solaris

PART 1:
CONFIGURE & USE ZFS-HOSTED iSCSI GUEST STORAGE WITHIN THE SUN xVM HYPERVISOR

> Technical Brief

Simon Hayler, UK Systems Practice
Version 1.0 | *Last updated: 08/13/08*

Sun microsystems

About This Technical Brief

This document is intended for users familiar with the Sun xVM hypervisor who have used the "Install the Sun xVM Hypervisor and Use it to Configure Domains" technical brief to install domains. In the first technical brief, we learned how to install, configure and use the Sun xVM hypervisor on a single server. We also learned how to install and configure two guest domains: a paravirtualized Solaris Express Community Edition and a hardware virtual machine Windows 2003 Server. The ZFS file system was used to provide a block-based boot environment for both guests, which were located on the single server internal disks.

This technical brief is intended as the next step with the Sun xVM hypervisor. It builds on the two previously created guests and describes the required steps to:

1. Configure NFS and iSCSI via ZFS on a Solaris Community Edition-based server
2. Capture the guest data and transfer to the NFS server
3. Configure a Sun xVM hypervisor server to access both NFS and iSCSI
4. Extract the guest data and recreate on iSCSI
5. Define and start the two guests from the XML descriptor files

The use of ZFS and iSCSI to host the Sun xVM hypervisor guests is intended to provide a choice of guest hosting rather than local disk storage or NFS. The use of iSCSI provides block-level storage via ethernet and also provides flexibility in enabling the moving of iSCSI hosted guests between the Sun xVM hypervisor servers. ZFS is used to both capture the guest from a snapshot and move the compressed snapshot between servers. It can also be used to clone guests images quickly and easily. These methods are not limited to the Sun xVM hypervisor; the same approach could be used with respect to hosting Solaris Zones or Sun Logical Domains.

This technical brief is based on the Sun xVM hypervisor components contained within the Solaris Express Community Edition (SXCE). For more information on Solaris Express, refer to:
http://opensolaris.org/os/downloads/

Contents

Sun xVM Hypervisor Technical Brief

Introduction to the Snapshot, Send, and Receive Features of ZFS

This section describes the snapshot, send, and receive features of ZFS. It is assumed that the reader is familiar with the main features of ZFS (Zettabyte File System), such as zpools and zfs filesystems.

For more information on ZFS, see:
http://www.sun.com/2004-0914/feature/
http://opensolaris.org/os/community/zfs/

A *ZFS snapshot* is a read-only copy of a file system or volume. Snapshots can be created almost instantly, and initially consume no additional disk space within the pool. However, as data within the active dataset changes, the snapshot consumes disk space by continuing to reference the old data, which prevents the space from being freed. Snapshots of volumes cannot be accessed directly, but they can be cloned, backed up, and rolled back to a previous level.

A *ZFS send* creates a stream representation of a snapshot that is written to standard output. The output can be redirected to a file or a different server, for example, by using ssh. By default, a full stream is generated. The -i switch can be used to generate an incremental stream.

A *ZFS receive* creates a snapshot in which the contents are as specified in the stream provided on standard input; in this case, via a file. If a full stream is received, then a new file system is created.

This brief shows how to use, send, and receive ZFS snapshots to do the following:

1. Take ZFS snapshots of the two guests
2. Use ZFS send to create stream representations of the guest snapshots to files
3. Use ZFS receive to create new ZFS filesystems and snapshots of the guests

Introduction to iSCSI

iSCSI is the protocol that allows SCSI commands to be sent in IP packets. To be more specific, iSCSI is designed to be a protocol for a storage initiator (i.e., client) to send SCSI commands to a storage target (i.e., tape or disk or any SCSI device) over the IP networking protocol.

Within Solaris Community Edition, both the initiator and target functionality are supported and able to be configured. Two configuration commands are used: iscsitadm and iscsiadm. The *iscsitadm* command configures and manages the Solaris iSCSI target devices. After the target devices are set up, the *iscsiadm* command is used on the initiator, which identifies and enables the iSCSI targets for use.

iSCSI Discovery is the process that presents the initiator with a list of available targets. The discovery method is the mechanism to find iSCSI targets. Three methods are currently available:

1. **Internet Storage Name Service (iSNS):** Potential targets are discovered by interacting with one or more iSNS servers. An iSNS (Internet storage name service) server enables:
   - Automated discovery
   - Management
   - Configuration of iSCSI Storage devices

   iSNS is akin to the Fiber Channel (FC) Fabric services typically served by the FC switch:
   - Sees devices come and go on the SAN
   - Allows zoning

2. **Send Targets:** If a node exposes a large number of iSCSI targets, such as an iSCSI to FC bridge, the IP/Port of the node can be used by the iSCSI initiator using the Send Targets option to perform the device discovery.
3. **Static:** If a node has a small number of iSCSI targets or if the initiator access to the targets needs to be restricted, Static device discovery should be used. Static discovery is also useful if access to targets needs to be restricted to a network interface type; for example, Infiniband as opposed to Ethernet.

For this technical brief, we will be using iSCSI targets and initiators to do the following:
1. Create iSCSI static bound targets on a Sun Ultra 24 Server
2. Configure iSCSI initiators on a Sun Ultra 40 to host the guests

Server Environment

The server environment is made up of the following components:

Sun Ultra 40 M2:
Hardware Configuration: - 2 x 2.2Ghz Dual Core AMD CPUs with 8GB RAM
Role: Sun xVM hypervisor server running Solaris Community Edition build 85

Sun Ultra 24:
Hardware Configuration: - 1 x 2.4Ghz Quad Core Intel CPU with 4GB RAM
Role: Sun iSCSI & NFS server running Solaris Community Edition build 85

Note: Both workstations are on a single flat subnet.

