Getting to know the Solaris iSCSI stack

Ryan Matteson
matty91@gmail.com
http://prefetch.net

Presentation overview

- Tonight we are going to cover the iSCSI protocol, and how to use the iSCSI stack that comes with Solaris
- I plan to split my 60-minutes into two parts. The first part will provide an overview of the protocol, and the second half will show how to use the Solaris initiator and target

What is iSCSI?

- iSCSI is a protocol that allows SCSI commands to be transmitted over a TCP/IP network
- The protocol was published in draft form by the IETF in 2001, and has since been ratified and documented in RFC 3720

Why should I consider using iSCSI?

- Reduce costs
  - iSCSI uses TCP/IP, so low cost Ethernet adaptors are all that are required to utilize the protocol (no need to buy expensive FC HBAs)
  - Ethernet switches can be used for both storage and public network traffic (this reduces the need to buy expensive fibre channel switches)
- Existing tools (e.g., wireshark, snoop) can be used to debug storage problems (who has the money for fibre channel analyzers?!?)
- Existing IP management and monitoring frameworks can be used with iSCSI networks

Does Solaris support iSCSI?

- It sure does!!!!!
- Solaris 10 GA shipped with an iSCSI initiator that has been certified by a number of tier one storage vendors (and it has been improved in subsequent Solaris 10 updates)
- Nevada (the development build leading up to the next version of Solaris) contains an iSCSI target, which will be available in the next Solaris 10 update (7/07?)
- An iSNS server is on the horizon, and should be available in Nevada in the near future

Terminology

SCSI

- SCSI is a set of standards for connecting and transferring data between computers and peripherals*
- A SCSI initiator is the endpoint responsible for initiating SCSI operations (i.e., the client)
- A SCSI target is the endpoint responsible for processing SCSI commands from an initiator (i.e., the server)
- SCSI Commands are sent between endpoints in CDBs (command descriptor blocks)

* This definition comes from the fine folks at wikipedia.org

iSCSI

- iSCSI is a protocol for sending SCSI commands over a TCP/IP network
- iSCSI, like SCSI, uses the term initiator to describe the endpoint that initiates SCSI operations (i.e. the client), and the term target to describe the endpoint that accepts and processes SCSI commands from one or more initiators (i.e. the server)

iSCSI naming

- A device (e.g., server, appliance, etc.) capable of acting as an iSCSI initiator or target is referred to as a network element
- Each network element can contain one or more iSCSI nodes (e.g., initiators and targets), and each node is assigned a unique iSCSI name
- iSCSI names can be in one of two formats: IQN or EUI
- IQN names contain a date string, the domain of a naming authority, a unique string to identify the node, and are prefixed by the string "iqn."
  - IQN address: iqn.1986-03.com.sun:01:0003ba0e0795.4455571f
- EUI names consist of 16 hexadecimal digits prefixed by the string "eui." (EUI addresses resemble fibre channel WWNs)
  - EUI address: eui.02004567A425678D

iSCSI portals

- All iSCSI network elements contain one or more portals, which are the entry and exit points for iSCSI communications
- Each portal consists of an IP address and TCP port
  - The default port for iSCSI targets is 3260
  - iSCSI initiators use ephemeral port ranges
- Portals can be grouped into portal groups to limit the set of interfaces that are allowed to participate in iSCSI communications, and to allow sessions to span portals
- NB: Portal addresses and iSCSI node names are not tied together in any shape or form. This allows a portals network address to change, while preserving the unique node name.

