VMAX ALL FLASH AND VMAX3 ISCSI DEPLOYMENT GUIDE FOR ORACLE DATABASES

EMC® VMAX® Engineering White Paper

ABSTRACT

IP-based connectivity between storage and servers offers a simple, safe, and robust alternative to FC networks. This paper discusses the deployment guidelines of iSCSI with VMAX® All Flash and VMAX3™ for Oracle databases.

May 2016

REDEFINE

EMC WHITE PAPER

EMC²

To learn more about how EMC products, services, and solutions can help solve your business and IT challenges, contact your local representative or authorized reseller, visit www.emc.com, or explore and compare products in the EMC Store

Part Number H15132.1

TABLE OF CONTENTS

EXECUTIVE SUMMARY

With VMAX® All Flash and VMAX3™ iSCSI, support was re-designed to provide customers with greater port connection densities using virtual storage ports, built-in multi-tenancy capabilities using VLAN, and easier isolation using VMAX initiator groups.

Note: This white paper is focused on VMAX All Flash, however the iSCSI implementation on VMAX3 is identical. Throughout this paper, references to VMAX will pertain to both VMAX All Flash and VMAX3.

The use of iSCSI offers many advantages, including:

- High performance and bandwidth due to higher adoption of 10GbE and faster network interfaces. IP-based connectivity can now deliver bandwidth equivalent to or faster than 8 Gb Fibre-Channel (FC) SAN networks for most workloads. For OLTP workloads, iSCSI and FC offer nearly identical performance.
- Benefits of converging storage and network infrastructure in the data center. These benefits include cost savings from maximizing existing network management skills, unifying infrastructure components, and the added simplicity of IP-based connectivity.
- Increased scale, utilizing VMAX virtual storage ports. Often iSCSI deployments can only allocate a single IP address to each storage target port, limiting the deployment scale. VMAX iSCSI targets are designed around virtual storage ports to overcome these limitations.
- Strong security using uni-directional or bi-directional Challenge-Handshake Authentication Protocol (CHAP) authentication.
- Multi-tenancy and network isolation, leveraging VLAN and VMAX host initiator groups. VLANs provide virtual networks so iSCSI traffic can be isolated from other network activity, or other tenants. VMAX host initiator groups are part of the VMAX device masking configuration which allows fast and flexible changes to relationships between host initiators, storage target ports, and storage devices. Only the participating members of a masking view are visible to each other.
- Improved support for lower-cost test/dev environments. As demonstrated in this paper, even when existing databases use FC interfaces, VMAX SnapVX™ can easily create database snapshots that can be accessed using iSCSI, for example, by development or QA groups.

iSCSI for Oracle database deployments on VMAX is easy and offers all the advantages described in this white paper. Oracle databases can use iSCSI as the primary storage protocol, or they can use an iSCSI connection to snapshots of the primary database, even if the primary database uses FC for connectivity.

This white paper provides an overview of iSCSI, describes how to configure it on VMAX and Linux, and provides Oracle workload examples comparing FC and iSCSI.

AUDIENCE

This white paper is intended for database and system administrators, storage administrators, and system architects who are responsible for implementing, managing, and maintaining Oracle Databases in environments with VMAX All Flash and VMAX3 storage systems. Readers should have some familiarity with the iSCSI protocol and EMC VMAX storage arrays.

VMAX ISCSI OVERVIEW

iSCSI is a protocol that uses the Transmission Control Protocol/Internet Protocol (TCP/IP) to transport SCSI commands, enabling the use of the existing TCP/IP networking infrastructure as a Storage Area Network (SAN). As with SCSI over Fibre Channel (FC), iSCSI presents SCSI targets and devices to iSCSI initiators (requesters). Unlike Network Area Storage (NAS), which presents devices at the file level, iSCSI presents block devices across an IP network to your local system. These can be consumed in the same way as any other block storage device.

Primary benefits of iSCSI

With the proliferation of 10GbE networking in the last few years, iSCSI has steadily gained footprint as a deployed storage protocol in datacenters. For datacenters with centralized storage, iSCSI offers customers many benefits. It is comparatively inexpensive and it is based on familiar SCSI and TCP/IP standards. In comparison to FC and Fibre Channel over Ethernet (FCoE) SAN deployments, iSCSI can use less or lower-cost hardware than FC, and may provide a decreased administrative cost, as more IT professionals are familiar with the technology. These factors contribute to lower-cost implementations, especially when various networking technologies can be converged onto the same network infrastructure.

Some of the primary benefits of iSCSI are:

- Makes consolidated storage possible for a wide range of businesses
- Enables cost-effective, scalable, secure, and highly-available SANs
- Leverages existing management skills and network infrastructure
- Delivers performance comparable to Fibre Channel
- Provides interoperability using industry standards
- Offers high availability
- Features no specific distance limitations, though latency requirements often limit the distance to local area networks (LANs)

VMAX iSCSI design

The new VMAX iSCSI target model has been designed from inception to meet customer demands regarding control and isolation of storage resources. The VMAX iSCSI target model accomplishes this through the following key design objectives, characteristics and components.

Design objectives

- Provide control and isolation of VMAX storage resources (storage multi-tenancy)

Key design characteristics

- Separation of iSCSI target from the physical port
- Individual iSCSI targets can have multiple network portals
- Support for VLANs, network namespaces (Network IDs), and routing

Figure 1 shows the VMAX iSCSI design, including the use of VLAN for network isolation, and the separation of storage IP interfaces and target nodes from the physical ports on each director and iSCSI storage module (SE).

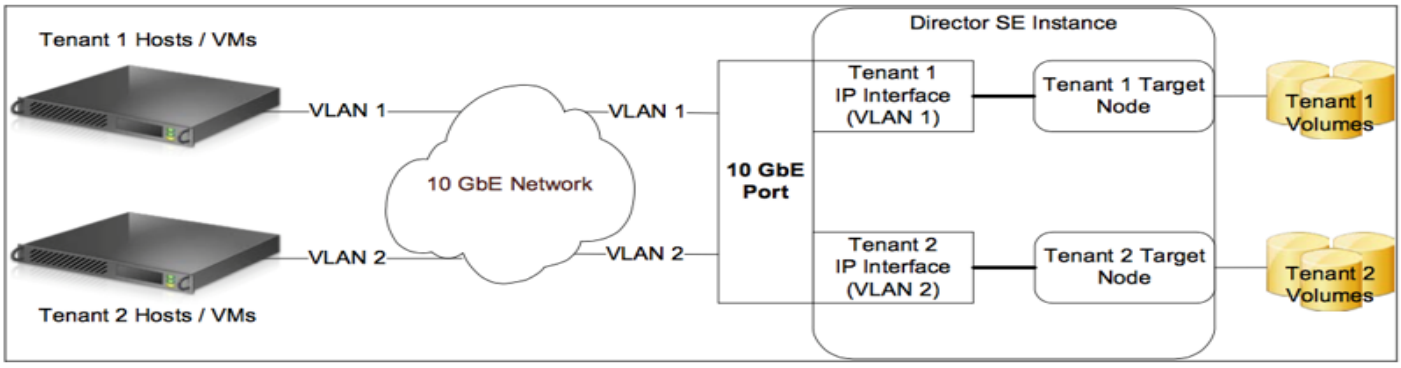

Figure 1. **Core components in a VMAX**

Core iSCSI components

An iSCSI architecture is made up of a set of core components. These components are initiator and target nodes, iSCSI names, IP interfaces, sessions, connections, and security. The following section details each of these components. Figure 2 shows the relationships between iSCSI target node and network portals.

