Red Hat Enterprise Linux 6.3 NIC Optimization and Best Practices with EqualLogic SANs

A Dell EqualLogic Reference Architecture

Dell Storage Engineering
January 2014

Revisions

Date	Description
January 2014	Initial release
July 2014	Added Broadcom iSOE VLAN setup content

Table of contents

Acknowledgements

This best practice white paper was produced by the following members of the Dell Storage team:

Engineering: Clay Cooper

Technical Marketing: Guy Westbrook

Editing: Margaret Boeneke

Additional contributors: Mike Kosacek, Richard Golasky, Pavel Viltres

Feedback

We encourage readers of this publication to provide feedback on the quality and usefulness of this information by sending an email to SISfeedback@Dell.com.

SISfeedback@Dell.com

Executive summary

This reference architecture explores the configuration options available for optimizing Dell™ EqualLogic™ PS Series SAN performance using the Broadcom® BCM57810 or Intel® X520 10 GbE network adapters and Red Hat® Enterprise Linux® (RHEL) 6.3 on a Dell™ PowerEdge™ 12th generation server. Recommended OS and NIC configurations are given based on the results of SAN performance testing.

1 Introduction

Dell EqualLogic PS Series arrays provide a storage solution that delivers the benefits of consolidated networked storage in a self-managing iSCSI storage area network (SAN) that is affordable and easy to use, regardless of scale.

In every iSCSI SAN environment, there are numerous configuration options at the storage host which can affect overall SAN performance. These effects can vary based on the size and available bandwidth of the SAN, the host/storage port ratio, the amount of network congestion, the I/O workload profile and the overall utilization of system resources at the storage host. One setting might greatly improve SAN performance for a large block sequential workload yet have an insignificant or slightly negative effect on a small block random workload. Another setting might improve SAN performance at the expense of host CPU utilization.

This technical paper quantifies the effect on iSCSI throughput and IOPS of several configuration options within the Broadcom and Intel 10 GbE network adapter properties and the RHEL 6.3 TCP stack using three common SAN I/O workloads. It also takes into account the value of certain settings in congested network environments and when host CPU utilization is high. From the results, recommended configurations for an EqualLogic PS Series SAN are given for each tested NIC type.

In order to focus on the pure SAN performance benefits of the tested configuration options, Data Center Bridging (DCB) and Broadcom Switch Independent Partitioning, also known as NIC Partitioning (NPAR) were not enabled.

Note: The performance data in this paper is presented relative to baseline configurations and is not intended to express maximum performance or benchmark results. Actual I/O workload, host to array port ratios, and other factors may also affect performance.

1.1 Audience

This technical white paper is for storage administrators, SAN system designers, storage consultants, or anyone who is tasked with configuring a host server as an iSCSI initiator to EqualLogic PS Series storage for use in a production SAN. It is assumed that all readers have experience in designing and/or administering a shared storage solution. Also, there are some assumptions made in terms of familiarity with all current Ethernet standards as defined by the Institute of Electrical and Electronic Engineers (IEEE) as well as TCP/IP and iSCSI standards as defined by the Internet Engineering Task Force (IETF).

2 Technical overview

Section 2 explains the components of an EqualLogic iSCSI SAN and the configuration options available for optimizing SAN I/O performance.

2.1 EqualLogic SAN environment

iSCSI SAN traffic takes place over an Ethernet network and consists of iSCSI protocol communication among the PS Series array members and the iSCSI initiators of the storage hosts. The Broadcom BCM57810 NetXtreme® II and the Intel X520 10 GbE network adapters were used as the SAN interface adapters during this project.

The Broadcom BCM57810 network adapter features iSCSI Offload Engine (iSOE) technology which offloads processing of iSCSI protocol communication to the network adapter. When using iSOE mode, the network adapter becomes a host bus adapter (HBA) and the native software iSCSI initiator is not used. This is as opposed to non-iSOE mode in which the network adapter functions as a traditional NIC and works with the native software iSCSI initiator. This paper refers to the non-iSOE mode of operation as software initiator mode.

The following three initiator modes of operation were tested:

1. Broadcom BCM57810 software initiator mode
2. Broadcom BCM57810 iSOE mode
3. Intel X520 software initiator mode

Appendix B provides a detailed list of the tested configuration options, the default values of the network adapters, and of the RHEL 6.3 TCP stack. It also provides instructions for making configuration changes.