Configuration Prerequisites and Assumptions

The following requirements must be met:
- Reader is familiar with the Sun xVM hypervisor and has created domains by following instructions in the "Install Sun xVM Hypervisor and Use it to Configure Domains" brief
- Access to ISO images for Solaris and Windows 2003
- Access to Windows 2003 licenses
- Reader is comfortable with basic administration of Solaris and Windows
- Logged in as user root
- Access to a local DHCP server or unused static IP addresses
- Solaris Community Edition build 85 is installed and the Sun xVM hypervisor components are enabled
- On the Sun xVM hypervisor Ultra 40 M2, two server guest domains are configured under ZFS:
  > Hardware Virtual Machine Windows Server 2003
  > Paravirtualized Solaris Community Edition Build 85
- Solaris Community Edition build 85 is installed with adequate spare disk space for guest domains and backup files on a single server. In this environment, the Sun Ultra 24 has a spare slice (c0t0d0s4) on the main boot disk and a spare internal disk (c0t1d0), which is used for a ZFS zpool (images) to store the guest and backup data.

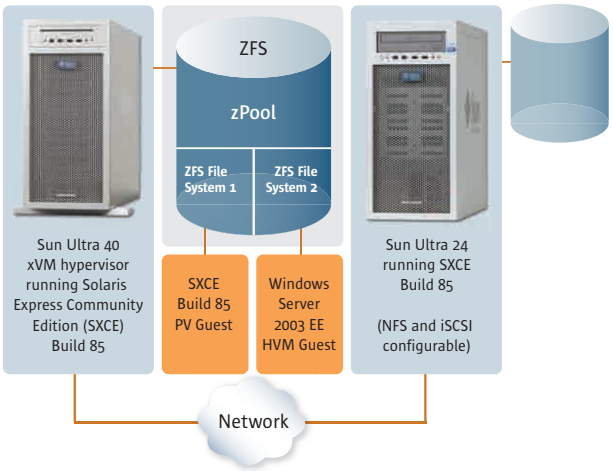

Figure 1—Details the configuration as it stands.

NFS & iSCSI Configuration on Sun Ultra 24 Server

In this section, the steps needed to configure both NFS and iSCSI on the Sun Ultra 24 server are described. This server will serve out NFS and iSCSI for use by the Sun Ultra 40 Server.

1. Create a zpool to host the Virtual Machines & backup data:

```
# /usr/sbin/zpool create -f images c0t0d0s4 c0t1d0
```

Note: The above example creates a zpool on a single slice (c0t0d0s4) of the boot disk and a whole spare disk (c0t1d0) on the Sun Ultra 24 workstation. The boot disk is partitioned as slice 0 for root and 1 for swap; the rest is allocated to slice 4 for part of the zfs pool. The -f option forces the action in case old UFS slices or zpools exist.

2. List and check new zpool images status:

```
# /usr/sbin/zpool list images
NAME        SIZE       USED       AVAIL      CAP        HEALTH     ALTROOT
images      232G       2.00G      231G       0%         ONLINE     -

# /usr/sbin/zpool status images
pool: images
state: ONLINE
scrub: none requested
config:

            NAME       STATE      READ       WRITE      CKSUM
            images     ONLINE     0          0          0
            c0t0d0s4   ONLINE     0          0          0
            c0t1d0     ONLINE     0          0          0

errors: No known data errors
```

3. Setup ZFS for NFS:

```
# /usr/sbin/zfs create images/nfs
# /usr/sbin/zfs list
NAME              USED       AVAIL      REFER      MOUNTPOINT
images            134K       229G       18K        /images
images/nfs        18K        229G       18K        /images/nfs
# /usr/sbin/zfs set sharenfs=on images/nfs
# /usr/sbin/zfs set sharenfs="rw,anon=0" images/nfs
# /usr/sbin/zfs list -o name,sharenfs
NAME              SHARENFS
images            off
images/nfs        rw,anon=0
```

Note: The NFS share is set up for read / write access (rw) and also root owner status (anon=0).

4. List the present iSCSI configuration:

```
# /usr/sbin/iscsiadm list discovery
Discovery:
            Static: disabled
            Send Targets: disabled
            iSNS: disabled
```

Note: Because we want to explicitly present targets, we need to enable static discovery rather than send targets. Go to “Introduction to iSCSI” on Page 1 for more information on iSCSI discovery options.

5. Enable and check iSCSI static discovery:

```
# /usr/sbin/iscsiadm modify discovery -s enable
# /usr/sbin/iscsiadm list discovery
            Static: enabled
            Send Targets: disabled
            iSNS: disabled
```

6. Create a ZFS filesystem for the iSCSI LUNs:

```
# /usr/sbin/zfs create images/iscsi_luns
# /usr/sbin/zfs list
NAME                   USED       AVAIL      REFER      MOUNTPOINT
images                 8.26G      221G       19K        /images
images/iscsi_luns      18K        221G       18K        /images/iscsi_luns
images/nfs             8.26G      221G       8.26G      /images/nfs
```

7. Configure and check iSCSI for ZFS:

```
# /usr/sbin/zfs set shareiscsi=on images/iscsi_luns
```

Note: Remove the existing mountpoint iscsi_luns. They will be used to house block rather than file-based data:

```
# /usr/sbin/zfs set mountpoint=none images/iscsi_luns
```

List zfs to verify that the mountpoints have been removed:

```
# /usr/sbin/zfs list
NAME                  USED       AVAIL      REFER      MOUNTPOINT
images                8.26G      221G       19K        /images
images/iscsi_luns     18K        221G       18K        none
images/nfs            8.26G      221G       8.26G      /images/nfs
```