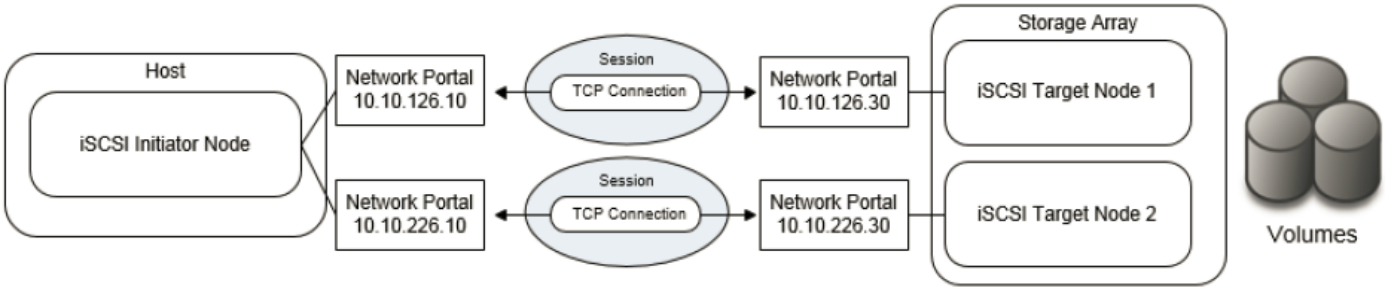

Figure 2. **Core iSCSI components**

iSCSI initiators and target nodes

A storage network consists of two types of equipment: initiator and target nodes. Initiators, such as hosts, are data consumers. Targets, such as disk arrays or tape libraries, are data providers.

- The iSCSI initiator can be implemented either as a driver installed on the host system or within the hardware of an iSCSI HBA, which typically includes TOE (TCP/IP Offload Engine). The host initiates a request to and receives responses from an iSCSI target (storage node).
- iSCSI target nodes expose one or more SCSI LUNs to specific iSCSI initiators. The target node listens and responds to commands from iSCSI initiators on the network. On the enterprise storage level, iSCSI target nodes are logical entities, not tied to a specific physical port.

iSCSI initiators must manage multiple, parallel communication links to multiple targets. Similarly, iSCSI targets must manage multiple, parallel communication links to multiple initiators. Several identifiers exist in iSCSI to make this happen, including iSCSI Name, ISID (iSCSI session identifiers), TSID (target session identifier), CID (iSCSI connection identifier) and iSCSI portals.

iSCSI target node

In VMAX iSCSI implementation iSCSI target nodes are also referred to as storage virtual ports to indicate the separation of a target node from its physical port. Multiple target nodes can be associated with each physical port allowing more scale and flexibility.

iSCSI names

iSCSI target nodes are identified by a unique iSCSI Name. iSCSI names are a human-readable ASCII string and must be unique on a per-namespace (Network ID) basis. iSCSI names would ideally be unique worldwide, but since they are both user- and algorithmically generated, there can be duplicates even on the same array. iSCSI Names are formatted in two different ways:

- Enterprise Unique Identifier (EUI) - example: eui.0123456789ABCDEF
- iSCSI Qualified Name (IQN) - most commonly used naming format
    - Example: iqn.2001-05.com.RedHat:ProdNode1

Note: As IQN formatting is most common, the examples in this paper are all based on IQN.

iSCSI IP interfaces (iSCSI network portals)

iSCSI target nodes are accessed through IP interfaces (also called network portals). iSCSI network portals contain key network configuration information such as:

- IP Address
- Network ID
- VLAN information
- MTU

An iSCSI network portal can only provide access to a single iSCSI target node; however, an iSCSI target node can be accessed through multiple network portals. These portals can be grouped together to form a portal group. Portal groups are identified by a unique portal group tag defined for the iSCSI target node. All portals in a portal group must provide access to the same iSCSI target node.

iSCSI sessions and connections

iSCSI initiator and target nodes communicate by a linkage called an iSCSI session. The session is the vehicle for the transport of the iSCSI packets, or Portal Data Units (PDUs) between the initiators and target nodes. Each session is started by the initiator logging into the iSCSI target. The session between the initiator and target is identified by an iSCSI session ID. Session IDs are not tied to the hardware and can persist across hardware swaps.

Session components are tied together by a TCP/IP connection. The IP addresses and TCP port numbers in the network portals (IP interfaces) define the end points of a connection.

VMAX iSCSI scale limits

The following is a list of the current support limits/scale. Please refer to release notes for up-to-date information.

- Maximum of 64 targets per physical port
- Maximum of 8 network portals (IP interfaces) per target
- Maximum of 512 network IDs (Range is 0–511)
- Maximum of 1024 routing instances per engine

Best practices for iSCSI deployment

Network considerations

Network design is key to making sure iSCSI works properly and delivers the expected performance in any environment. The following are best practice considerations for iSCSI networks:

- 10GbE networks are essential for enterprise production-level iSCSI. Anything less than 10GbE should be relegated to test and development.
- iSCSI should be considered a local-area technology, not a wide-area technology, because of latency issues and security concerns.

- Segregate iSCSI traffic from general traffic by using either separate physical networks or layer-2 VLANs. Best practice is to have a dedicated LAN for iSCSI traffic and not to share the network with other network traffic. Aside from minimizing network congestion, isolating iSCSI traffic on its own physical network or VLAN is considered a must for security as iSCSI traffic is transmitted in an unencrypted format across the LAN.

- Implement jumbo frames (by increasing the default network MTU from 1,500 to 9,000) in order to deliver additional throughput, especially for small block read and write traffic. However, if jumbo frames are to be implemented they require all devices on the network to be jumbo-frame compliant and to have jumbo frames enabled. When implementing jumbo frames, set host and storage MTUs to 9,000 and set switches to higher values such as 9,216 (where possible).

Note: Especially in an Oracle database environment, where database block size is typically 8KB, a 9,000 MTU will be able to transport an Oracle block in a single frame, where a 1,500 MTU will require transmitting multiple packets for each database block read or write I/O operation.