2.2 Host configuration options

The following section defines the available configuration options evaluated in this paper.

2.2.1 Jumbo frames

Jumbo frames enable Ethernet frames with payloads greater than 1500 bytes. In environments where large packets make up the majority of traffic and additional latency can be tolerated, jumbo frames can reduce CPU utilization and improve bandwidth efficiency.

2.2.2 Flow control

Flow control, defined by the IEEE 802.x standard, is a link-level mechanism that enables the adapter to respond to or generate flow control (PAUSE) frames. Flow control helps to prevent network congestion and packet loss and is enabled by default.

2.2.3 Receive and transmit buffers

The receive and transmit NIC ring buffers, also referred to as the driver queue layer, provide a layer of independence and buffering between the NIC device driver and the networking layer protocols. The ring buffer is an area of memory in which outgoing packets can be stored prior to transmission by the device driver and incoming packets can be stored prior to processing by the TCP/IP stack. Queuing packets ensures that the driver and the TCP/IP stack are continuously processing packets, thus enhancing throughput. Maximizing buffer allocation is particularly important on a server with heavy CPU utilization and can also be beneficial during times of network congestion.

However, a large ring buffer can increase latency. Since the ring buffer is processed using the first-in, first-out (FIFO) method, a particular packet entering the ring buffer waits for the entire ring buffer to clear before being transmitted or received.

The ring buffer size is specified by the number of allowed packet descriptors, for example 4096, and not by the byte size of the buffer. This means that the actual memory usage of the buffer could be as high as the number of descriptors multiplied by the maximum transmission unit (MTU). For a transmit buffer with 4096 descriptors and an MTU of 9216 bytes that is 36 MB of system memory.

2.2.4 TCP/IP offload engine

TCP/IP offload engine (TOE) offloads the processing of the entire TCP/IP stack to the network adapter and is an available feature on the Broadcom 57810 network adapter. However, there is currently no support for TOE in RHEL. See the link below for a detailed explanation of why the Linux community has chosen not to integrate TOE support into the Linux kernel.

http://www.linuxfoundation.org/collaborate/workgroups/networking/toe

2.2.5 iSCSI Offload Engine

iSOE offloads the processing of the entire iSCSI stack to the network adapter and is available on the Broadcom 57810 network adapter. To use the iSOE adapter, it must be configured using the Linux

iscsiadm utility. Also, the Host Integration Toolkit must be configured to use the Linux Broadcom iSOE driver (bnx2i) as the iSCSI initiator. For more detailed instructions see Appendix B.

2.2.6 TCP checksum offload

TCP checksum offload enables the adapter to verify received packet checksums and compute transmitted packet checksums. This can improve TCP performance, reduce CPU utilization, and is enabled by default. iSCSI connection instability and increased array packet retransmission have been observed with TCP checksum offload disabled; therefore it is recommended that this feature remain enabled.

2.2.7 TCP segmentation offload

TCP segmentation offload (TSO) enables the adapter to offload from the OS the task of segmenting TCP packets into valid Ethernet frames. Because the adapter hardware is able to complete data segmentation much faster than operating system software, this feature may improve transmission performance while using fewer CPU resources. It is enabled by default on both Broadcom and Intel adapters.

One thing to note is that TSO can increase the memory usage of the transmit ring buffer by allowing packets of up to the IPv4 maximum of 64 KB to enter. For a transmit buffer with 4096 descriptors this would mean up to 256 MB of system memory used.

TSO can also increase latency while outgoing segments are bundled together prior to placement in the transmit ring buffer.

2.2.8 Large receive offload

Large receive offload (LRO) aggregates incoming TCP packets from a single stream into a single larger packet for processing by the TCP/IP stack. It is enabled by default on both Broadcom and Intel adapters.

Like TSO, LRO can lower CPU utilization at the expense of an increase in latency and of the system memory usage of the ring buffer.

2.2.9 Scatter / gather for direct memory access

Scatter / gather, also known as Vectored I/O, allows the network adapter to read from and write to non-contiguous areas during direct memory access (DMA). The benefit to performance is greater with larger blocks of data. Scatter / gather also enables copy avoidance by separating the header from the payload, allowing an application to access the payload without copying its location in memory.