8. Create and check two guest iSCSI LUNs:

```
# /usr/sbin/zfs create -V 10G images/iscsi_luns/guest1
# /usr/sbin/zfs create -V 10G images/iscsi_luns/guest2

# /usr/sbin/zfs get mountpoint
NAME                          PROPERTY       VALUE           SOURCE
images                        mountpoint     /images         default
images/iscsi_luns             mountpoint     none            local
images/iscsi_luns/guest1      mountpoint     -               -
images/iscsi_luns/guest2      mountpoint     -               -
images/nfs                    mountpoint     /images/nfs     default

# /usr/sbin/zfs get shareiscsi
NAME                          PROPERTY       VALUE           SOURCE
images                        shareiscsi     off             default
images/iscsi_luns             shareiscsi     on              local
images/iscsi_luns/guest1      shareiscsi     on              inherited from
images/iscsi_luns
images/iscsi_luns/guest2      shareiscsi     on              inherited from
images/iscsi_luns
images/nfs                    shareiscsi     off             default
```

Note: Because the guest [1-2] ZFS file systems were created under images/iscsi_luns, they inherit the shareiscsi. Note the value off for images/nfs.

9. Check the iSCSI target configuration:

```
# /usr/sbin/iscsitadm list target -v
Target: images/iscsi_luns/guest1
     iSCSI Name: iqn.1986-03.com.sun:02:6fb72f78-9b42-c080-b4fe-efa3d26f4f79
     Alias: images/iscsi_luns/guest1
     Connections: 0
     ACL list:
     TPGT list:
     LUN information:
          LUN: 0
               GUID: 0
               VID: SUN
               PID: SOLARIS
               Type: disk
               Size:   10G
```

```
            Backing store: /dev/zvol/rdsk/images/iscsi_luns/guest1
            Status: online
Target: images/iscsi_luns/guest2
    iSCSI Name: iqn.1986-03.com.sun:02:5dfef158-d101-6d48-cd25-e9580e65f42c
    Alias: images/iscsi_luns/guest2
    Connections: 0
    ACL list:
    TPGT list:
    LUN information:
        LUN: 0
            GUID: 0
            VID: SUN
            PID: SOLARIS
            Type: disk
            Size:   10G
            Backing store: /dev/zvol/rdsk/images/iscsi_luns/guest2
            Status: online
```

10. Gather the iSCSI target information for initiator configuration:

To enable the iSCSI initiators running on the Sun Ultra 40 to access the static iSCSI targets running on the Sun Ultra 24, it is necessary to generate the 'iscsiadm add static-config' commands for each target. Because the syntax is fairly complex, a script can be run on the iSCSI target host to generate each command to be run on the iSCSI initiator hosts.

"Appendix A: iscsi.bash Script" contains this script. Copy the script into a directory (such as /var/tmp) and assign UNIX permissions of 755. The script requires a single input: the IP address of the target server. The IP used here is 129.146.229.223.

The script is run as follows:

```
# /var/tmp/iscsi.bash 129.146.229.223
Target: images/iscsi_luns/guest1
    iSCSI Name: iqn.1986-03.com.sun:02:6fb72f78-9b42-c080-b4fe-efa3d26f4f79
    Connections: 0
Target: images/iscsi_luns/guest2
    iSCSI Name: iqn.1986-03.com.sun:02:5dfef158-d101-6d48-cd25-e9580e65f42c
    Connections: 0
```

Run these commands on both initiators. Please note that your iSCSI addresses will be different:

```
iscsiadm add static-config iqn.1986-03.com.sun:02:6fb72f78-9b42-c080-b4fe-
efa3d26f4f79,129.146.229.223
iscsiadm add static-config iqn.1986-03.com.sun:02:5dfef158-d101-6d48-cd25-
e9580e65f42c,129.146.229.223
```

The commands gathered by the script will be used in "NFS and iSCSI Configuration on Sun Ultra 40 Server".

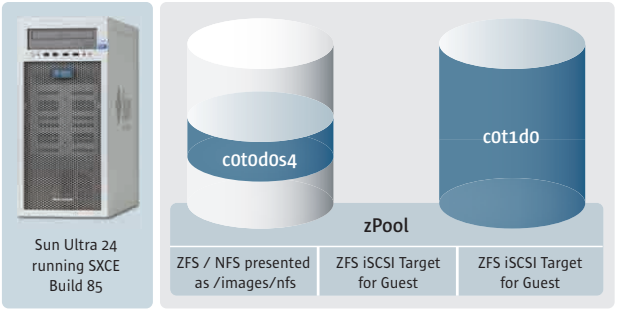

Figure 2—Details the configuration of the Sun Ultra 24 Server following the ZFS, NFS and iSCSI steps described above.

Sun Ultra 40 Guest Information Capture and Transfer to Sun Ultra 24 NFS Server Steps

On the Sun Ultra 40 server, this section describes the capture of the guest data and the export of the guest images to the Sun Ultra 24 NFS server. For completeness, the existing guest environment needs to be destroyed.

1. Mount nfs share from Sun Ultra 24 NFS/iSCSI server:

```
# /usr/bin/mkdir -p /nfs
# /usr/sbin/mount -F nfs -o forcedirectio 129.146.229.223:/images/nfs /nfs
# /usr/bin/mkdir -p /nfs/dumps
```

Note: The Sun Ultra 24 server has IP 129.146.229.223.

For performance reasons, the *forcedirectio* option is used to disable any buffering in the client.
The NFS share /nfs will contain the guest backup data.