- To minimize latency, avoid routing iSCSI traffic between hosts and storage arrays. Try to keep hops to a minimum. Ideally host and storage should coexist on the same subnet and be not more than one hop away.

Multipathing and availability

The following are iSCSI considerations with regard to multipathing and availability.

- Use either EMC PowerPath or native Linux multipathing (DM-Multipath). It is important that the two do not coexist on the same server as this can cause instability.

- Utilize multipathing software enabled on the host rather than multiple connections per session (MC/S). MC/S is not supported for VMAX targets.

- For Linux systems using device mapper multipath (DM-Multipath), use the "Round Robin (RR)" load balancing policy. Round Robin uses an automatic path selection rotating through all available paths, enabling the distribution of load across the configured paths. This path policy can help improve I/O throughput. For VMAX storage arrays, all paths will be used in the Round Robin policy.

- For Linux systems using DM-Multipath, change “path_grouping_policy” from “failover” to “multibus” in the multipath.conf file. This will allow the load to be balanced over all paths. If one fails, the load will be balanced over the remaining paths. With “failover” only a single path will be used at a time, negating any performance benefit. Ensure that all paths are active using “multipath –l” command. If paths display an “enabled” status, they are in failover mode.

- Use the “Symmetrix Optimized” algorithm for EMC PowerPath software. This is the default policy and means that administrators do not need to change or tweak configuration parameters. PowerPath selects a path for each I/O according to the load balancing and failover policy for that logical device. The best path is chosen according to the algorithm. Due to the propriety design and patents of PowerPath, the exact algorithm for this policy cannot be detailed here.

Resource consumption

The processing of iSCSI traffic consumes initiator- and target-side CPU resources. The following are considerations to understand:

- To minimize CPU consumption of iSCSI traffic, employ iSCSI HBAs with a built in TCP Offload Engine (TOE). TOE HBAs offload the processing of the datalink, network, and transport layers from the CPU and process it on the iSCSI HBA itself.

- VMAX CPU cores are a critical resource when planning for performance. VMAX automatically allocates cores to each emulation, such as FC, iSCSI, SRDF, eNAS, etc. The number of cores allocated for each emulation can easily be listed, such as when using the Solutions Enabler command: ‘symcfg list –dir all’. In certain cases—especially with a low number of VMAX engine count and many emulations—the default core allocation may not take into account specific application I/O bottlenecks. While it is rare, if you suspect that this is a problem, contact your EMC account representative to review the core allocation distribution between emulations.

For more detailed information regarding the VMAX iSCSI architecture, see the technical note: VMAX3 iSCSI Implementation.

Oracle and VMAX iSCSI Tests

Test environment

To demonstrate the performance and replication use cases with iSCSI and FC in an Oracle OLTP environment, we used the following test environment.

VMAX configuration

We created two storage groups with associated masking views. The first masking view contained the FC host initiators, FC ports, and the first set of database devices used for the FC workload. The second masking view contained the iSCSI host initiators, storage target ports, and the second set of database devices used for the iSCSI workload. We confirmed that both iSCSI and FC VMAX emulations on the storage had the same number of CPU cores assigned to each. Both sets of devices were identical, as well as the database which resided on them. We ensured this by using SnapVX to replicate the Oracle database between the FC and iSCSI devices.

Hardware resources

Table 1 describes the hardware components used for the tests.

Table 1. Hardware components used in tests

Device	Quantity	Configuration	Description
EMC VMAX 200K	1	• 2 Engines	VMAX3 array
Servers	1	• 10 core Intel Xeon CPU E5-2680 v2 @ 2.80GHz • 98 GB memory • 2 x 10GbE network NICs 2 x 8Gb FC initiators	Database server
Ethernet Switch	1	• Ethernet switch – 10 GbE/S • Jumbo frames enabled • Separate VLAN for iSCSI traffic	Cisco Nexus 3500

Software resources

Table 2 describes the software components used for the tests.

Table 2. Software components used in tests

Device	Version	Description
VMAX HYPERMAX OS	5977.811.784 (Q1 2016)	Operating environment for VMAX
EMC Solutions Enabler	8.1.0.0	API/CLI
EMC Unisphere®	8.2.0.163	VMAX management GUI
DM-Multipath	0.4.9-77.el7_1.2.x86_64	Native Linux multipathing
Oracle Enterprise Linux	7.1	Operating system for database servers

Oracle Database 12c	Enterprise Edition 12.1.0	Oracle Database software
Oracle Grid Infrastructure 12c	Enterprise Edition 12.1.0	Software support for Oracle ASM

Multipathing

We used native Linux device mapper multipathing (DM-Multipath) to configure multiple I/O paths between the server and the VMAX. DM-Multipath provides a method for organizing the I/O paths logically by creating a single multipath device on top of the underlying devices. See Multipathing and availability

section for more information.

Test 1: iSCSI and FC comparative performance

The goal of this use case was to compare Oracle workload performance by running a set of identical tests: first using FC protocol, and then using iSCSI protocol. By comparing the results we wanted to verify that Oracle was performing similarly with both protocols.

Test workload

For the OLTP tests, we used SLOB 2.3. SLOB is an easy-to-use and free Oracle database benchmark tool with a choice of database capacity, number of workload processes, threads, rate of update, and workload locality of reference.

Three tests were performed for each protocol, with 5 users (workload processes), 10 users, and 15 users. Each such user, or process, ran against a data set of 64GB for a total of 320GB data set size in the 5 users test case, 640GB data set size in the case of 10 users, and 960GB in the case of 15 users. SLOB was set with a 25 percent update rate to simulate an OLTP-type workload and each test was run for 30 minutes, excluding ramp-up time.

Oracle Database 12c was used with ASM, where the database used 16 devices of 128 GB for +DATA ASM disk group, and 8 devices of 8 GB for +REDO ASM disk group. SnapVX was used to create an exact copy of the database. During the tests, the first copy was accessed using FC protocol, and the second copy was accessed using iSCSI protocol. In both cases ASM disk groups were mounted in the exact same way, and the exact same workload was run with each copy.

Note: This configuration was not designed for all-out performance benchmarking, but rather as a means for running identical tests between both FC and iSCSI protocols for comparison. Better performance could be achieved by leveraging stronger servers, Oracle RAC for scale-out and higher concurrency, more host/storage connectivity, and TOE for host CPU offload.

Results collection and metrics

All results produced during the 30 minute runtime of each test case were collected using Oracle AWR report. While Unisphere for VMAX provides excellent performance views and details, it is the database performance that the end user cares about eventually. For this reason we focused on the database metrics instead of storage metrics.

