  '/oracle/ora03/oradata/stkdb/redo4_01.log',
  '/oracle/ora04/oradata/stkdb/redo4_02.log'
 ) SIZE 50M,
 GROUP 5 (
  '/oracle/ora03/oradata/stkdb/redo5_01.log',
  '/oracle/ora04/oradata/stkdb/redo5_02.log'
 ) SIZE 50M,
 ) SIZE 50M,
 GROUP 6 (
  '/oracle/ora03/oradata/stkdb/redo6_01.log',
  '/oracle/ora04/oradata/stkdb/redo6_02.log'
 ) SIZE 50M
-- STANDBY LOGFILE
DATAFILE
 '/oracle/ora02/oradata/stkdb/system01.dbf',
 '/oracle/ora02/oradata/stkdb/tools01.dbf',
 '/oracle/ora02/oradata/stkdb/rbs01.dbf',
 '/oracle/ora03/oradata/stkdb/temp01.dbf',
 '/oracle/ora03/oradata/stkdb/users01.dbf',
 '/oracle/ora02/oradata/stkdb/stk_1998_01.dbf',
 '/oracle/ora02/oradata/stkdb/stk_1998_02.dbf',
 '/oracle/ora03/oradata/stkdb/stk_1999_01.dbf',
 '/oracle/ora04/oradata/stkdb/stk_2000_01.dbf'
CHARACTER SET UTF8
;
RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL CAN-
CEL;
AUTO;
ALTER DATABASE OPEN RESETLOGS;
```

The following feedback will display as a result of the STARTUP NOMOUNT command and the create controlfile script invocation:

```
stklnx04:/oracle/admin/stkdb/arch:stkdb$ sqlplus
/nolog

SQL*Plus: Release 9.2.0.1.0 - Production on Fri Oct
18 12:07:17 2002

Copyright (c) 1982, 2002, Oracle Corporation.
All rights reserved.

SQL> connect/as sysdba
Connected to an idle instance.
SQL> @ccf.sql
ORACLE instance started.

Total System Global Area 169779280 bytes
Fixed Size           450640 bytes
Variable Size        46137344 bytes
Database Buffers     122880000 bytes
Redo Buffers          311296 bytes
Control file created.
```

The following feedback will display after the recover command and AUTO is specified as the response for automatic archive log specification.

```
SQL> recover database using backup controlfile until
cancel;
ORA-00279: change 775260 generated at 10/15/2002
16:52:33 needed for thread 1
ORA-00289: suggestion :
/oracle/admin/stkdb/arch/arch_1_37.arc
ORA-00280: change 775260 for thread 1 is in sequence
#37

Specify log: {<RET>=suggested | filename | AUTO |
CANCEL}
auto
ORA-00279: change 787022 generated at 10/15/2002
16:54:20 needed for thread 1
ORA-00289: suggestion :
/oracle/admin/stkdb/arch/arch_1_38.arc
ORA-00280: change 787022 for thread 1 is in sequence
#38
ORA-00278: log file
'/oracle/admin/stkdb/arch/arch_1_37.arc' no longer
needed for this recovery

ORA-00279: change 808011 generated at 10/15/2002
16:57:44 needed for thread 1
ORA-00289: suggestion :
/oracle/admin/stkdb/arch/arch_1_39.arc
ORA-00280: change 808011 for thread 1 is in sequence
#39
ORA-00278: log file
'/oracle/admin/stkdb/arch/arch_1_38.arc' no longer
needed for this recovery

ORA-00279: change 828975 generated at 10/15/2002
17:02:30 needed for thread 1
ORA-00289: suggestion :
/oracle/admin/stkdb/arch/arch_1_40.arc
ORA-00280: change 828975 for thread 1 is in sequence
#40
ORA-00278: log file
'/oracle/admin/stkdb/arch/arch_1_39.arc' no longer
needed for this recovery

ORA-00279: change 849857 generated at 10/15/2002
17:06:34 needed for thread 1
ORA-00289: suggestion :
/oracle/admin/stkdb/arch/arch_1_41.arc
ORA-00280: change 849857 for thread 1 is in sequence
#41
```

```
ORA-00278: log file
'/oracle/admin/stkdb/arch/arch_1_40.arc' no longer
needed for this recovery