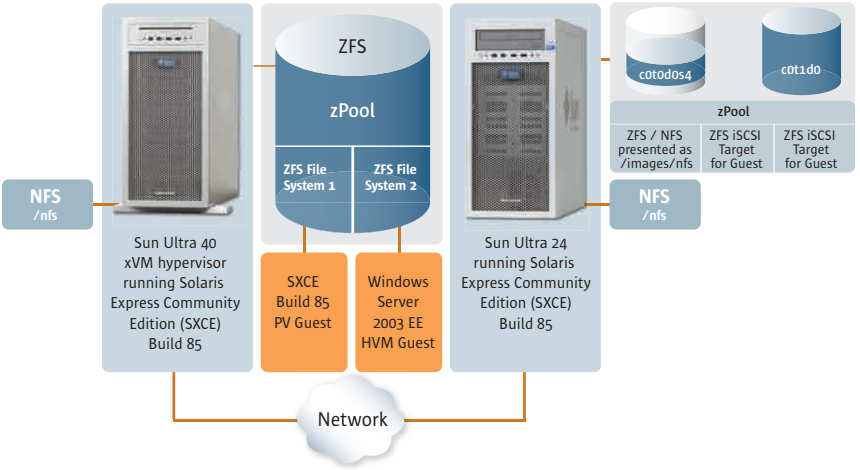

Figure 3—Details this configuration.

2. List the active guest configuration:

```
# /usr/bin/virsh list --all
Id Name                          State
----------------------------------
 0 Domain-0                      running
14 guest-nvb85-PV                blocked
16 guest-w2003se-HVM             blocked

# /usr/sbin/zfs list
NAME                             USED    AVAIL   REFER   MOUNTPOINT
xvm_pool                         84.1G   86.2G   12.1G   /xvm_pool
xvm_pool/guest-nvb85-PV          10G     90.1G   6.09G   -
xvm_pool/guest-w2003se-HVM       11.9G   96.2G   1.94G   -
```

Note: There are two guests running on ZFS filesystems under the zpool xvm_pool.

3. Take a ZFS snapshot of guest-w2003se-HVM:

```
# /usr/sbin/zfs snapshot xvm_pool/guest-w2003se-HVM@guest-w2003se-HVM-snap
# /usr/sbin/zfs list
NAME                             USED    AVAIL   REFER   MOUNTPOINT
xvm_pool                         84.1G   86.2G   12.1G   /xvm_pool
xvm_pool/guest-nvb85-PV          10G     90.1G   6.09G   -
xvm_pool/guest-w2003se-HVM       11.9G   96.2G   1.94G   -
xvm_pool/guest-w2003se-HVM
@guest-w2003se-HVM-snap          0       -       1.94G   -
```

4. ZFS send the guest-w2003se-HVM snapshot to NFS server dump location:

```
# /usr/sbin/zfs send xvm_pool/guest-w2003se-HVM@guest-w2003se-HVM-snap >
/nfs/dumps/guest-w2003se-HVM-snap.backup
```

5. Dump XML format of guest-w2003se-HVM to NFS server dump location:

```
# /usr/bin/virsh dumpxml guest-w2003se-HVM > /nfs/dumps/guest-w2003se-HVM.xml
```

6. Take a ZFS snapshot of guest-nvb85-PV:

```
# /usr/sbin/zfs snapshot xvm_pool/guest-nvb85-PV@guest-nvb85-PV-snap
# /usr/sbin/zfs list
NAME                             USED    AVAIL   REFER   MOUNTPOINT
xvm_pool                         95.0G   75.3G   16.9G   /xvm_pool
xvm_pool/guest-nvb85-PV          16.1G   85.3G   6.09G   -
xvm_pool/guest-nvb85-PV
@guest-nvb85-PV-snap             0       -       6.09G   -
xvm_pool/guest-w2003se-HVM       11.9G   85.2G   1.94G   -
xvm_pool/guest-w2003se-HVM
@guest-w2003se-HVM-snap          63.3M   -       1.94G   -
```

7. ZFS send guest-nvb85-PV snapshot to NFS server dump location:

```
# /usr/sbin/zfs send xvm_pool/guest-nvb85-PV@guest-nvb85-PV-snap >
/nfs/dumps/guest-nvb85-PV-snap.backup
```

8. Dump XML format of guest-nvb85-PV to NFS server dump location:

```
# /usr/bin/virsh dumpxml guest-nvb85-PV > /nfs/dumps/guest-nvb85-PV.xml
```

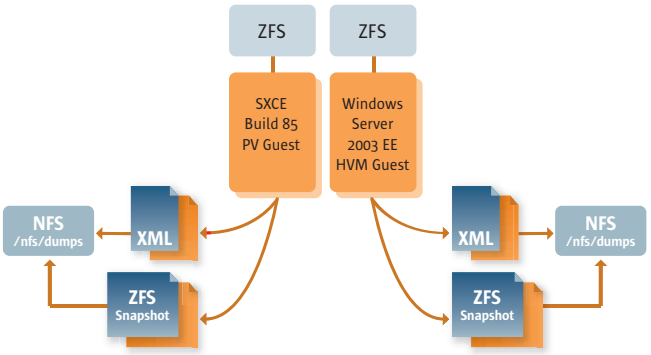

Figure 4—Details the present configuration.

9. Shutdown and undefine guests:

```
# /usr/bin/virsh list --all
Id Name                 State
----------------------------------
 0 Domain-0             running
14 guest-nvb85-PV       blocked
16 guest-w2003se-HVM    blocked

# /usr/bin/virsh shutdown guest-nvb85-PV
Domain guest-nvb85-PV is being shutdown

# /usr/bin/virsh shutdown guest-w2003se-HVM
Domain guest-w2003se-HVM is being shutdown

# /usr/bin/virsh list --all
Id Name                 State
----------------------------------
 0 Domain-0             running
 - guest-nvb85-PV       shut off
 - guest-w2003se-HVM    shut off

# /usr/bin/virsh undefine guest-nvb85-PV
Domain guest-nvb85-PV has been undefined
```

```
# /usr/bin/virsh undefine guest-w2003se-HVM
Domain guest-w2003se-HVM has been undefined

# /usr/bin/virsh list --all
Id Name                 State
----------------------------------
 0 Domain-0             running
```

10. Destroy the ZFS snapshots and volumes associated with the guests:

```
# /usr/sbin/zfs destroy -r xvm_pool/guest-w2003se-HVM
# /usr/sbin/zfs destroy -r xvm_pool/guest-nvb85-PV
```

Note: The -r flag removes any children and snapshots associated with the filesystem.

