This page last changed on Feb 06, 2009 by elena_levashova.

Searching Blueprints

Click here for tips on how to improve your search.

Topics

- Flash/Open Storage
- CMT
- Security
- Virtualization
- HPC
- Web
- MySQL
- Storage
- x64
- Datacenter

Archives

- 2009
- 2008
- 2007
- 2006
- 2005
- 2004
- Pre-2004
- BigAdmin
- Books
- Reference Architectures
- Sun BluePrints Blog

Additional Resources

BigAdmin

docs.sun.com

RSS

Sun BluePrints RSS Feed

by Dominic Kay
September, 2008

Introduction

A flexible, low-cost alternative to Fibre Channel interfaces and dedicated storage area network (SAN) hardware, the Internet Small Computer System (iSCSI) standard is an Internet Protocol (IP)-based storage networking standard that is typically implemented over Ethernet technology.

Creating a configuration consisting of devices with iSCSI interfaces that provides state-of-the-art reliability, availability, and serviceability (RAS) that is easy to administer has long proven difficult. Now, these low-cost devices can be combined with the Solaris ZFS file system to take advantage of file system robustness and ease of administration.

This article provides a streamlined set of instructions for provisioning an iSCSI array with the Solaris ZFS file system. Because the iscsi.d software used to implement iSCSI targets in the Solaris 10 Operating System (OS) is different in both design and syntax from the COMSTAR framework used in the OpenSolaris operating system, this article provides instructions for both environments. These instructions also discuss creating Solaris ZFS file systems on clients/initiators. However, users can choose to use a distributed file system, such as the Network File System (NFS) or Common Internet File System (CIFS), for presenting file-level storage on clients.

Terminology

Primarily used for presenting block-based storage, the iSCSI interface defines targets and initiators in its protocol.

- Target – A storage device, or a device that behaves like a storage device, such as a storage array or a server running the iSCSI protocol
- Initiator – A client that initiates requests to read and write data stored on the target device

Building and Using an iSCSI LUN in the Solaris 10 OS

The following sections describe the basic steps for building and sharing iSCSI storage in the Solaris 10 OS.

Creation

1. Check to see that the iSCSI Service Management Facility (SMF) service is enabled and online on the server/target.

```
# svcadm enable iscsitgt
# svcs iscsitgt

STATE          STIME    FMRI
online         14:20:11 svc:/system/
iscsitgt:default
```

2. Create a Solaris ZFS pool. The example below creates a pool named mypool.

```
# zpool create -f mypool c0d0s3
# zpool list

NAME    SIZE   USED  AVAIL
CAP  HEALTH  ALTROOT
```

```
mypool   163M    94K   163M
0%  ONLINE  -
```

3. Create a volume within the pool. The example below creates a volume named myvol.

```
# zfs create -V 100m mypool/
myvol
```

4. Use the `zfs list` command to verify the size of the newly created pool and volume.

```
# zfs list

NAME          USED  AVAIL  REFER
MOUNTPOINT
mypool        144K   131M    18K
 /mypool
mypool/myvol    30K   131M
30K  -
```

5. Create the iSCSI target using the volume as backing storage. For Solaris ZFS volumes, use the `zfs set` command and specify the volume name mypool/myvol for the target. For other backing storage, such as a file or disk slice, use the `iscsitadm` command.

```
# zfs set shareiscsi=on mypool/
myvol
 # iscsitadm create target -b /dev/
zvol/rdsk/mypool/myvol mytarget
```

6. List the target name using the `iscsitadm` command.

```
# iscsitadm list target -v

Target: mypool/myvol
iSCSI Name:
iqn.1986-03.com.sun:02:ba474bd6-13d3-
caf9-ea29-89cf96e01df6
Connections: 0
ACL list:
TPGT list:
    LUN information:
        LUN: 0
            GUID: 0
            VID: SUN
            PID: SOLARIS
```

```
            Type: disk
            Size: 100M
            Backing store: /dev/zvol/
rdsk/mypool/myvol
            Status: online
```

7. Attach the target on the client/ initiator using the `iscsitadm` command. The string passed to the `iscsitadm` command consists of the iSCSI name followed by a comma, the IP address of the server/target followed by a colon, and the port number 3260 (default).

```
 # iscsiadm add static-config
 iqn.1986-03.com.sun:02:ba474bd6-13d3-
caf9-
ea29-89cf96e01df6,192.168.0.19:3260
 # iscsiadm list static-config

 Static Configuration Target:
 iqn.1986-03.com.sun:02:ba474bd6-13d3-
caf9-
ea29-89cf96e01df6,192.168.0.19:3260
```