ORA-00279: change 870861 generated at 10/15/2002
17:10:46 needed for thread 1
ORA-00289: suggestion :
/oracle/admin/stkdb/arch/arch_1_42.arc
ORA-00280: change 870861 for thread 1 is in sequence
#42
ORA-00278: log file
'/oracle/admin/stkdb/arch/arch_1_41.arc' no longer
needed for this recovery

ORA-00279: change 885292 generated at 10/15/2002
18:56:21 needed for thread 1
ORA-00289: suggestion :
/oracle/admin/stkdb/arch/arch_1_43.arc
ORA-00280: change 885292 for thread 1 is in sequence
#43
ORA-00278: log file
'/oracle/admin/stkdb/arch/arch_1_42.arc' no longer
needed for this recovery

ORA-00308: cannot open archived log
'/oracle/admin/stkdb/arch/arch_1_43.arc'
ORA-27037: unable to obtain file status
Linux Error: 2: No such file or directory
Additional information: 3

SQL> alter database open resetlogs;
Database altered.
```

The recovery process ends when it cannot find the next archive log in the archive directory. After the database is opened, it can be made available for use.

5 Hot backup efficiencies

According to an article by Venkat S. Devraj in the *Oracle 24x7 Tips and Techniques*, it is not recommended to place all tablespaces in hot backup mode simultaneously. It is also necessary to put the tablespaces into backup mode before operating system level copies are conducted against the physical datafiles since the files are open.

Mr. Devraj explains the reasons for the backup mode state of the database files: "When a tablespace is placed in hot backup mode, Oracle performs a fast checkpoint on all data-files belonging to that tablespace, thus flushing all dirty buffers belonging to that tablespace to disk. The SCN, System Change Number, which was current when the ALTER TABLESPACE… BEGIN BACKUP command was issued, is written to the data-file header as the checkpoint SCN. The checkpoint SCN is used during recovery to determine what changes need to be applied to the data-file from the redo-logs (at least the redo-logs that were created during the backup need to be applied for the recovery to be valid). Once the checkpoint SCN is written, the data-file header will not be updated with subsequent checkpoints until the ALTER TABLESPACE … END BACKUP command is issued. For each tablespace in hot backup mode, additional redo is generated due to entire block images being logged. To reduce the impact on performance due to excessive redo, hot backups should be performed sequentially on each tablespace. Thus, one tablespace may be placed in hot backup mode, the datafiles pertaining to that tablespace may be backed up via O/S commands, and then the hot backup mode for that tablespace needs to be ended, prior to placing another tablespace in hot backup mode. In other words, each tablespace should be kept in hot backup mode individually for as little time as possible. The more tablespaces are concurrently placed in hot backup mode, the higher the amount of redo generated."

These single-threaded file backup techniques were necessary with typical storage architectures; today we have eliminated the need for the tablespace-by-tablespace backup of the datafiles. From SnapVantage software, the virtual cloning process is accomplished in seconds after backup modes are switched on all tablespaces. Compared to the 15-hour single-threaded example on page 14, there is no interruption in service and no performance impact when tablespaces are in all backup mode for the clone operation. In addition, the recovery process is much simpler since the archived redo logs should reflect a low volume during the cloning operation.

Setting each tablespace to backup mode causes archived redo log files to be generated during a conventional hot backup of an Oracle database. These files all follow the backup file set to enable recovery of the database copy. Following is a timeline comparison illustrating how the traditional hot backup is done serially, generating volumes of archived redo logs, compared to the SnapVantage software Clone Creation which is accomplished at one point in time:

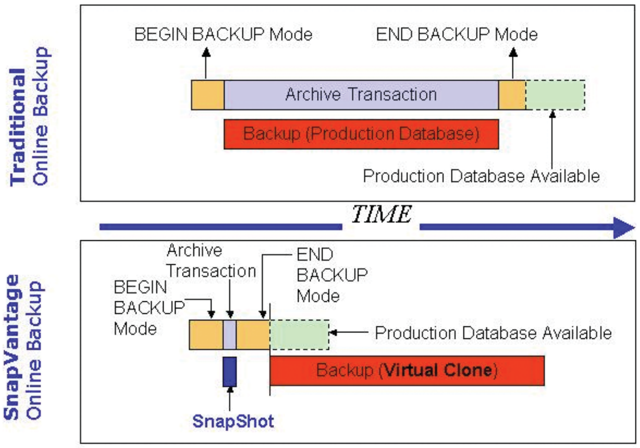


Figure 5a. Traditional Hot Backup Timeline versus SnapVantage Cloning Timeline

5.1 Media recovery scenario

The production database can be cloned at any time without disruption of service to the user community. The data written to the database at the time of the cloning operation is exactly reflected in the resulting clone, which provides a reliable base for a media recovery datafile source. If a production database datafile experiences corruption or is destroyed, the cloned datafiles can be used to recover the production database along with any required redo logs. The following media recovery scenario demonstrates the steps involved in recovering the production database after a media failure.

Identify the datafiles associated with the tablespace to be recovered. In this example, the ci_product table was loaded into the tablespace named stk_1998.