Note that disabling scatter / gather also disables TSO.

2.2.10 TCP receive window scaling

The TCP receive window is the amount of unacknowledged data that the receiver is willing to accept from the sender. A certain amount of in-transit data is required in order for actual TCP throughput to approach the available network bandwidth. The amount of in-transmit data required to achieve maximum throughput can be approximated by calculating the bandwidth delay product (BDP). To calculate BDP,

multiply the available bandwidth by the network latency. Network latency can be determined by reading the average roundtrip time (RTT) from a simple ping test between the host and the storage.

The TCP receiver passes the receive window size to the sender using a field in the TCP header. In the original specification, the TCP header field limited the receive window size to 64 K. When TCP receive window scaling is enabled, the receiver can advertise a receive window size of greater than 64 K, allowing the sender to increase the amount of unacknowledged data being sent. It is enabled by default in RHEL 6.3.

2.2.11 TCP congestion control

The analogue to the TCP receive window is the sender's congestion window. A TCP sender will not send more unacknowledged data than the congestion window specifies, even if the receive window size is greater. As transmitted packets are acknowledged, the TCP congestion control algorithm will gradually increase the size of the congestion window until the maximum TCP send buffer size has been reached or the receiver fails to acknowledge sent packets. The default congestion control algorithm in RHEL 6.3 is Cubic.

2.2.12 TCP low latency

When TCP low latency is enabled, the RHEL 6.3 TCP stack chooses to minimize latency whenever possible, even at the expense of higher throughput. By default it is disabled.

2.2.13 Delayed ACK algorithm

The delayed ACK algorithm is a technique to improve TCP performance by combining multiple ACK responses into a single response. This algorithm has the potential to interact negatively with a TCP sender using Nagle's algorithm, since Nagle's algorithm delays data transmission until a TCP ACK is received.

The RHEL 6.3 TCP stack enables the delayed ACK algorithm by default and there is no global method for disabling it. It can only be disabled on a per-TCP socket basis by the application which opens the socket, however the Linux TCP/IP stack may choose to re-enable it as I/O conditions change. The default delayed ACK timeout in RHEL 6.3 is 40 ms.

2.2.14 Nagle's algorithm

Nagle's algorithm is a technique to improve TCP performance by buffering output in the absence of an ACK response until a packet's worth of output has been reached. This algorithm has the potential to interact negatively with a TCP receiver using the delayed ACK algorithm, since the delayed ACK algorithm may delay sending an ACK under certain conditions up to 500 ms.

The RHEL 6.3 TCP stack enables Nagle's algorithm by default and there is no global method for disabling it. It can only be disabled on a per-TCP socket basis by the application which opens the socket. To achieve a result similar to disabling Nagle's algorithm, enable TCP low latency.

3 Test configurations and methodology

The following section addresses the reasoning behind the test environment design and details the SAN configurations. Performance testing methodology, test case sequence, and results analysis are also explained.

3.1 Simplified SAN

Every effort was made to simplify and optimize the test configurations so that the performance effects of each option could be isolated. The following configuration and design elements helped to achieve this goal.

- Eight 100 GB volumes within a single storage pool, evenly distributed across array members
- An isolated SAN with no LAN traffic
- Load balancing (volume page movement within pools) disabled on the array members
- DCB disabled
- Single Function NIC (no NIC partitioning)
- Host Integration Tools for Linux installed with default MPIO settings

Array member load balancing is recommended for production environments because it can improve SAN performance over time by optimizing volume data location based on I/O patterns. It was disabled for performance testing to maintain consistent test results. It is enabled by default.

Base and congested SAN designs were chosen and are described in the following sections. See Appendix A for more detail about the hardware and software infrastructure.

3.1.1 Base SAN configuration

The first SAN design chosen was a basic SAN with a redundant SAN fabric and an equal number of host and storage ports. Having a 1:1 host/storage port ratio is ideal from a bandwidth perspective. This helped to ensure that optimal I/O rates were achieved during lab testing. Figure 1 shows only the active ports of the PS Series array members.