8. Import the LUN.

```
 # iscsiadm modify discovery -s
 enable
 # devfsadm -i iscsi -v
```

9. Use the `format(1M)` utility to label the disk for use once the import is complete.

```
 # format

 Searching for disks...done
 AVAILABLE DISK SELECTIONS:
 0. c0d0 <DEFAULT cyl 4092 alt 2
 hd 128 sec 32>
 /pci@0,0/pci-ide@7,1/ide@0/
cmdk@0,0
 1.
 c3t0100000C29DA9B4C00002A004885F38Fd0
 <DEFAULT cyl 98 alt 2 hd 64 sec
 32>
 /scsi_vhci/
disk@g0100000c29da9b4c00002a004885f38f
```

10. Select the iSCSI disk, run the `fdisk` program, and label the disk.

```
Specify disk (enter its number): 1
selecting
c3t0100000C29DA9B4C00002A004885F38Fd0
[disk formatted]
FORMAT MENU:
        disk       - select a disk
        type       - select (define) a
disk type
        partition  - select (define) a
partition table
        current    - describe the
current disk
        format     - format and
analyze the disk
        fdisk      - run the fdisk
program
        repair     - repair a defective
sector
        label      - write label to the
disk
        analyze    - surface analysis
        defect     - defect list
management
        backup     - search for
backup labels
        verify     - read and display
labels
        save       - save new disk/
partition definitions
        inquiry    - show vendor,
product and revision
        volname    - set 8-character
volume name
        !<cmd>     - execute <cmd>
then return
        quit
format> p
WARNING - This disk may be in use
by an application that has
         modified the fdisk table.
Ensure that this disk is
         not currently in use before
proceeding to use fdisk.
format> fdisk
No fdisk table exists. The default
partition for the disk is:
a 100% SOLARIS System partition
Type 'y' to accept the default
partition, otherwise type 'n' to edit
the
partition table.
y
format> p
PARTITION MENU:
        0      - change `0' partition
        1      - change `1' partition
        2      - change `2' partition
        3      - change `3' partition
        4      - change `4' partition
        5      - change `5' partition
        6      - change `6' partition
        7      - change `7' partition
        select - select a predefined
table
```

```
        modify - modify a predefined
partition table
        name   - name the current
table
        print  - display the current
table
        label  - write partition map
and label to the disk
        !<cmd> - execute <cmd>,
then return
        quit
partition> p
Current partition table (original):
Total disk cylinders available: 97 +
2 (reserved cylinders)

Part      Tag    Flag     Cylinders
Size            Blocks
  0 unassigned    wm       0
0         (0/0/0)       0
  1 unassigned    wm       0
0         (0/0/0)       0
  2     backup    wu       0 - 96
97.00MB    (97/0/0) 198656
  3 unassigned    wm       0
0         (0/0/0)       0
  4 unassigned    wm       0
0         (0/0/0)       0
  5 unassigned    wm       0
0         (0/0/0)       0
  6 unassigned    wm       0
0         (0/0/0)       0
  7 unassigned    wm       0
0         (0/0/0)       0
  8       boot    wu       0 -  0
1.00MB    (1/0/0)    2048
  9 unassigned    wm       0
0         (0/0/0)       0
partition> label
Ready to label disk, continue? y
```

The new disk is ready for use by any application that requires block-based storage, such as a database application. Alternatively, a Solaris ZFS pool can be created on the disk, along with a Solaris ZFS file system.

1. Create a Solaris ZFS pool on the new disk.

```
# zpool create iscsipool
c3t0100000C29DA9B4C00002A004885F38Fd0
# zpool list

NAME        SIZE   USED  AVAIL
CAP  HEALTH  ALTROOT
iscsipool    87M  95.5K  86.9M
0%  ONLINE  -
```

2. Create a Solaris ZFS file system.

```
# zfs create iscsipool/myfs
# zfs list

NAME             USED  AVAIL
REFER  MOUNTPOINT
iscsipool        134K  54.9M    18K
 /iscsipool
iscsipool/myfs    18K  54.9M    18K
 /iscsipool/myfs
```

Removing a File System and Solaris ZFS Pool on the Solaris 10 OS

The following steps describe how to remove a file system and Solaris ZFS pool.

1. Remove the file system and pool on the client/initiator.

```
# zpool destroy -f iscsipool
```

2. Remove the LUN. The string to pass to the `iscsiadm` command can be retrieved using the `iscsiadm list static-config` command.

```
# iscsiadm remove static-config
iqn.1986-03.com.sun:02:ba474bd6-13d3-
caf9-
ea29-89cf96e01df6,192.168.0.19:3260
# iscsiadm modify discovery -s
disable
```

3. Delete the target using the `iscsitadm` command or the `zfs set` command.

```
# iscsitadm delete target -u 0
mypool/myvol
# zfs set shareiscsi=off mypool/
myvol
```

4. Remove the file system and Solaris ZFS pool.

```
# zpool destroy -f mypool
```

Building and Using an iSCSI LUN in the OpenSolaris Operating System

The following sections describe the basic steps for building and sharing iSCSI storage in the OpenSolaris operating system.