```
SQL> select file_name from dba_data_files
 2 where tablespace_name like '%1998%';

FILE_NAME
-------------------------------------------------------
/oracle/ora02/oradata/stkdb/stk_1998_01.dbf
/db1/oracle1/oradata/stkdb/stk_1998_02.dbf
/db1/oracle1/oradata/stkdb/stk_1998_idx_01.dbf
/db1/oracle1/oradata/stkdb/stk_1998_idx_02.dbf
```

For this example, a count of rows in the existing table is selected for comparison to the post recovery select count.

```
SQL> select count(*) from ci_product;

 COUNT(*)
----------
  1382153
```

The following remove and shutdown commands simulate the destroyed datafiles scenario.

```
stklnx05:/opt/oracle:stkdb$ rm
/oracle/ora02/oradata/stkdb/stk_1998_01.dbf
stklnx05:/opt/oracle:stkdb$ rm
/db1/oracle1/oradata/stkdb/stk_1998_02.dbf
stklnx05:/opt/oracle:stkdb$ sqlplus /nolog

SQL*Plus: Release 9.2.0.1.0 - Production on Fri Nov
22 10:16:26 2002

Copyright (c) 1982, 2002, Oracle Corporation. All
rights reserved.

SQL> connect/as sysdba
Connected.
SQL> shutdown
ORA-01116: error in opening database file 6
ORA-01110: data file 6:
'/oracle/ora02/oradata/stkdb/stk_1998_01.dbf'
ORA-27041: unable to open file
Linux Error: 2: No such file or directory
Additional information: 3
```

It is sometimes necessary to use the `SHUTDOWN ABORT` command to abruptly shut down the database processes after a corrupt or missing datafile is detected. The database must be shut down before the file manipulation can be done to replace the destroyed datafiles with FTP'd copies from the cloned database. It is important that these datafiles have not participated in any database recovery since they were originally cloned.

```
SQL> shutdown abort
ORACLE instance shut down.
SQL>
```

After the database is shut down, FTP the datafiles required for recovery of the production database to their original locations. Invoke FTP from the production path and connect to the cloned system. It is necessary to change directories to the production path on the cloned system and set binary mode for the file transfer.

When the cloned datafiles are in place on the production system, mount the database and conduct the recovery process. In this example, tablespace level recovery may also be applied. Since all of the redo was still in online redo logs, no archived redo log locations were prompted for access during the recovery processing.

```
stklnx05:/db1/oracle1/oradata/stkdb:stkdb$ sqlplus
/nolog

SQL*Plus: Release 9.2.0.1.0 - Production on Fri Nov 22
11:06:59 2002

Copyright (c) 1982, 2002, Oracle Corporation. All
rights reserved.

SQL> connect/as sysdba
Connected to an idle instance.
SQL> startup mount
ORACLE instance started.

Total System Global Area 169779280 bytes
Fixed Size           450640 bytes
Variable Size       46137344 bytes
Database Buffers     122880000 bytes
Redo Buffers          311296 bytes
Database mounted.
SQL> recover database;
Media recovery complete.
SQL> alter database open;