We compared five specific Oracle AWR metrics:

- *Physical read total IOPS*: This metric provides the average number of read I/O operations per second that the Oracle database performed during the test period.
- *Physical write total IOPS*: This metric provides the average number of write I/O operations per second that the Oracle database performed during the test period.

Note: By looking at total read and write IOPS we can combine them to get the total IOPS of the workload, or compare them to get the read/write ratio of the workload.

- *DB File Sequential Read Avg (ms)*: This metric provides the average response time in milliseconds of the database data files read I/Os. With an OLTP workload and all-flash storage array we expect a lot of small-block random read activity (8KB in the case of Oracle) with latency near 1ms when measured from the database.
- *Log File Parallel Write Avg (ms)*: This metric provides the average response time in milliseconds of the database log writer write I/Os. It is always important to keep an eye on the log write latencies because flash media and flash arrays tend to perform very

well for read operations, but vary in their performance for write operations. VMAX acknowledge writes to the host as soon as they registered with its persistent cache and can maintain low latencies for Oracle log writes which are large write I/Os (typically 128KB-512KB).

Test results

Table 3 shows the test results from our SLOB workload runs for each set of tests (5, 10, and 15 users). The results from both iSCSI and FC tests were very similar.

Table 3. iSCSI and FC Oracle workload results

SLOB	AWR			
Slob Configuration	**Physical Read Total IOPS**	**Physical Writes Total IOPS**	**DB File Sequential read Avg (ms)**	**Log File Parallel Write Avg (ms)**
iSCSI 5 users x 1 thread	20,087	5,037	0.72	0.87
iSCSI 10 users x 1 thread	30,127	7,438	0.98	1.14
iSCSI 15 users x 1 thread	36,604	9,164	1.21	1.4
FC 5 users x 1 thread	20,980	5,257	0.69	0.77
FC 10 users x 1 thread	34,658	8,683	0.84	1.08
FC 15 users x 1 thread	36,118	9,040	1.21	1.04

Test conclusion

We concluded that iSCSI is a viable alternative to FC connectivity and can provide similar performance numbers for Oracle workloads.

Test 2: Using iSCSI with database snapshots

The goal of this use case was to show the value of using SnapVX to create a replica of a source Oracle database connected via FC, and connect to the database replica using iSCSI. This use case is relevant in test and development environments, for example. It demonstrates how to quickly and easily connect hosts via IP to replicas of the production database, which is often still using FC for connectivity.

Test steps

Step 1 – Create a snapshot of the source database.

In this test, the "Production" (source) database was not shut down during the snapshot creation as SnapVX inherently creates consistent replicas. A database replica created with storage consistency does not require any database quiescing (for example, Oracle hot-backup mode), and can be simply taken while the source database is active. The database replica will open as a point-in-time copy from the time of the snapshot. When creating test/dev environments, a similar process can take place, where Production operations continue while restartable database snapshots are taken and used as iSCSI targets.

In our test, we used the VMAX Storage Group (SG) 'slob_FC_SG' that contained the Production database devices, and we named the snapshot 'slob_fc_snap'. This can be done using Unisphere for VMAX, or using Solutions Enabler CLI, as shown in the following example:

```
symsnapvx -sg slob_FC_SG -name slob_fc_snap establish
```

As soon as the snapshot was created (it only takes a few seconds), we linked it to a target storage group called 'slob_iSCSI_SG' containing a set of devices matching the Production database devices. For example, using the CLI:

```
symsnapvx -sg slob_FC_SG -snapshot_name slob_fc_snap -lnsg slob_iSCSI_SG link -copy
```

By using 'link –copy' syntax, we created a stand-alone 'gold' copy replica of the production database. This is a common practice in test/dev environments, where a stand-alone copy is created first. From this gold copy, the test/dev/QA teams can create and use additional snapshots.

Note: Since VMAX All Flash delivers very high performance, some customers may prefer to only link the snapshot to the target storage group *without* '-copy', thereby using pointers to a single set of data used by both Production and the target storage groups, and not consuming any additional storage capacity. This is a common practice with All Flash arrays.

Step 2 – Start the iSCSI database replica

To start the iSCSI database replica we first bring up the ASM disk groups contained in the replica, and then open the database.

In the case that the database replicas are opened on a host that already has the same ASM disk groups mounted, or database name used, it is important to first rename the ASM disk groups and database name to be unique. Oracle documentation describes both ASM and database rename options and therefore these are not covered in this paper.

The following commands show an example of starting the ASM disk groups and Oracle database, executed by the Oracle user.

```
export ORACLE_SID=$GRID_SID; export ORACLE_HOME=$GRID_HOME
srvctl start diskgroup -diskgroup DATA
srvctl start diskgroup -diskgroup REDO

export ORACLE_SID=$DB_SID; export ORACLE_HOME=$DB_HOME
srvctl start database -db SLOB
```

Since Oracle Grid Infrastructure and the Oracle database each had a different home directory and binaries, we used the environment variables $GRID_HOME and $GRID_SID to specify the values appropriate to the Grid Infrastructure environment, and $DB_HOME and $DB_SID to specify the values appropriate to the Oracle database environment.

Test conclusion

This test showed that iSCSI can easily be used to connect to database replicas, even if the source database is using FC for connectivity.

Conclusion

As this white paper describes, iSCSI is a viable option for Oracle database connectivity that can simplify the data center infrastructure, converge SAN to IP networks, and utilize the simplicity of IP network management.

VMAX support for iSCSI provides flexibility, scale, high performance, and security. It provides a good alternative to FC SANs and supports mission critical Oracle databases as well as test/dev/QA and other type of database replica usage, even if the source database still uses FC.

Appendices

Appendix I – Configuring iSCSI using Unisphere for VMAX

This section shows the steps needed to configure iSCSI using the Unisphere for VMAX graphical user interface.

Create an iSCSI Target using Unisphere for VMAX

We created multiple iSCSI targets for high availability and scale,. The following example shows the creation of a single iSCSI target.

To create an iSCSI target, follow these steps, as shown in Figure 3:

1. From the System tab select **iSCSI Dashboard**. Then select **Create iSCSI Target**.
2. From the "Create iSCSI Target" dialog box, select the director that your target will be presented from, the target name (can choose custom name or have the system automatically generate one), the network id, and the TCP port (default 3260 for iSCSI).