Creation

1. Download the COMSTAR framework from http://www.opensolaris.org/os/project/comstar and install the software packages using the directions located on the OpenSolaris project site.

2. Disable the Solaris 10 OS iSCSI framework on the server/target, if needed.

```
# svcs iscsitgt

STATE          STIME    FMRI
online         14:20:11 svc:/system/
iscsitgt:default

# svcadm disable iscsitgt
```

3. Create a ZFS pool on the server/target. The example below creates a pool named mypool.

```
# zpool create -f mypool c0d0s6
# zpool list

NAME     SIZE   USED  AVAIL
CAP  HEALTH  ALTROOT
mypool   294M   111K   294M
0%  ONLINE  -
```

4. Create a volume within the pool. The example below creates a volume named myvol.

```
# zfs create -V 200m mypool/
myvol
# zfs list

NAME          USED  AVAIL  REFER
MOUNTPOINT
mypool        200M  61.9M    18K
 /mypool
mypool/myvol   200M   262M
16K  -
```

5. Create a target using the `itadm` command.

```
# itadm create-target
```

6. List the target using the `itadm` command.

```
# itadm list-target

1 Target(s) found
Target:
iqn.1986-03.com.sun:02:97c1caa8-5732-
ec53-b7a2-a722a946fead
```

7. Create a LUN using the newly created volume.

```
# sbdadm create-lu /dev/zvol/dsk/
mypool/myvol

Created the following LU:
            {{GUID
DATA SIZE          SOURCE
--------------------------------
---------------- --------------
600144f0000c291ac068486e146a0001
     209649664       /dev/zvol/dsk/
mypool/myvol
```

8. Create a view of the LUN and add it into the framework. List the view to validate its entry into the framework.

```
# stmfadm add-view
600144f0000c291ac068486e146a0001
```

```
# stmfadm list-view -l
600144f0000c291ac068486e146a0001

View Entry: 0
    Host group   : All
    Target group : All
    LUN          : 0
```

9. Retrieve the name of the target for use in a later step.

```
# itadm list-target

1 Target(s) found
Target:
iqn.1986-03.com.sun:02:97c1caa8-5732-
ec53-b7a2-a722a946fead
```

10. Retrieve and note the IP address of the server/initiator for use in a later step.

```
# ifconfig pcn0

pcn0:
flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS>
mtu 1500 index 2
        inet 192.168.0.19 netmask
ffffff00 broadcast 192.168.0.255
        ether 0:c:29:1a:c0:68}}
```

11. Add the target on the client/ initiator add the target. The string passed to the `iscsiadm` command consists of the IQN number followed by a comma, the IP address of the server/target followed by a colon, and the port number 3260 (default). List the results for verification.

```
# iscsiadm add static-config
iqn.1986-03.com.sun:02:97c1caa8-5732-
ec53-b7a2-
a722a946fead,192.168.0.19:3260
# iscsiadm list static-config

Static Configuration Target:
iqn.1986-03.com.sun:02:97c1caa8-5732-
ec53-b7a2-
a722a946fead,192.168.0.19:3260
```

12. Turn on discovery.

```
# *iscsiadm modify discovery -s
enable
```

13. Verify the results.

```
# iscsiadm list discovery

Discovery:
        Static: enabled
        Send Targets: disabled
        iSNS: disabled
```

14. Bring the disk online, if it has not already come online.

```
# devfsadm -i iscsi -v
```

15. Use the `fdisk` utility to label the disk:

```
# format -e

Searching for disks...done
AVAILABLE DISK SELECTIONS:
       0. c0d0 <DEFAULT cyl 4092 alt
2 hd 128 sec 32>
          /pci@0,0/pci-ide@7,1/
ide@0/cmdk@0,0
       1.
c3t600144F0000C291AC068486E33620001d0
<DEFAULT cyl 197 alt 2 hd 64 sec
32>
          /scsi_vhci/
disk@g600144f0000c291ac068486e33620001
Specify disk (enter its number): 1
selecting
c3t600144F0000C291AC068486E33620001d0
[disk formatted]
FORMAT MENU:
        disk       - select a disk
        type       - select (define) a
disk type
        partition  - select (define) a
partition table
        current    - describe the
current disk
        format     - format and
analyze the disk
        fdisk      - run the fdisk
program
        repair     - repair a defective
sector
        label      - write label to the
disk
        analyze    - surface analysis
```

```
        defect     - defect list
management
        backup     - search for
backup labels
        verify     - read and display
labels
        save       - save new disk/
partition definitions
        inquiry    - show vendor,
product and revision
        scsi       - independent SCSI
mode selects
        cache      - enable, disable or
query SCSI disk cache
        volname    - set 8-character
volume name
        !<cmd>     - execute <cmd>,
then return
        quit
format> p
WARNING - This disk may be in
use by an application that has
         modified the fdisk table.
Ensure that this disk is
         not currently in use before
proceeding to use fdisk.
format>
format> fdisk
No fdisk table exists. The default
partition for the disk is:
  a 100% SOLARIS System partition
Type 'y' to accept the default
partition,  otherwise type 'n' to
edit the
format> y
format> q
```

The new disk is ready for use by any application that requires block-based storage, such as a database application. Alternatively, a Solaris ZFS pool can be created on the disk, along with a Solaris ZFS file system.