Database altered.
```

The recovered production database is now available for the user community.

6 Test results

The SnapVantage software cloning operation takes only an instant of time by copying file pointers and not the physical files themselves. Although it is not necessary to set the tablespaces to hot backup mode during the SnapVantage software cloning operation of an online database, the tablespaces are all set to backup mode at the same time to ensure data consistency. The database resulting from the clone operation is in a state as though it has experienced a server crash and only requires instance or crash recovery, which is done automatically at database startup. Automatic recovery brings the database up to the point in time when the clone operation was conducted. This capability allows a clone to be created without any interruption of service or any negative impact to performance, when the files are located on the SVA.

The SnapVantage software clone test was conducted while a two-million row table was in the process of loading data through a sqlloader routine. The database files were open and active during the cloning operation while the table load was in process. The SnapShot was invoked from SnapVantage software, roughly halfway through the table load operation, then the data load continued to completion.

After successful database startup of the clone and completion of the data load on the primary database, disk usage is analyzed from the VM SVA Administrator (SVAA) software. Initially, the disk shared between the virtual clone and the primary systems is at 100 percent. Over time, as the primary system and database both change, the shared disk diverges. This divergence can be calculated by estimating the percentage of change which may occur within a given timeframe. Based upon the rate of change to the primary database, it may be advantageous to recreate clones frequently.

Determine the rate of change by invoking the SVAA utility SIBADMIN and query the 'net capacity load shared', nclshared for each minidisk. This percentage represents the amount of the physical disk files that are shared between the primary and cloned databases.

```
sibadmin
SIB:
q mdisk(dev(153) nclshared)
Userid Vdev Volser Rdev Cyls Start Type SSname FDID
Physical-MB-%
DVTSVAA  153 DJD36A D36A 7000   1 33909 LUNAR   36A
3038.992 98
```

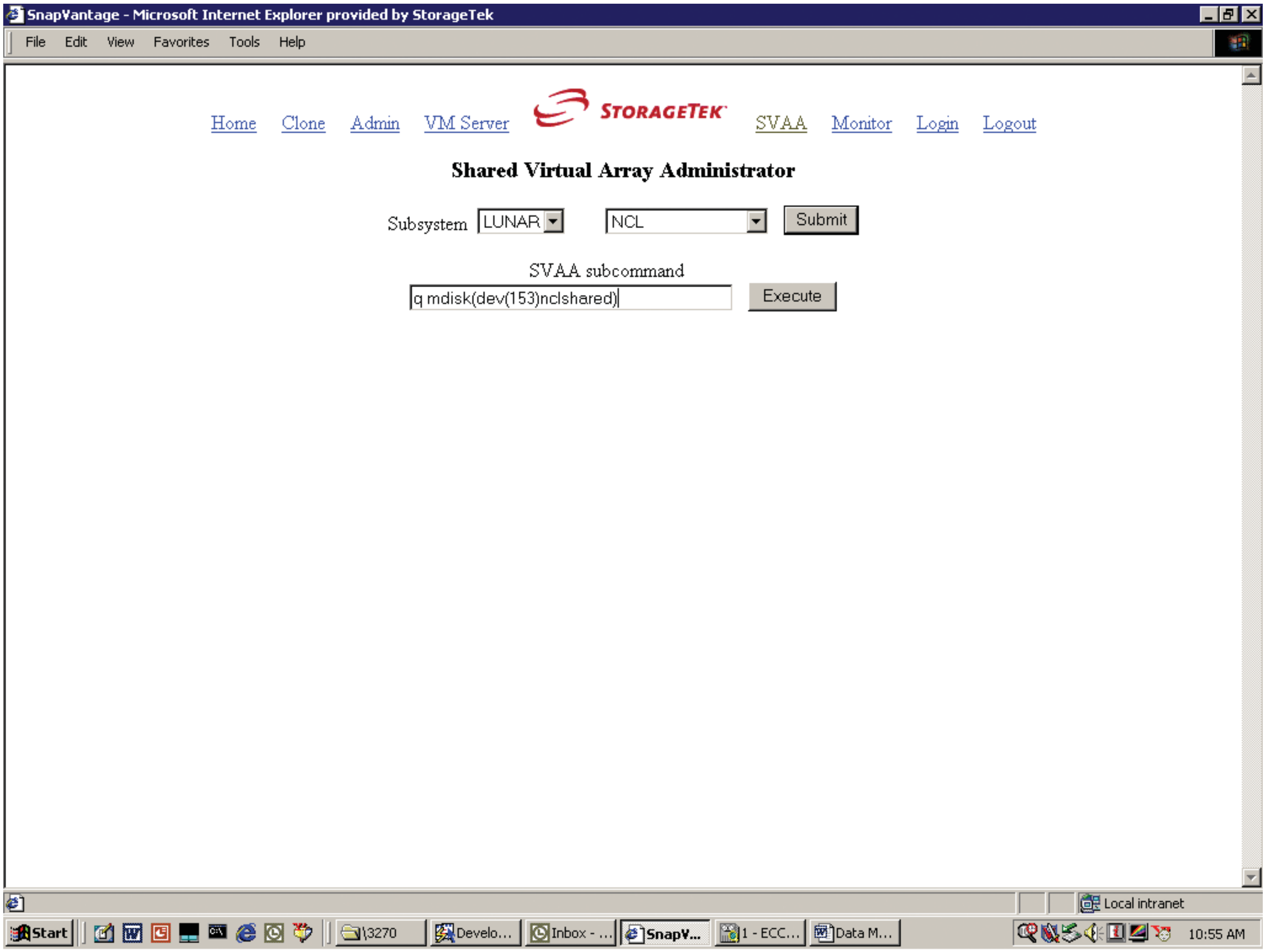
The "153" minidisk contains the Oracle software installation, which is an excellent candidate for sharing since changes to the software don't occur very often. In this case, only 60 kilobytes of storage was duplicated after creating the minidisk clone, a significant savings over conventional storage device capabilities. Although only the pointers are copied during the clone creation, enough space will need to be available on the SVA back end to accommodate the first track of each datafile, since the header records of the database files are updated as soon as the cloned database starts up.

The “152” minidisk contains the cloned database files, and the query indicates the files are 80 percent shared. In this test, the two-million row table continued to load after the clone was created, causing the primary database to diverge by 20 percent. This is because the initial size of the database is small, causing an extreme case of divergence.