11. Destroy the ZFS zpool xvm:

```
# /usr/sbin/zpool destroy xvm_pool
```

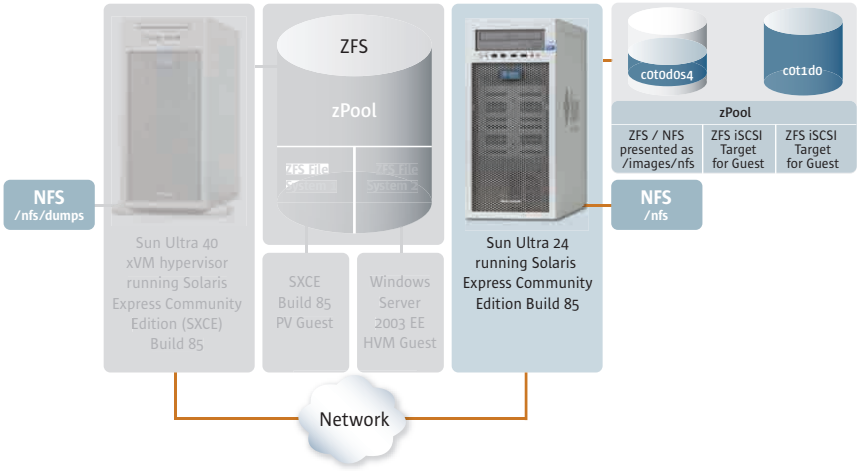

Figure 5—Details the present configuration.

NFS and iSCSI Configuration on the Sun Ultra 40 Server

In this section, the steps needed to configure both NFS and iSCSI on the Sun Ultra 40 server are described. This server will utilize the NFS and iSCSI resources served by the Sun Ultra 24 Server.

1. Check and enable iSCSI configuration:

```
# /usr/sbin/iscsiadm list discovery
Discovery:
        Static: disabled
        Send Targets: disabled
        iSNS: disabled
```

Enable Static Discovery:

```
# /usr/sbin/iscsiadm modify discovery -s enable
# /usr/sbin/iscsiadm list discovery
Discovery:
        Static: enabled
        Send Targets: disabled
        iSNS: disabled
```

2. Run the output from the iscsi.bash script to configure access to the iSCSI LUNs.

   Go to "Gather the iSCSI Target Information for Initiator Configuration", and to "Appendix A iscsi.bash Script" for more information and access to the script.

```
# /usr/sbin/iscsiadm add static-config iqn.1986-03.com.sun:02:6fb72f78-9b42-
c080-b4fe-efa3d26f4f79,129.146.229.223
# /usr/sbin/iscsiadm add static-config iqn.1986-03.com.sun:02:5dfef158-d101-
6d48-cd25-e9580e65f42c,129.146.229.223
```

3. Check that the iSCSI targets are present:

```
# /usr/sbin/iscsiadm list target
Target: iqn.1986-03.com.sun:02:5dfef158-d101-6d48-cd25-e9580e65f42c
        Alias: images/iscsi_luns/guest2
        TPGT: 1
        ISID: 4000002a0000
        Connections: 1
Target: iqn.1986-03.com.sun:02:6fb72f78-9b42-c080-b4fe-efa3d26f4f79
        Alias: images/iscsi_luns/guest1
        TPGT: 1
        ISID: 4000002a0000
        Connections: 1
```

4. Check that the iSCSI targets are presented as disks via format:

```
# /usr/sbin/format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <DEFAULT cyl 9726 alt 2 hd 255 sec 63>
          /pci@0,0/pci108e,534d@5/disk@0,0
       1. c3t010000144F5857B200002A0047F12CF6d0 <DEFAULT cyl 1303 alt 2 hd
255 sec 63>
          /scsi_vhci/disk@g010000144f5857b200002a0047f12cf6
       2. c3t010000144F5857B200002A0047F12CF7d0 <DEFAULT cyl 1303 alt 2 hd
255 sec 63>
          /scsi_vhci/disk@g010000144f5857b200002a0047f12cf7
Specify disk (enter its number):
```

Note: Use ^D (CONTROL key and D key) to exit format.

5. Mount the NFS share from the Sun Ultra 24 with directIO:

```
# /usr/bin/mkdir -p /nfs
# /usr/sbin/mount -o forcedirectio 129.146.229.223:/images/nfs /nfs
```

Note: The Sun Ultra 24 server has IP 129.146.229.223. For performance reasons, the *forcedirectio* option is used to disable any buffering in the client.

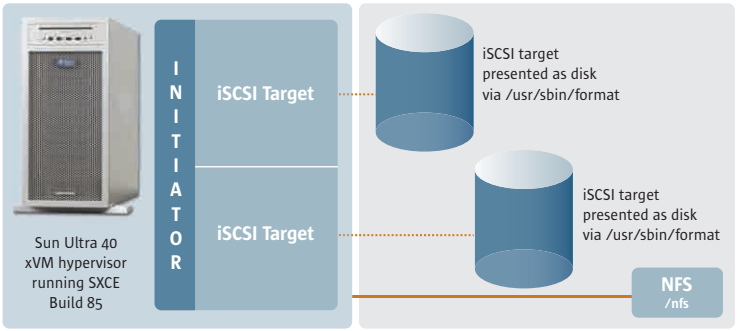

Figure 6—Details this configuration.

ZFS Guest Configuration on Sun Ultra 40 Server

In this section, the steps needed to configure ZFS for the Sun xVM guests to be hosted on the Sun Ultra 40 server are described.