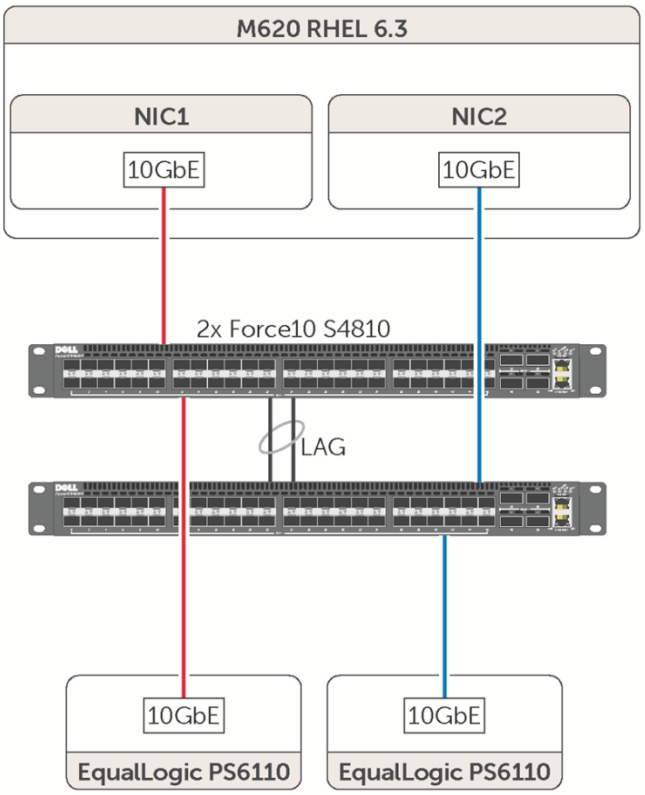

Figure 1 Physical diagram of the base SAN configuration

- Two 10 GbE switches
- Two array members, each with a single port
- A single host with two 10 GbE NIC ports
- A 1:1 storage/host port ratio

3.1.2 Congested SAN configuration

The second SAN design was constructed to mimic a host port experiencing network congestion. In this SAN design, a single host port was oversubscribed by four storage ports for a 4:1 storage/host port ratio. Since only one host port was in use, the SAN fabric was reduced to a single switch. Figure 2 shows only the active ports of the PS Series array members.

A non-redundant SAN fabric is not recommended for a production SAN environment.

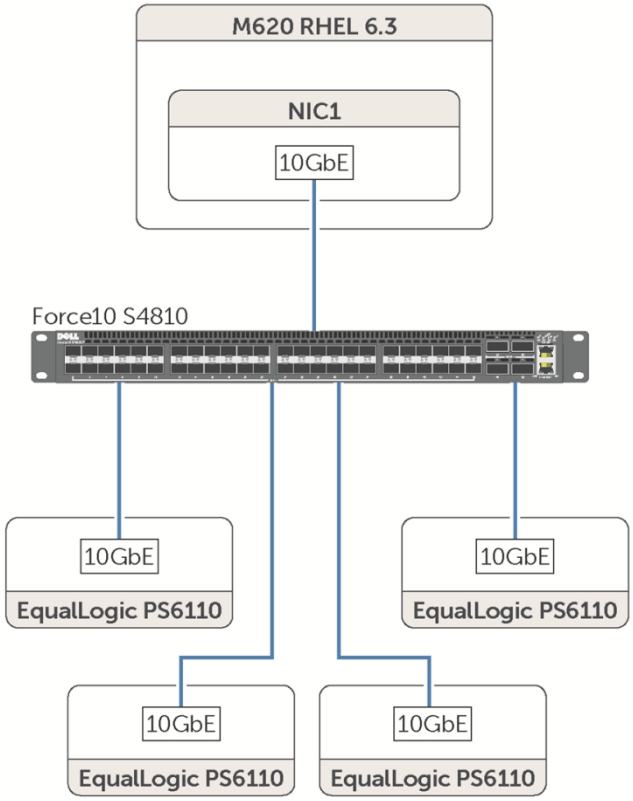

Figure 2 Physical diagram of the congested SAN configuration

- One 10 GbE switch
- Four array members each with one port
- A single host with one 10 GbE NIC
- A 4:1 storage/host port ratio

3.2 I/O execution and evaluation

Prior to each test run, the host was rebooted to confirm configuration changes were in effect. After boot, the even distribution of iSCSI connections across host and storage ports and of active array member ports across SAN switches was confirmed.