1. Create a ZFS pool and verify its existence.

```
# zpool create iscsipool
c3t600144F0000C291AC068486E33620001d0
# zpool status

  pool: iscsipool
 state: ONLINE
 scrub: none requested
config:
        NAME
  STATE     READ WRITE CKSUM
        iscsipool
 ONLINE       0     0     0

c3t600144F0000C291AC068486E33620001d0
 ONLINE       0     0     0
```

```
errors: No known data errors
```

16. Create the file system and verify its existence.

```
# zfs create iscsipool/myfs
# zfs list

NAME             USED  AVAIL
REFER  MOUNTPOINT
iscsipool        134K   155M    18K
 /iscsipool
iscsipool/myfs    18K   155M    18K
 /iscsipool/myfs
```

Removing a File System and ZFS Pool on the OpenSolaris Operating System

1. Get the iSCSI IQN name associated with the LUN on the client/initiator.

```
# LUN=`iscsiadm list static-config
 | awk '{print $4}'`
# echo $LUN

 iqn.1986-03.com.sun:02:769ec050-6f0a-4317-
f53a-
bc1d9493da91,192.168.0.19:3260
```

2. Remove the LUN on the client/ initiator and verify its deletion.

```
# iscsiadm remove static-config
$LUN
# iscsiadm list static-config
```

3. Notice that the disk is also gone.

```
# format < /dev/null

Searching for disks...done
AVAILABLE DISK SELECTIONS:
       0. c0d0 <DEFAULT cyl 4092 alt
2 hd 128 sec 32>
```

```
            /pci@0,0/pci-ide@7,1/
ide@0/cmdk@0,0
 Specify disk (enter its number):
```

4. Find and delete the target on the server/target, and verify its deletion.

```
# itadm list-target

1 Target(s) found
Target:
iqn.1986-03.com.sun:02:769ec050-6f0a-4317-
f53a-bc1d9493da91

# itadm delete-target
iqn.1986-03.com.sun:02:769ec050-6f0a-4317-
f53a-bc1d9493da91
# itadm list-target


0 Target(s) found
```

5. Find the name and view of the LUN.

```
# stmfadm list-lu

LU Name:
600144F0000C291AC0684864ADF30001

# stmfadm list-view -l
600144F0000C291AC0684864ADF30001

View Entry: 0
    Host group   : All
    Target group : All
    LUN          : 0
```

6. Remove the view.

```
# stmfadm remove-view -l
600144F0000C291AC0684864ADF30001
0
```

7. Find the name of the LUN.

```
# sbdadm list-lu

Found 1 LU(s)
            GUID                    DATA
SIZE          SOURCE
--------------------------------
------------------- ----------------
```

```
600144f0000c291ac0684864adf30001
    314507264        /dev/dsk/
c0d0s6
```

8. Remove the LUN.

```
# sbdadm delete-lu
600144f0000c291ac0684864adf30001
```

9. Remove the file system, pool, or other backing store, if desired.

For More Information

Solaris ZFS manual: http://docs.sun.com

The man pages for the Solaris ZFS file system: http://docs.sun.com/app/docs/doc/819-2240/zfs-1m
http://docs.sun.com/app/docs/doc/819-2240/zpool-1m

ZFS Learning Center: http://www.sun.com/software/solaris/zfs_learning_center.jsp

OpenSolaris ZFS Community: http://www.opensolaris.org/os/community/zfs/
The OpenSolaris ZFS manual can be found here.

ZFS Wiki: http://www.solarisinternals.com//wiki/index.php?title=Category:ZFS

OpenSolaris advocacy group presentations: http://www.opensolaris.org/os/community/advocacy/os-presentations/

Opensolaris mail alias archive: http://www.opensolaris.org/jive/forum.jspa?forumID=80

Search for ZFS blogs at http://blogs.sun.com

About the Author

Dominic is a Senior Product Marketing Manager working in Solaris Storage software. He has been at Sun about 10 years working in storage, software and performance. Prior to

that he built and led technical teams at Dell and HSBC.

Acknowledgments

I would like to thank those who took the time to review this article: Paul Eggleton, Deirdre Straughan, Jesse Butler and Peter Dunlap.

Rate this publication	
Choices	Your Vote
Great	
Good	
Fair	
Poor	