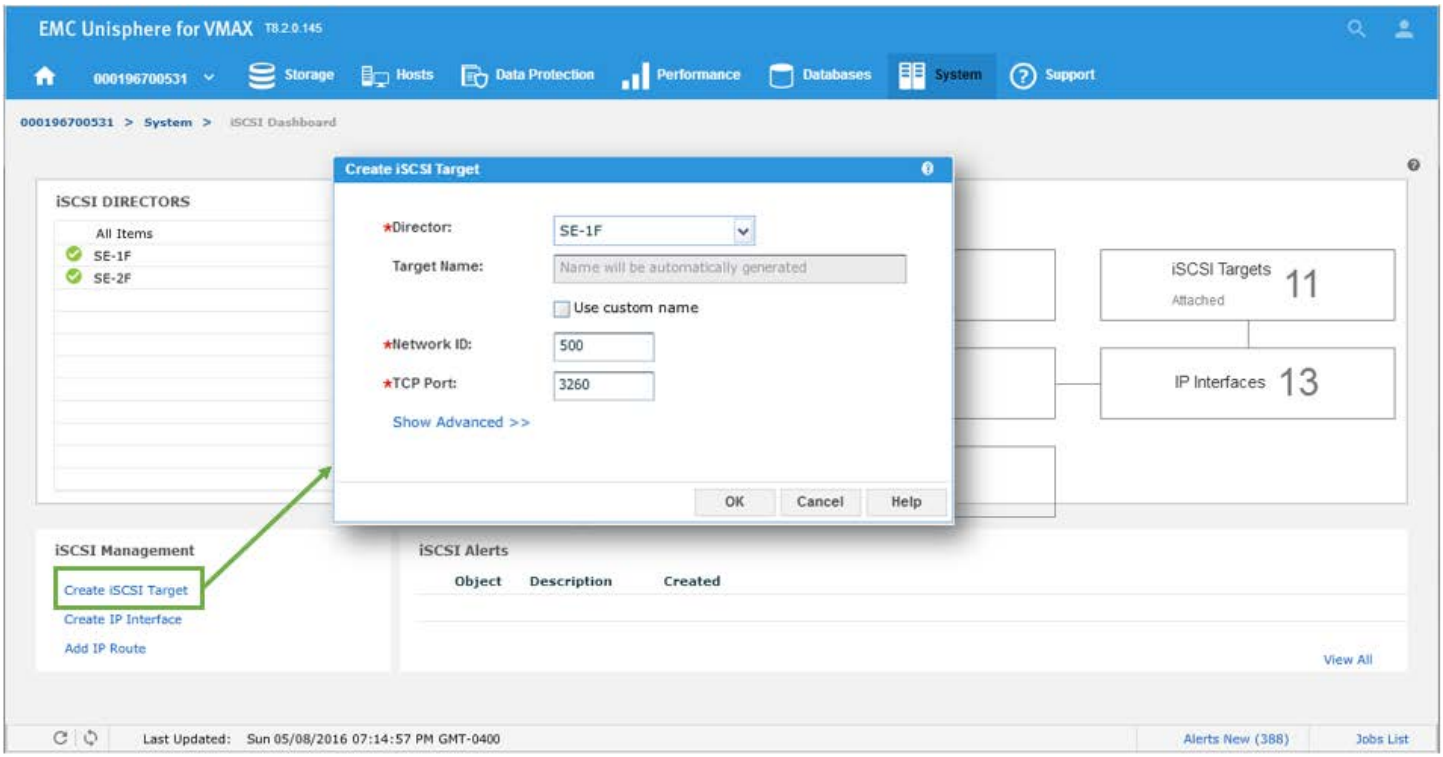

Figure 3. iSCSI Target creation

Enable iSCSI Target using Unisphere for VMAX

To enable iSCSI targets, follow these steps, as shown in Figure 4:

1. Select **iSCSI Targets** from the dashboard window
2. Select the target that was created and choose **Enable** from the menu at the bottom.

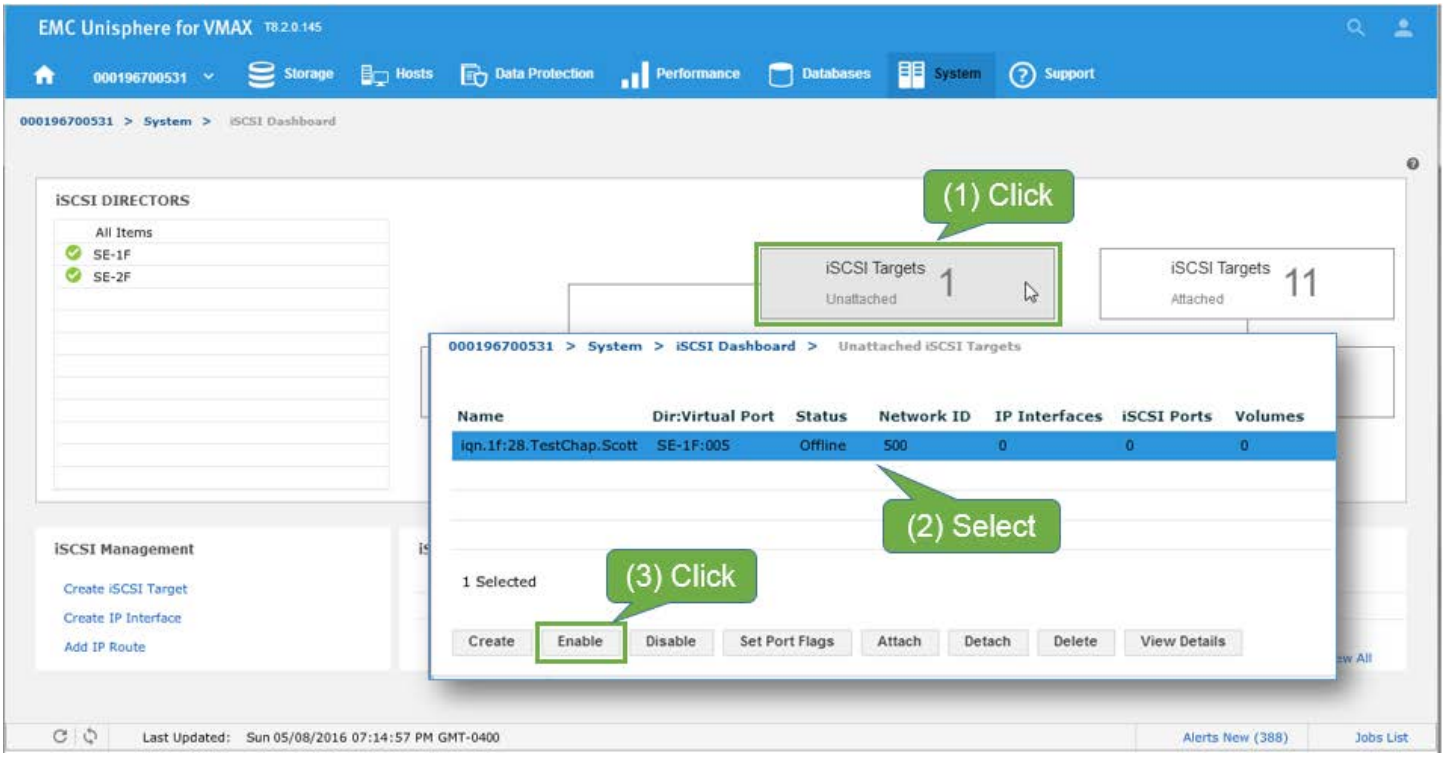

Figure 4. **Enable iSCSI Targets**

Attach New IP Interface to iSCSI Target using Unisphere for VMAX

To attach a new IP interface, follow these steps, as shown in Figure 5:

1. Select the enabled target and choose **Attach** from the bottom menu.
2. From the "Attach IP Interface to iSCSI Target" dialog box, select the director/port your target will be presented from, and choose an IP address, subnet prefix, VLAN ID and max transaction unit (1500 default, 9000 for jumbo frames).
3. Your network ID is pre-selected based upon the selected target's ID. Click **OK.**

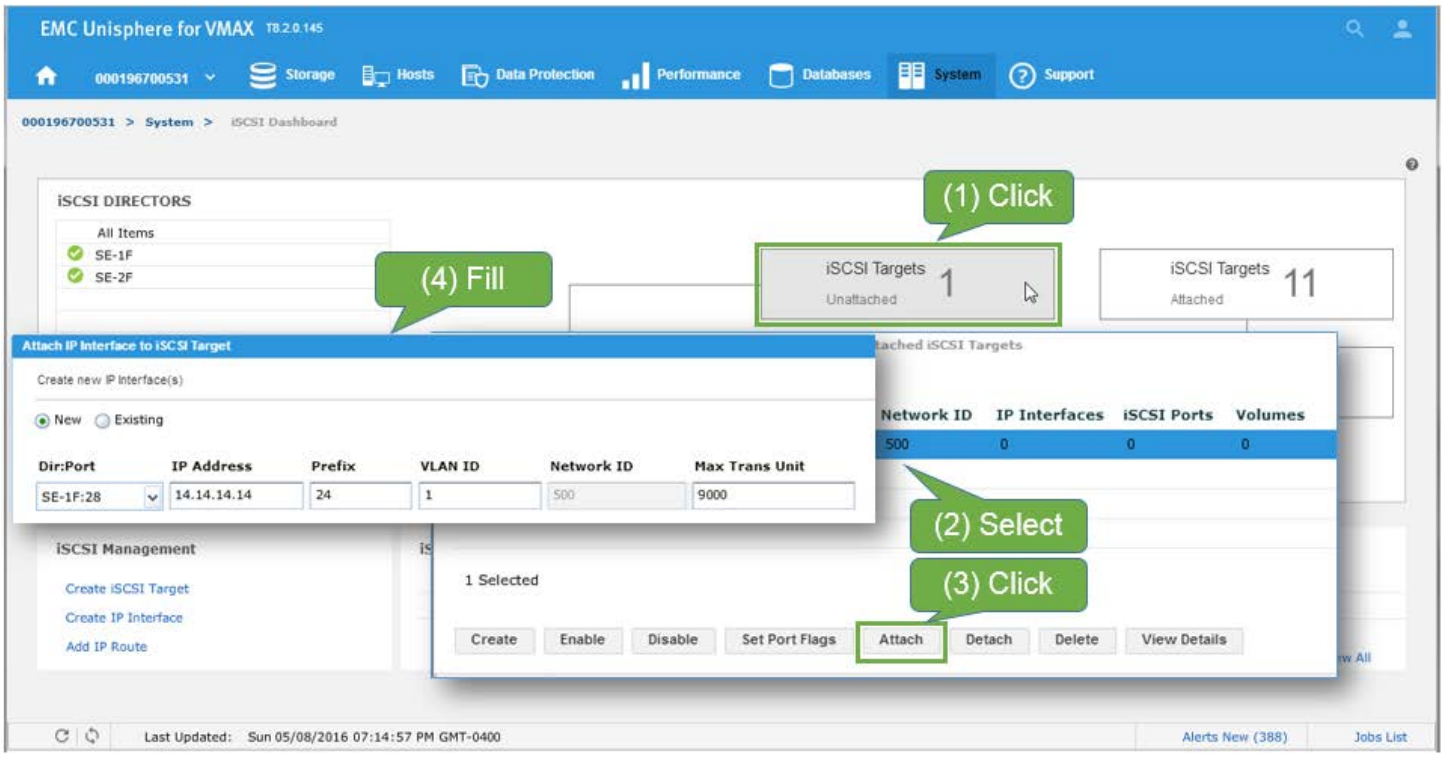

Figure 5. **Attach IP interface to iSCSI Target**

Create Port Group with iSCSI Target using Unisphere for VMAX

To create a port group, follow these steps, as shown in Figure 6:

1. Select **Create Port Group** from the "Port Groups" dashboard (Hosts -> Port Groups).

2. Enter a port group name, check the "iSCSI" radio button and select the newly created target from the list. Click **OK.**