1. Create a zpool for guest1:

```
# /usr/sbin/zpool create guest1 c3t010000144F5857B200002A0047F12CF6d0
# /usr/sbin/zfs list
NAME     USED     AVAIL    REFER    MOUNTPOINT
guest1   106K     9.78G    18K      /guest1

# /usr/sbin/zpool list
NAME     SIZE     USED     AVAIL    CAP      HEALTH   ALTROOT
guest1   9.94G    478K     9.94G    0%       ONLINE   -

# /usr/sbin/zpool status guest1
pool: guest1
state: ONLINE
scrub: none requested
config:

NAME                                        STATE    READ     WRITE    CKSUM
guest1                                      ONLINE   0        0        0
c3t010000144F5857B200002A0047F12CF6d0       ONLINE   0        0        0
```

Note: c3t010000144F5857B200002A0047F12CF6d0 used for the guest1 ZFS zpool is disk 1 captured from the format command in “Check the iSCSI Targets are Present as Disks Via Format”.

2. Create a zpool for guest2:

```
# /usr/sbin/zpool create guest2 c3t010000144F5857B200002A0047F12CF7d0
# /usr/sbin/zfs list
NAME     USED     AVAIL    REFER    MOUNTPOINT
guest1   106K     9.78G    18K      /guest1
guest2   106K     9.78G    18K      /guest2

# /usr/sbin/zpool list
NAME     SIZE     USED     AVAIL    CAP      HEALTH   ALTROOT
guest1   9.94G    478K     9.94G    0%       ONLINE   -
guest2   9.94G    478K     9.94G    0%       ONLINE   -

# /usr/sbin/zpool status guest2
pool: guest2
state: ONLINE
scrub: none requested
config:

NAME                                       STATE    READ     WRITE    CKSUM
guest2                                     ONLINE   0        0        0
c3t010000144F5857B200002A0047F12CF7d0      ONLINE   0        0        0
```

Note: c3t010000144F5857B200002A0047F12CF7d0 used for the guest1 ZFS zpool is disk 2 captured from the format command in "Check the iSCSI Targets are Presented as Disks Via Format".

3. Restore the guest-nvb85-PV ZFS zpool and filesystem from the ZFS backup:

```
# /usr/sbin/zfs receive guest1/guest1 < /nfs/dumps/guest-nvb85-PV-snap.backup
# /usr/sbin/zfs list
NAME                                  USED     AVAIL    REFER    MOUNTPOINT
guest1                                6.11G    3.67G    18K      /guest1
guest1/guest1                         6.11G    3.67G    6.09G    -
guest1/guest1@guest-nvb85-PV-snap     22.9M    -        6.09G    -
```

Note: Remove the mountpoint guest1 because it will house block rather than file based data:

```
# /usr/sbin/zfs set mountpoint=none guest1
# /usr/sbin/zfs list
NAME                                  USED     AVAIL    REFER    MOUNTPOINT
guest1                                6.11G    3.67G    18K      none
guest1/guest1                         6.11G    3.67G    6.09G    -
guest1/guest1@guest-nvb85-PV-snap     22.9M    -        6.09G    -
```

4. Restore the guest-w2003se-HVM ZFS zpool and filesystem from the ZFS backup:

```
# /usr/sbin/zfs receive guest2/guest2 < /nfs/dumps/guest-w2003se-HVM-
snap.backup
# /usr/sbin/zfs list
NAME                                    USED    AVAIL   REFER   MOUNTPOINT
guest1                                  6.11G   3.67G   18K     /guest1
guest1/guest1                           6.11G   3.67G   6.09G   -
guest1/guest1@guest-nvb85-PV-snap       22.9M   -       6.09G   -
guest2                                  6.11G   3.67G   18K     /guest2
guest2/guest2                           6.11G   3.67G   6.09G   -
guest2/guest2@guest-w2003se-HVM-snap    22.9M   -       6.09G   -
```

Note: Remove the mountpoint guest2 because it will house block rather than file-based data:

```
# /usr/sbin/zfs set mountpoint=none guest2
# /usr/sbin/zfs list
NAME                                    USED    AVAIL   REFER   MOUNTPOINT
guest1                                  6.11G   3.67G   18K     none
guest1/guest1                           6.11G   3.67G   6.09G   -
guest1/guest1@guest-nvb85-PV-snap       22.9M   -       6.09G   -
guest2                                  6.11G   3.67G   18K     none
guest2/guest2                           6.11G   3.67G   6.09G   -
guest2/guest2@guest-w2003se-HVM-snap    22.9M   -       6.09G   -
```

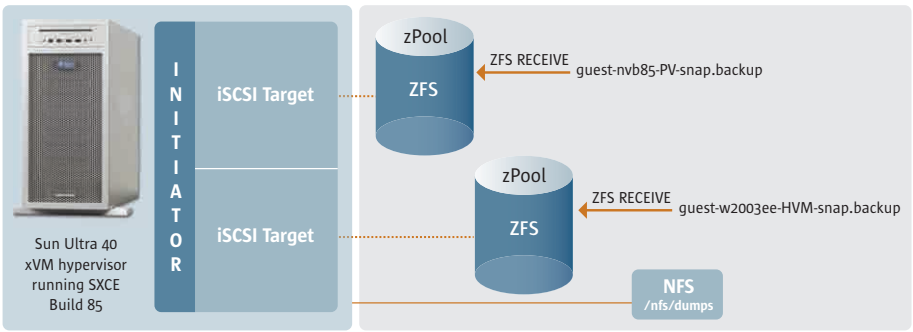

Figure 7—Details this configuration.

Configure and Start Guests on Sun Ultra 40 Server

In this section, the steps needed to configure and start the Sun xVM guests hosted on the Sun Ultra 40 are described.

1. Review the guest-nvb85-PV xml backup file.

   Because the backup XML file could have been captured while the guest was active on another server, some edits might be required to successfully activate the PV guest on the new server.