iSCSI connections and sessions

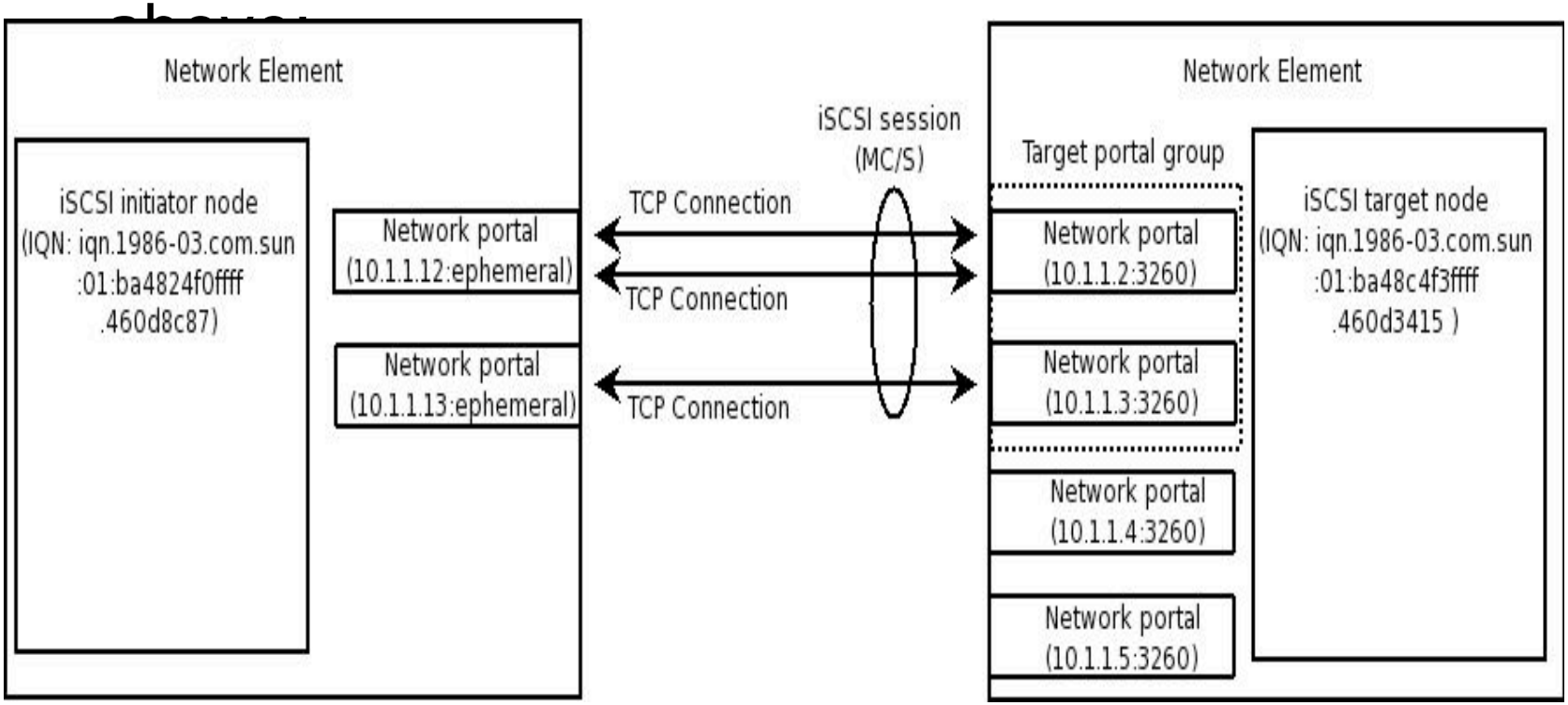
- iSCSI uses the TCP protocol to ensure reliable delivery of data
- Initiators can create one or more TCP connections to a target
- Each TCP connection between an initiator and target is associated with a “session,” which is used to link logical connections together, and to ensure the ordered delivery of SCSI commands
- Initiators can create one or more sessions to a target, and each session can contain one or more TCP connections (multiple connections per session is often abbreviated MC/S)
- iSCSI sessions are made unique on an initiator by combining the node name with a unique initiator session ID (ISID), and on the target by combing the node name with the target session id (TSID)

iSCSI discovery methods

- Discovery allows an initiator to find one or more targets and portals in a network
- Solaris supports three discovery methods:
  - Static discovery
  - SendTargets
  - iSNS

Piecing it all together

- The following diagram is an attempt to visually represent the concepts discussed above:



Using iSCSI

Configuring the iSCSI target

- The iSCSI target is configured with the iscsitadm utility, which takes a command as it's primary argument, a subcommand as it's secondary argument, and one or more options to control what the command and subcommands are applied to
- The "-?" option can be used with iscsitadm to get context sensitive help:

$ iscsitadm show stats -?

iscsitadm show stats [OPTIONS] [<local-target>]

OPTIONS:

-v, --verbose

-I, --interval <seconds>

-N, --count <number>

For more information, please see iscsitadm(1M)

Steps to configure the target

1. Create a base directory
2. Configure a backing store
3. Create a target
4. Optionally configure aliases, ACLs, CHAP authentication and IPSEC security associations
5. Verify the target configuration

Create the base directory

- The base directory is used to store the iSCSI target configuration data, and needs to be defined prior to using the iSCSI target for the first time
- You can create a base directory with the iscistadm utility:

$ iscsitadm modify admin -d /etc/iscsitgt

Configure a backing store

- The backing store contains the physical storage that is exported as a target
- The Solaris target supports several types of backing stores:
  - Flat files
  - Physical devices
  - SVM meta devices
  - ZFS volumes (zvols for short)
- To create a backing store from a ZFS volume, the zfs utility can be run with the create subcommand, the create zvol option (“-V”), the size of the zvol to create, and the name to associate with the zvol:

$ zfs create -V 9g stripedpool/iscsivol000

Creating a target

- Once a backing store has been created, it can be exported as an iSCSI target with the iscsitadm "create" command, the "target" subcommand, and by specifying the backing store type to use:

**$ iscsitadm create target -b **
/dev/zvol/dsk/stripedpool/iscsivol000
host1-tgt0

Add an ACL to a target

- Access control lists (ACLs) can be used to limit the node names that are allowed to access a target
- To ease administration of ACLs, the target allows you to associate an alias with a node name (you can retrieve the node name of a Solaris initiator by running the iscsiadm utility with the “list” command, and “initiator-node” subcommand):

  **$ iscsitadm create initiator -n iqn.1986- **
  03.com.sun:01:0003ba0e0795.4455571f host1

- After an alias is created, it can be added to a target’s ACL by passing the alias to the “target” subcommands “-l” option:

  $ iscsitadm modify target -l host1 host1-tgt0

Verify the target configuration

- To verify the configuration of a target, iscsitadm can be run with “list” command, the “target” subcommand and optionally the “-v” (verbose output) option:

```
$ iscsitadm list target -v
Target: host1-tgt0
  iSCSI Name: iqn.1986-03.com.sun:02:cd7c60ee-f015-e2e2-d5e7-cf529127d20f.host1-tgt0
  Connections: 0
  ACL list:
    Initiator: iqn.1986-03.com.sun:01:0003ba0e0795.4455571f
 TPGT list:
 LUN information:
    LUN: 0
       GUID: 0
      VID: SUN
      PID: SOLARIS
      Type: disk
      Size: 9.0G
      Backing store: /dev/zvol/dsk/stripedpool/iscsivol000
      Status: online
```

Target demo

Configuring the Solaris initiator

- The iSCSI initiator is configured with the iscsiadm utility, which takes a command as it's primary argument, a subcommand as it's secondary argument, and one or more options to control what the command and subcommands are applied to
- The "-?" option can be used with iscsiadm to get context sensitive help:

$ iscsiadm list target -?

iscsiadm list target [OPTIONS] [<target-name ...>]

OPTIONS:

-v, --verbose

-S, --scsi-target

For more information, please see iscsiadm(1M)

Steps to configure the initiator

1. Configure a discovery method
2. Verify the targets
3. Initialize and use the new targets

Configuring a discovery method

- The iscsiadm utility can be used to configure a discovery method and the discovery parameters
- Configuring static discovery:

  $ scsiadm modify discovery --static enable

  $ iscsiadm add static-config iqn.1999-08.com.array:sn.01234567,192.168.1.3:3260
- Configuring SendTargets discovery:

  $ iscsiadm modify discovery --sendtargets enable

  $ iscsiadm add discovery-address 192.168.1.13:3260
- Configuring iSNS discovery:

  $ iscsiadm modify discovery --isns enable

  $ iscsiadm add isns-server 192.168.1.13:3205

Verifying the targets

- Once a discovery method is configured, the iscsiadm utility can be used to list the targets that were discovered:

```
$ iscsiadm list target -vS
 Target: iqn.1986-03.com.sun:02:cd7c60ee-f015-e2e2-d5e7-cf529127d20f.host1-tgt0
       Alias: host1-tgt0
       TPGT: 1
        ISID: 4000002a0000

       < ….. >

       LUN: 0
           Vendor:  SUN Product: SOLARIS OS Device Name:
           /dev/rdsk/c1t010000CBC18475E900002A00457C908Ad0s2
```

Initialize and use targets

- Prior to using newly discovered targets, the devfsadm utility needs to be run to create device entries:

  $ devfsadm -Cv -i iscsi

- Once the device nodes are created, the format utility can be used to label the new targets, and your favorite file system management tool (e.g., mkfs, zpool, etc) can be used to convert the target(s) into file systems:

  $ zpool create iscsipool c4t0100080020A76DF400002A00458BFE9Ad0

- NB: The LUN component of an iSCSI device contains a GUID instead of a LUN id, which is comprised of the Target Portal MAC address and a timestamp

Initiator demo

Security

- Since iSCSI transmits data over IP networks, it is ***imperative*** to protect iSCSI traffic from eavesdroppers
- This can be accomplished with a layered security approach that includes one or more of the following:
  - Dedicated storage networks
  - Client ACLs
  - Auditing
  - IPSEC and header digests
  - Port security on Ethernet switches
- Security is a whole presentation in and of itself. Please see the references if you are interested in learning more about iSCSI security

Performance

- iSCSI performance can be quite good, especially if you follow a few basic rules
  - Use Enterprise class NICs (they make a ***HUGE*** difference)
  - Enable jumbo frames on storage ports
  - Use layer-2 link aggregation and IPMP to boost throughput
  - Ensure that you are using the performance guidance listed in bug #6457694 on opensolaris.org
  - Increase send and receive buffers, disable the nagle algorithm and make sure TCP window scaling is working correctly
- Ttcp and netperf are awesome tools for benchmarking network throughput, and measuring the impact of a given network tunable
- As with security, performance is a complete presentation in and of itself. Please see the references if your interested in learning more about tuning iSCSI communications for maximum

Conclusion

- The opensolaris storage community is leading the pack when it comes to iSCSI
- All of the technology discussed in this presentation is opensource, comes with a $0 price tag, and can be downloaded from opensolaris.org / sun.com

References

- Cuddletech (all things iSCSI)
  http://cuddletech.com
- iSCSI Dtrace scripts
  http://www.solarisinternals.com
- iSCSI multipathing
  http://www.sun.com/blueprints/1205/819-3730.pdf
- iSCSI Security
  http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-Dwivedi-update.pdf
- Opensolaris storage community
  http://opensolaris.org/os/community/storage/
- SNIA IP Storage White Paper
  http://www.snia.org/tech_activities/ip_storage/iSCSI_Technical_whitepaper.PDF
- T10 organization
  http://t10.org

Questions?