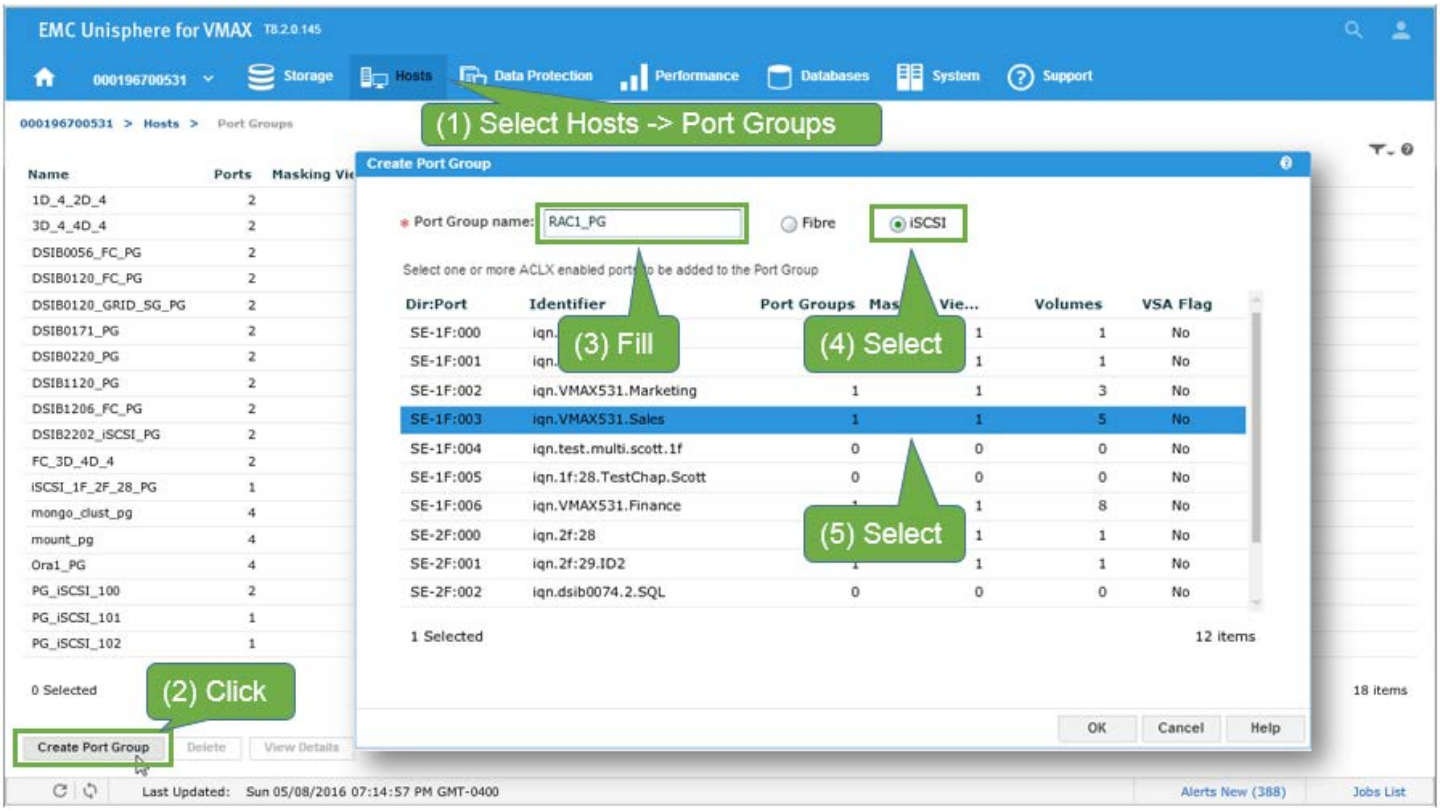

Figure 6. **Create Port Group**

Create Masking View using Unisphere for VMAX

To create a masking view, follow these steps, as shown in Figure 6Figure 7:

1. Select the masking view dashboard (Hosts -> Masking View).
2. Click **Create Masking View**
3. Enter masking view name
4. Select the iSCSI initiator host, port group and associated storage group. Click **OK.**

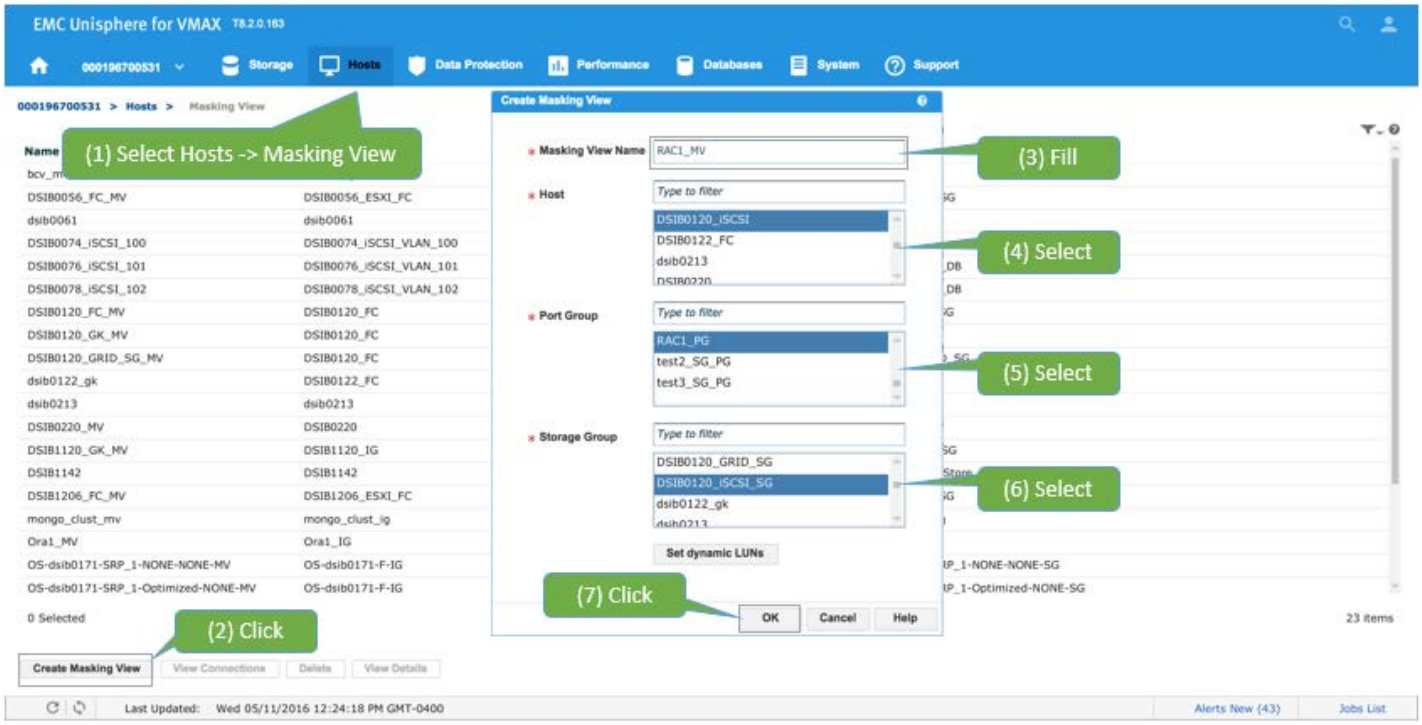

Figure 7. **Create a masking view**

Conclusion

We have successfully created a VMAX iSCSI target and port group that is discoverable from hosts that:

- Are on the same subnet/VLAN as the IP interface associated with the target.
- Belong to a masking view with the port group that contains the newly created target.

Note: Typically multiple iSCSI ports would be added to a port group for scale and availability. The port group can then be added into a masking view so the associated devices in the storage group can be made visible to the host.

Appendix II – Configuring iSCSI using Solutions Enabler CLI

This section will show the steps needed to configure VMAX iSCSI using Solutions Enabler Command Line Interface (CLI).

Create an iSCSI Target

From the storage management host, use the following command to create the iSCSI target.

```
# symconfigure -sid 531 -cmd "create iscsi_tgt dir 1f, iqn=iqn.1f:28.test1, network_id=1;" commit -nop
```

Enable iSCSI Target using SYMCLI

From the storage management host, use the following command to enable the iSCSI target.

Note: iSCSI targets are disabled by default.