   The captured XML file (/nfs/dumps/guest-nvb85-PV.xml) for guest-nvb85-PV is shown below. The fields in italics are the fields that may need to be changed:

```
<domain type='xen' id='20'>
  <name>guest-nvb85-PV</name>
  <uuid>3225728d-e558-8e3f-19a6-7720904f6b55</uuid>
  <bootloader>/usr/lib/xen/bin/pygrub</bootloader>
  <os>
    <type>linux</type>
  </os>
  <memory>1048576</memory>
  <vcpu>1</vcpu>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <interface type='ethernet'>
      <target dev='vif20.0'/>
      <mac address='00:16:3e:61:47:7c'/>
      <script path='vif-vnic'/>
    </interface>
    <disk type='block' device='disk'>
      <driver name='phy'/>
      <source dev='/dev/zvol/dsk/xvm_pool/guest-nvb85-PV'/>
      <target dev='0'/>
    </disk>
    <console tty='/dev/pts/6'/>
  </devices>
</domain>
```

   Note: Change options:

   <name>
   This field can be changed if the name of the guest has to be different within Sun xVM hypervisor. Changing this field will not affect the host or computer name within the guest operating system.

   <uuid>
   If removed, this field will automatically be regenerated using a randomly generated 32bit hex number when the guest is defined.

<mac address>
If removed, this field will automatically be regenerated using a randomly generated address when the guest is defined.

<source dev>
This field must be changed if the guest boot device location has changed.

For our purposes, the complete edited file for the PV guest-nvb85-PV would be:

```
<domain type='xen' id='20'>
  <name>guest-nvb85-PV</name>
  <uuid>3225728d-e558-8e3f-19a6-7720904f6b55</uuid>
  <bootloader>/usr/lib/xen/bin/pygrub</bootloader>
  <os>
    <type>linux</type>
  </os>
  <memory>1048576</memory>
  <vcpu>1</vcpu>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <interface type='ethernet'>
      <target dev='vif20.0'/>
      <mac address='00:16:3e:61:47:7c'/>
      <script path='vif-vnic'/>
    </interface>
    <disk type='block' device='disk'>
      <driver name='phy'/>
      <source dev='/dev/zvol/dsk/guest1/guest1'/>
      <target dev='0'/>
    </disk>
    <console tty='/dev/pts/6'/>
  </devices>
</domain>
```

Note: The only change made (in italics) is to the <source dev> field, where the new zpool and zfs filesystem location has been changed to reflect the new iSCSI based ZFS location. The name, MAC address, and UUI remain the same.

2. Define guest-nvb85-PV from the edited XML backup file:

```
# /usr/bin/virsh define /nfs/dumps/guest-nvb85-PV.xml
Domain guest-nvb85-PV defined from /nfs/dumps/guest-nvb85-PV.xml
```

3. Start and check guest-nvb85-PV:

```
# /usr/bin/virsh start guest-nvb85-PV
Domain guest-nvb85-PV started

# /usr/bin/virsh list --all
Id Name                    State
----------------------------------
 0 Domain-0                running
40 guest-nvb85-PV          blocked

# /usr/bin/virsh console guest-nvb85-PV
v3.1.2-xvm chgset 'Mon Mar 03 23:05:33 2008 -0800 15630:a1a16c966b70'
SunOS Release 5.11 Version snv_85 64-bit
Copyright 1983-2008 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Hostname: dhcp-umpk17-229-254
Reading ZFS config: done.


dhcp-umpk17-229-254 console login:
```

Note: To exit the console, use the CRTL] keys together.

4. Review the guest-w2003se-HVM XML backup file:

Because the backup XML file could have been captured while the guest was active on another server, some edits might be required to successfully activate the HVM guest on the new server.

Following is the captured XML file (/nfs/dumps/guest-w2003se-HVM.xml) for guest-w2003se-HVM. The fields in italics are the fields that might need to be changed:

```
<domain type='xen' id='10'>
  <name>guest-w2003se-HVM</name>
  <uuid>af910de2-0ddb-58ed-05fa-7ffa0affacc3</uuid>
  <os>
    <type>hvm</type>
    <loader>/usr/lib/xen/boot/hvmloader</loader>
    <boot dev='hd'/>
  </os>
  <memory>1048576</memory>
  <vcpu>1</vcpu>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <features>
    <pae/>
  </features>
  <clock offset='utc'/>
```

```
  <devices>
    <emulator>/usr/lib/xen/bin/qemu-dm</emulator>
    <interface type='ethernet'>
      <target dev='vif10.0'/>
      <mac address='00:16:3e:51:34:a9'/>
      <script path='vif-vnic'/>
    </interface>
    <disk type='block' device='disk'>
      <driver name='phy'/>
      <source dev='/dev/zvol/dsk/xvm_pool/guest-w2003se-HVM'/>
      <target dev='hda'/>
    </disk>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5900'/>
  </devices>
</domain>
```

Note: Change options:

<name>
This field can be changed if the name of the guest has to be different within Sun xVM hypervisor. Changing this field will not affect the host or computer name with the guest operating system.

<uuid>
If removed, this field will automatically be regenerated using a randomly generated 32bit hex number when the guest is defined.

<mac address>
If removed, this field will automatically be regenerated using a randomly generated address when the guest is defined.

<source dev>
This field must be changed if the guest boot device location has changed.

<input type>
This field should be changed as shown to work around a tracking issue with the mouse:
Change the :

```
<input type='mouse' bus='ps2'/>
```

to:

```
<input type='tablet' bus='usb'/>
```

<graphics type>
This field defines a port number for the VNC server to connect to for HVM guests. This number must be unique from all other HVM guests running on the system. Issue the following command to ascertain which port numbers any HVM guests are set to:

```
# /usr/bin/virsh vncdisplay <guest_name>
```

Note: If the output of the above command returns a blank result, the guest is either shutdown or it is a PV guest that does not require VNC.