```
SIB:
q mdisk(dev(152) nclshared)
Userid  Vdev Volser Rdev Cyls Start Type  SSname
FDID Physical-MB-%
DVTSVAA  152 DJD36B D36B 5000   1 33909 LUNAR   36B
643.203 80
```

The SVAA subcommand Q MDISK can also be invoked from the SnapVantage software SVAA panel selection:

The “151” minidisk contains the Linux operating system, which is indicating 99 percent shared. Typically, only 80 kilobytes of storage is actually written to the cloned Linux boot disk. Since the operating system kernel doesn’t change over time, this level of minidisk sharing should remain constant.



Shared Virtual Array Administrator

Subsystem LUNAR NCL

SVAA subcommand

q mdisk(dev(153)nclshared)

SnapVantage - Microsoft Internet Explorer provided by StorageTek
File Edit View Favorites Tools Help
Home Clone Admin VM Server
StorageTek
SVAA Monitor Login Logout
Submit
Execute
Local intranet
Start
\3270
Develo...
Inbox - ...
SnapV...
1 - ECC...
Data M...
10:55 AM

Figure 6a. Shared Virtual Array Administrator

```
SIB:
q mdisk(dev(151)nclshared)
Userid  Vdev Volser Rdev Cyls Start Type  SSname
FDID Physical-MB-%
DVTSVAA  151 DJD36C D36C 3000   1 33909 LUNAR   36C
743.729 99
```

The entire hot backup process using the virtual cloning technique takes only minutes compared to hours. Since there is no interruption in service, backups can be done several times per day, if required. The *Oracle 24x7 Tips and Techniques* scenario references a 250-gigabyte-plus database and discusses how the single-stream datafile backup was very slow, taking up to 15 hours to complete. There are many use cases for copies of databases that don't warrant the effort involved in conducting a full recovery of a traditional hot backup database.

The approach described in this application note is referred to as a Point-In-Time Image (PITI) in the document *Guidelines for Using SnapShot Storage Systems for Oracle Databases*, which accompanies the Oracle Storage Compatibility Program (OSCP) SnapShot tests. It describes the advantage of PITI on a second host by having the ability to start a second database independently of the primary database instance. A single disadvantage is described when using this method: the DBA cannot apply archive logs to the PITI since it diverges from the primary database once it is opened. When the PITI is preserved after the cloning operation and is not opened, it can be used for production backup and recovery scenarios as well.

7 Optimal Flexible Architecture (OFA) and tuning simplified with Virtual Disk

The Optimal Flexible Architecture (OFA) dictates the layout of Oracle software and database files. The OFA facilitates a consistent logical separation of database files. When applying I/O tuning considerations to the database file layout plan, a level of complexity is added which can become a time consuming task for the Database Administrator (DBA). The DBA is concerned about the I/O contention between data and index files as well as redo logs, rollback segments and temporary tablespaces.

StorageTek's SVA Raid6+ technology and the SVA channel I/O interface architecture eliminate these concerns. The logical layout of the file system can still follow the OFA, but the physical representation on the back end disk is very different. Since the Raid6+ implementation spreads data with parity across multiple physical devices, the channel-to-physical-disk relationship is no longer part of the equation. StorageTek's SVA architecture implements the Virtual device model, which is the logical host view of the mapping to physical devices on the back end. The same logical devices and paths can be referenced as following the OFA compliant file system structures.