Note: The "-iscsi_port" value is the target virtual port not the physical port. To get the value of the virtual port associated with the target run the following: symcfg -sid 531 list -se all -iscsi_tgt -iqn iqn.1f:28.test1

```
# symcfg -sid 531 -se 1f -iscsi_port 5 online
```

Create New IP Interface using SYMCLI

From the storage management host. use the following command to create a new IP interface.

```
# symconfigure -sid 531 -cmd "create ip_interface dir 1f port 28, ip_address=14.14.14.14, ip_prefix=24,
network_id=1, vlanid=1;" commit -nop
```

Attach New IP Interface to iSCSI Target using SYMCLI

From the storage management host, use the following command to attach an IP interface to a target.

```
# symconfigure -sid 531 -cmd "attach ip_interface ip_address=14.14.14.14 to iscsi_tgt iqn=iqn.1f:28.test1;"
commit -nop
```

Create Port Group with iSCSI Target using SYMCLI

From the storage management host, use the following command to create a port group with an iSCSI target.

```
# symaccess -sid 531 create -name test1_PG -type port -iqn iqn.1f:28.test1
```

Create Masking View using SYMCLI

From the storage management host, use the following command to create the masking view

```
# symaccess -sid 531 create view -name RAC1_MV -sg DSIB0120_iSCSI_SG -pg RAC1_PG -ig DSIB0120_iSCSI
```

Conclusion

We have successfully created a VMAX iSCSI target and port group that is discoverable from hosts that:

- Are on the same subnet/VLAN as the IP interface associated with the target
- Belong to a masking view with the port group that contains the newly created target.

Note: Typically multiple iSCSI ports would be added to a port group for scale and availability. The port group can then be added into a masking view so the associated devices in the storage group can be made visible to the host.

Appendix III – Other useful CLI commands for managing VMAX iSCSI storage

This section will provide a list of useful Solutions Enabler CLI commands specific to VMAX iSCSI. Please refer to the EMC® Solutions Enabler CLI Command Reference available on support.emc.com support.emc.com for more info.

- List iSCSI target details

  Displays detailed iSCSI target information.

```
# symcfg -sid 531 list -se all -iscsi_tgt -detail
```

- Disable iSCSI target

  Note: The "-iscsi_port" value is the target virtual port not the physical port. To get the value of the virtual port associated with the target run the following: symcfg -sid 531 list -se all -iscsi_tgt -iqn iqn.1f:28.test1

```
# symcfg -sid 531 -se 1f -iscsi_port 5 offline
```

- Rename iSCSI target

```
# symconfigure -sid 531 -cmd "rename iscsi_tgt iqn=iqn.1f:28.test1 to new_iqn=iqn.1f:28.test1.RENAME;"
commit -nop
```

- Delete iSCSI target

```
# symconfigure -sid 531 -cmd "delete iscsi_tgt iqn=iqn.1f:28.test1;" commit -nop
```

- List iSCSI ports

  Displays detailed iSCSI port information.

```
# symcfg -sid 531 list -se all -port -detail
```

- List IP Interfaces

  Lists all configured IP interfaces on the array.

```
# symcfg -sid 531 list –ip -se all
```

- Modify IP Interface

  Modify the network ID, and IP of an existing IP interface. Only IP interfaces not attached to an iSCSI target can be modified.

```
# symconfigure -sid 531 -cmd "modify ip_interface dir 1f, ip_address=14.14.14.14, network_id=1,
new_network_id=2, new_ip_address=15.15.15.15, ip_prefix=24, mtu=9000;" commit -nop
```

- Detach IP interface from iSCSI target

```
# symconfigure -sid 531 -cmd "detach ip_interface ip_address=14.14.14.14 from iscsi_tgt
iqn=iqn.1f:28.test1;" commit -nop
```

- Delete IP interface

  Delete an existing IP interface.

```
# symconfigure -sid 531 -cmd "delete ip_interface dir 1f, ip_address=15.15.15.15, network_id=2;" commit -nop
```

- Add IP Route

  Add a static IP route on a given director emulation.

```
# symconfigure -sid 531 -cmd "add ip_route dir 1f, ip_address=10.10.10.0, ip_prefix=24, gateway=10.10.9.1,
network_id=1;" commit -nop
```

- Remove IP Route

  Remove a static IP route on a given director emulation.

```
# symconfigure -sid 531 -cmd "remove ip_route dir 1f, ip_address=10.10.10.0, network_id=1;" commit -nop
```

- List CHAP

  List CHAP security records.

```
# symaccess –sid 531 list chap -v
```

- Enable Uni-directional CHAP

  Enable one-way CHAP on an initiator.

```
# symaccess -sid 531 -iscsi iqn.1988-12.com.oracle:28525e1f5755 set chap -cred iqn.1988-
12.com.oracle:28525e1f5755 -secret TargetCHAPSecret
```

- Enable Bi-directional CHAP

  Enable two-way CHAP on a target.

```
symaccess -sid 531 –iscsi_dirport 1f:5 set chap -cred iqn.1f:28.test1 -secret InitiatorCHAPSecret
```

- Disable Uni-directional CHAP

  Disable one-way CHAP on an initiator.

```
# symaccess -sid 531 -iscsi iqn.1988-12.com.oracle:28525e1f5755 disable chap
```

- Disable Bi-directional CHAP

  Disable two-way CHAP on a target.

```
# symaccess -sid 531 -iscsi_dirport 1f:5 delete chap
```

Appendix IV – Connect to VMAX iSCSI Target from a Linux host

This section will show how to discover and connect to a VMAX iSCSI target. In these examples, we will be using a Red Hat Linux server. The process is relatively the same for all supported Linux distributions. The exception is for the installation of the "open-iSCSI" package only. Please see your specific Linux distribution documentation for more information regarding package installation. Target discovery will be done without CHAP for this example. The host has IP connections for both management (via 1 Gb) and SAN (via 10 Gb private network).

Assumptions

- Host has been cabled to VMAX iSCSI port/s.
- Masking view, which includes the host initiator, port group, and storage group have been created. (See Appendix I or II)
- Host is able to ping the iSCSI target ports.

Steps

1. Install Open-iSCSI Package

   The iSCSI package provides the server daemon for the iSCSI protocol, as well as the utility programs used to manage it.

```
# yum install iscsi-initiator-utils
```

2. Start the iSCSI Service

```
# /etc/init.d/iscsi start
```

3. Scan for VMAX iSCSI targets

   The following command will issue a "send targets" command to the VMAX iSCSI array. You must know the IP portal address for your iSCSI target to discover (see Appendix I or II).

```
# iscsiadm –m discovery –t sendtargets –p 10.10.10.20
```

   VMAX iSCSI target is shown with its associated IP portal.

```
# 10.10.10.20:3280,0 iqn.1f:28
```

4. Log in to VMAX iSCSI targets

   The following command will log in to the VMAX iSCSI target. This is required to access the VMAX iSCSI devices associated with the target.

```
# iscsiadm –m node –l –T iqn.1f:28 –p 10.10.10.20
```