Once the ports in use are identified, start at port 5900 and add 1 to the greatest number returned by the output of the vncdisplay command.

If multiple HVM guests are to be configured, then the following option can be used within the file:

```
<graphics type='vnc' port='-1'/>
```

This setting will enable the allocation of a port above 5900 automatically.

For our purposes, the complete edited file for the HVM guest guest-w2003se-HVM would be:

```
<domain type='xen' id='16'>
  <name>guest-w2003se-HVM</name>
  <uuid>af910de2-0ddb-58ed-05fa-7ffa0affacc3</uuid>
  <os>
    <type>hvm</type>
    <loader>/usr/lib/xen/boot/hvmloader</loader>
    <boot dev='hd'/>
  </os>
  <memory>1048576</memory>
  <vcpu>1</vcpu>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <features>
    <pae/>
  </features>
  <clock offset='utc'/>
  <devices>
    <emulator>/usr/lib/xen/bin/qemu-dm</emulator>
    <interface type='ethernet'>
      <target dev='vif16.0'/>
        <mac address='00:16:3e:51:34:a9'/>
      <script path='vif-vnic'/>
    </interface>
    <disk type='block' device='disk'>
      <driver name='phy'/>
      <source dev='/dev/zvol/dsk/guest2/guest2'/>
      <target dev='hda'/>
    </disk>
```

```
    <input type='tablet' bus='usb'/>
    <graphics type='vnc' port='5900'/>
  </devices>
</domain>
```

Note: The only changes made (in italics) are:

The <source dev> field, where the new zpool and zfs filesystem location has been changed to reflect the new iSCSI-based ZFS location.

The <input type>, because this edit fixes a tracking issue with the mouse.

We are not concerned with the name, MAC address and UUI remaining the same. The VNC graphics option is not changed as guest-w2003se-HVM is the only HVM guest running on the system.

5. Define guest-w2003se-HVM from the edited XML backup file:

```
# /usr/bin/virsh define /nfs/dumps/guest-w2003se-HVM.xml
Domain guest-w2003se-HVM defined from /nfs/dumps/guest-w2003se-HVM.xml
```

6. Start and check guest-w2003se-HVM:

```
# /usr/bin/virsh start guest-w2003se-HVM
Domain guest-w2003se-HVM started

# /usr/bin/virsh list --all
Id Name                 State
----------------------------------
 0 Domain-0             running
40 guest-nvb85-PV       blocked
41 guest-w2003se-HVM    blocked
```

7. Confirm the VNC details:

```
# /usr/bin/virsh vncdisplay guest-w2003se-HVM
:0
```

Using an xterm where the correct DISPLAY variable is set run:

```
# /usr/bin/vncviewer :0 &
```

Summary

In summary, using Figure 8, we have built the following:

Step 1: Configured NFS and iSCSI via ZFS on a Solaris Community Edition based server

Step 2: Captured the guest data and transfer to the NFS server

Step 3: Configured a Sun xVM hypervisor server to access both NFS and iSCSI

Step 4: Extracted the guest data and recreated on iSCSI Luns

Step 5: Defined and started the two guests from the XML descriptor files

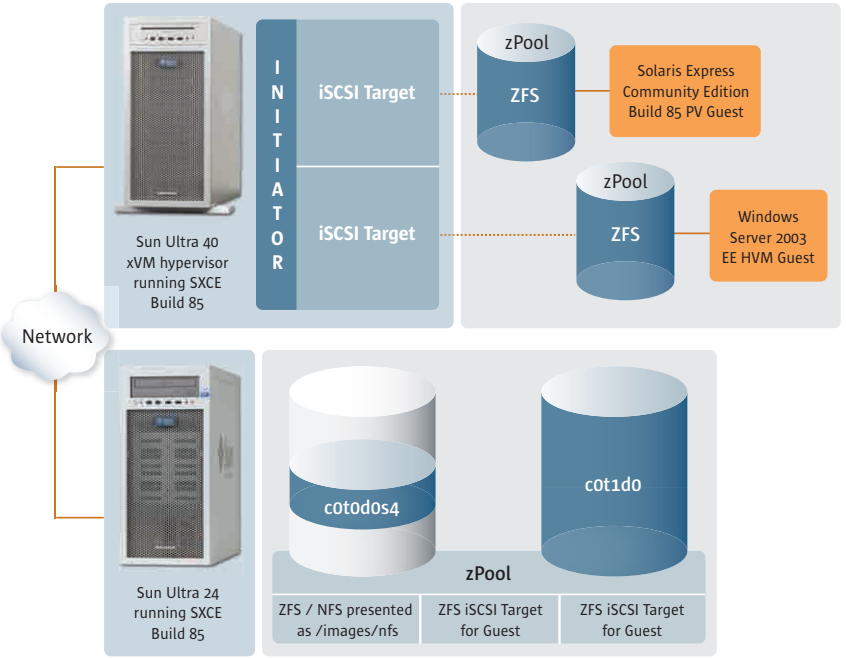

Figure 8.

Appendix A: iscsi.bash Script

```
#!/bin/bash
# Script to help gather configuration detail
# run on the iSCSI target hosting server
# The output commands can then be run on an iSCSI initiator host
echo ""
/usr/sbin/iscsitadm list target
echo ""
echo "Run these commands on both initiators:"
echo ""
        for target in `/usr/sbin/iscsitadm list target | grep "^Target" | awk -
F: '{print $2}'`
        do
                echo /usr/sbin/iscsiadm add static-config `/usr/sbin/iscsitadm
list target -v $target | grep "iSCSI Name" | awk '{print $NF}'`,$1
        done
echo ""
```

developers.sun.com

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web sun.com