In addition, it is also no longer necessary for the DBA to be concerned about controller relationships to the physical layout of files on disks due to the logical-to-physical mapping of the SVA arrays. The arrays utilize fault tolerant technology with 13 data disks, two parity disks and four global spares per subsystem. The separation of data and index across controllers and disks is no longer a requirement since the virtual volumes are written at a track level on the SVA backend within the arrays. Also, the V2X Shared Virtual Array (SVA) disk system provides up to 32 data channels that can be shared across multiple servers or can provide increased throughput on just a few servers.

8 Summary

StorageTek's SnapShot technology, controlled by the SnapVantage software web-based interface, enables the DBA to save a significant amount of time creating non-production database clones and daily or periodic online backups. The virtual database clone is created in a fraction of the time compared to the traditional hot backup approach. The impact to production database performance and system resources are relieved, and the savings can enhance the data processing environment and improve the corporation's daily operations. System administration and Oracle software installation efforts can also be reduced by cloning the Linux and Oracle installations on many separate servers instead of repeating the installation process on each server.

The DBA now has the option to choose between a crash recovery cloning method and the full recovery approach for creating the non-production clones. There are many uses for the cloned database that don't require the full recovery capability since they can simply be cloned again and replaced. Often during test phases of the software development process, or after student course work is completed, the cloned database is no longer needed, so it may be removed and cloned again for the next phase or class. The typical student database is very small, hence a huge time savings is gained in the repetitive cloning process. Cloning a test or development database for a large data warehouse system can be accommodated with minimal incremental disk usage and can save many hours of processing compared to the traditional tablespace-by-tablespace backup approach.

This application note described a SnapShot backup solution for the z/Linux database environment based on virtual cloning. The SnapShot capability is also available for many Open Systems environments. The Oracle cloning methodology is the same for Open Systems platforms, but the customer must use the SVA Administrator (SVAA) software to manually perform the process.

Appendix 1 — Database datafile list

/db1/oracle1/oradata/stkdb/stk_1998_02.dbf

/db1/oracle1/oradata/stkdb/stk_1998_idx_01.dbf

/db1/oracle1/oradata/stkdb/stk_1998_idx_02.dbf

/opt/oracle/oradata/stkdb/users_idx_01.dbf

/oracle/ora02/oradata/stkdb/control01.ctl

/oracle/ora02/oradata/stkdb/rbs01.dbf

/oracle/ora02/oradata/stkdb/stk_1998_01.dbf

/oracle/ora02/oradata/stkdb/system01.dbf

/oracle/ora02/oradata/stkdb/tools01.dbf

/oracle/ora03/oradata/stkdb/control02.ctl

/oracle/ora03/oradata/stkdb/redo1_01.log

/oracle/ora03/oradata/stkdb/redo2_01.log

/oracle/ora03/oradata/stkdb/redo3_01.log

/oracle/ora03/oradata/stkdb/redo4_01.log

/oracle/ora03/oradata/stkdb/redo5_01.log

/oracle/ora03/oradata/stkdb/redo6_01.log

/oracle/ora03/oradata/stkdb/stk_1999_01.dbf

/oracle/ora03/oradata/stkdb/temp01.dbf

/oracle/ora03/oradata/stkdb/users01.dbf

/oracle/ora04/oradata/stkdb/control03.ctl

/oracle/ora04/oradata/stkdb/redo1_02.log

/oracle/ora04/oradata/stkdb/redo2_02.log

/oracle/ora04/oradata/stkdb/redo3_02.log

/oracle/ora04/oradata/stkdb/redo4_02.log

/oracle/ora04/oradata/stkdb/redo5_02.log

/oracle/ora04/oradata/stkdb/redo6_02.log

/oracle/ora04/oradata/stkdb/stk_2000_01.dbf

/oracle/ora05/oradata/stkdb/stk_2001_01.dbf

/oracle/ora06/oradata/stkdb/stk_2002_01.dbf

Appendix 2 — Clone database alert log

Starting up ORACLE RDBMS Version: 9.2.0.1.0.

System parameters with non-default values:

```
  processes          = 50
  shared_pool_size      = 4194304
  control_files       =
/oracle/ora02/oradata/stkdb/control01.ctl, /oraclel
  db_block_buffers      = 15000
  db_block_size       = 8192
  compatible         = 9.2.0
  log_archive_start     = TRUE
  log_archive_dest_1     =
location=/oracle/admin/stkdb/arch
  log_archive_format     = arch_%t_%s.arc
  log_buffer         = 163840
  log_checkpoint_interval = 100000
  log_checkpoint_timeout  = 0
  rollback_segments     = RBS1_01, RBS2_01, RBS3_01,
RBS4_01, RBS5_01
  max_enabled_roles     = 30
  instance_name       = stkdb
  service_names       = stkdb
  background_dump_dest    = /oracle/admin/stkdb/bdump
  user_dump_dest       = /oracle/admin/stkdb/udump
  core_dump_dest       = /oracle/admin/stkdb/cdump
  sort_area_size       = 5242880
  sort_area_retained_size = 5242880
  db_name          = stkdb
  open_cursors        = 50
  os_authent_prefix     =
PMON started with pid=2
DBW0 started with pid=3
LGWR started with pid=4
CKPT started with pid=5
SMON started with pid=6
RECO started with pid=7
Wed Dec 18 11:32:39 2002
ARCH: STARTING ARCH PROCESSES
ARC0 started with pid=8
ARC0: Archival started
ARC1 started with pid=9
Wed Dec 18 11:32:40 2002
ARCH: STARTING ARCH PROCESSES COMPLETE
Wed Dec 18 11:32:40 2002
ARC0: Thread not mounted
Wed Dec 18 11:32:40 2002
ARC1: Archival started
ARC1: Thread not mounted
Wed Dec 18 11:32:40 2002
ALTER DATABASE  MOUNT
Wed Dec 18 11:32:44 2002
Successful mount of redo thread 1, with mount id
2552519384.
Wed Dec 18 11:32:44 2002
Database mounted in Exclusive Mode.
Completed: ALTER DATABASE  MOUNT
Wed Dec 18 11:33:20 2002
alter tablespace SYSTEM end backup
Wed Dec 18 11:33:20 2002
Completed: alter tablespace SYSTEM end backup
Wed Dec 18 11:33:20 2002
alter tablespace TOOLS end backup
Completed: alter tablespace TOOLS end backup
Wed Dec 18 11:33:20 2002
alter tablespace RBS end backup
Completed: alter tablespace RBS end backup
Wed Dec 18 11:33:21 2002
alter tablespace TEMP end backup
Completed: alter tablespace TEMP end backup
Wed Dec 18 11:33:21 2002
alter tablespace USERS end backup
Completed: alter tablespace USERS end backup
Wed Dec 18 11:33:21 2002
alter tablespace STK_1998 end backup
Completed: alter tablespace STK_1998 end backup
Wed Dec 18 11:33:21 2002
alter tablespace STK_1998_IDX end backup
Completed: alter tablespace STK_1998_IDX end backup
Wed Dec 18 11:33:21 2002
alter tablespace STK_1999 end backup
Completed: alter tablespace STK_1999 end backup
Wed Dec 18 11:33:21 2002
alter tablespace STK_2000 end backup
Completed: alter tablespace STK_2000 end backup
Wed Dec 18 11:33:21 2002
alter tablespace STK_2001 end backup
Completed: alter tablespace STK_2001 end backup
Wed Dec 18 11:33:25 2002
alter tablespace STK_2002 end backup
Completed: alter tablespace STK_2002 end backup
Wed Dec 18 11:34:22 2002
alter database open
Wed Dec 18 11:34:22 2002
Beginning crash recovery of 1 threads
Wed Dec 18 11:34:23 2002
Started first pass scan
Wed Dec 18 11:34:27 2002
Completed first pass scan
 5268 redo blocks read, 561 data blocks need recovery
Wed Dec 18 11:34:30 2002
Started recovery at
 Thread 1: logseq 154, block 2, scn 0.2564155
Recovery of Online Redo Log: Thread 1 Group 4 Seq 154
Reading mem 0
 Mem# 0 errs 0:
/oracle/ora03/oradata/stkdb/redo4_01.log
 Mem# 1 errs 0:
/oracle/ora04/oradata/stkdb/redo4_02.log
Wed Dec 18 11:34:37 2002
Ended recovery at
 Thread 1: logseq 154, block 5270, scn 0.2588276
 561 data blocks read, 561 data blocks written, 5268
redo blocks read
Crash recovery completed successfully
Wed Dec 18 11:34:37 2002
LGWR: Primary database is in CLUSTER CONSISTENT mode
Thread 1 advanced to log sequence 155
Thread 1 opened at log sequence 155
 Current log# 5 seq# 155 mem# 0: /oracle/ora03/ora-
```

```
data/stkdb/redo5_01.log
 Current log# 5 seq# 155 mem# 1: /oracle/ora04/ora-
data/stkdb/redo5_02.log
Successful open of redo thread 1.
Wed Dec 18 11:34:38 2002
MTTR advisory is disabled because FAST_START_MTTR_TAR-
GET is not set
Wed Dec 18 11:34:38 2002
ARC0: Evaluating archive  log 4 thread 1 sequence 154
ARC0: Beginning to archive log 4 thread 1 sequence 154
Creating archive destination LOG_ARCHIVE_DEST_1:
'/oracle/admin/stkdb/arch/arch'
Wed Dec 18 11:34:38 2002
SMON: enabling cache recovery
Wed Dec 18 11:34:46 2002
ARC0: Completed archiving log 4 thread 1 sequence 154
Wed Dec 18 11:34:48 2002
SMON: enabling tx recovery
Wed Dec 18 11:34:48 2002
Database Characterset is UTF8
replication_dependency_tracking turned off (no async
multimaster replication fo)
Completed: alter database open
```

References

Oracle9i DBA Handbook — Manage a Robust, High-Performance Oracle Database. Kevin Loney TUSC, Marlene Theriault, Oracle Press, 2002.

Oracle 24 x 7 — Tips and Techniques — Real-World Approaches to Ensuring Database Availability. Venkat S. Devraj, Raymond James Consulting, Oracle Press, 2000.

Mainframe Linux and StorageTek's SVA SnapVantage Whitepaper. Scott Ledbetter, Storage Technology Corporation. June 2002.

- http://www.stortek.com/pdfs/MainframeLinu_v5WP _MV9113A.FINAL.POSTED.pdf

Oracle9i Backup and Recovery Concepts Release 2 (9.2) — Oracle Corporation. 2002.

Using SVA SnapShot with Oracle, April 2001. Toulouse Research and Development Center. Part Number: TL-ENG-PRD-0477-A1.

Shared Virtual Array Administrator for VM Configruation and Administration — Storage Technology Corporation, September 2002. Part Number:313462902.

SnapVantage Installation, Customization, and Usage Guide — Storage Technology Corporation, October 2002. Part Number: 313494002.

Oracle Metalink: Bulletins & Documents — Oracle Corporation

- http://www.oracle.com/support/metalink/index.html

Guidelines for Using Snapshot Storage Systems for Oracle Databases — Nabil Osorio and Bill Lee, Oracle Corporation, October 2001.

STORAGETEK®

ABOUT STORAGETEK®

Storage Technology Corporation (NYSE: STK), a $2 billion worldwide company with headquarters in Louisville, CO, has been delivering a broad range of storage management solutions designed for IT professionals for over 30 years. StorageTek offers solutions that are easy to manage, integrate well with existing infrastructures and allow universal access to data across servers, media types and storage networks. StorageTek's practical and safe storage solutions for tape automation, disk storage systems and storage integration, coupled with a global services network, provide IT professionals with confidence and know-how to manage their entire storage management ecosystem today and in the future.

StorageTek products are available through a worldwide network. For more information, visit www.storagetek.com, or call 1.800.275.4785 or 01.303.673.2800.

WORLD HEADQUARTERS

Storage Technology Corporation
One StorageTek Drive
Louisville, Colorado 80028 USA
1.800.877.9220 or 01.303.673.5151