The following three vdbench workloads were run:

- 8 KB transfer size, random I/O, 67% read
- 256 KB transfer size, sequential I/O, 100% read
- 256 KB transfer size, sequential I/O, 100% write

For every test case, each vdbench workload was run three times for twenty minute durations and the results were averaged.

Vdbench IOPS results were used to evaluate 8 KB random workload performance. Vdbench throughput results were used to evaluate 256 KB sequential workload performance. Array member retransmission rates and CPU utilization were also examined.

See Appendix C for a list of vdbench parameters.

3.3 Test case sequence

Network adapter and RHEL 6.3 TCP stack options were evaluated using the test cases listed below for each NIC mode:

1. Jumbo frame performance was compared to standard frame performance for each workload.

2. The performance of the baseline configuration was determined. The baseline configuration included the following *non-default* settings at the host:

   - Jumbo frames enabled
   - Flow control auto-negotiation off (if on by default)
   - Maximum network adapter receive and transmit buffers (for software initiator modes)

3. After testing the baseline configuration defined above, each subsequent test case consisted of a *single* option being toggled from its default setting to show its effect relative to the baseline configuration. At the conclusion of each test case, the test option value was returned to default and another option was set to the test case value.

3.3.1 Broadcom BCM57810 software initiator mode test case sequence

The following tables show the test case sequence used to evaluate the effect of tested configuration options for the Broadcom BCM57810 in software initiator mode. **Bold text** indicates the changed value for each test scenario.

Table 1 Baseline test case sequence for Broadcom BCM57810 software initiator mode

Test case	Frame size	Flow Control	Rx / Tx buffers	Other adapter settings	RHEL 6.3 TCP stack setting	Comments
1	Standard	Autoneg / on	Default	Default	Default	Default configuration
2	**Jumbo**	Autoneg / on	Default	Default	Default	Jumbo frames enabled
3	Jumbo	**Forced / on**	**Maximum**	Default	Default	Baseline configuration to evaluate all subsequent settings

Table 2 Test case sequence for Broadcom BCM57810 software initiator mode to evaluate other options

Test case	Frame size	Flow Control	Rx / Tx buffers	Other adapter settings	RHEL 6.3 TCP stack setting	Comments
4	Jumbo	Forced / on	Maximum	Default	**Receive Window Scaling disabled**	
5	Jumbo	Forced / on	Maximum	Default	**Congestion control = reno**	
6	Jumbo	Forced / on	Maximum	Default	**TCP low latency enabled**	
7	Jumbo	Forced / on	Maximum	**TCP / Generic Segmentation Offload disabled**	Default	
8	Jumbo	Forced / on	Maximum	**Scatter Gather (DMA) disabled**	Default	Disabling Scatter Gather turns off TSO also
9	Jumbo	Forced / on	Maximum	**Large Receive Offload disabled**	Default	

3.3.2 Broadcom BCM57810 iSOE mode test case sequence

The following table shows the test case sequence used to evaluate the effect of tested configuration options for the Broadcom BCM57810 in iSOE mode. **Bold text** indicates the changed value for each test scenario.

As can be seen in the table below, when in iSOE mode the Broadcom 57810 has a limited set of adapter options. Also, in iSOE mode the RHEL 6.3 TCP stack options have no effect since the entire iSCSI and TCP stack are offloaded to the adapter.

Table 3 Test case sequence for Broadcom BCM57810 iSOE mode

Test case	Frame size	Flow Control	Rx / Tx buffers	Other adapter settings	Windows Server TCP stack	Comments
1	Standard	N/A	N/A	N/A	N/A	Default configuration
2	**Jumbo**	N/A	N/A	N/A	N/A	Jumbo frames enabled

Bold text indicates the changed value for each test scenario.

3.3.3 Intel X520 software initiator mode test case sequence

The following table shows the test case sequence used to evaluate the effect of tested configuration options for the Intel X520 in software initiator mode. **Bold text** indicates the changed value for each test scenario.

Table 4 Baseline test case sequence for Intel X520

Test case	Frame size	Flow Control	Rx / Tx buffers	Other adapter settings	RHEL 6.3 TCP stack setting	Comments
1	Standard	Forced / on	Default	Default	Default	Default configuration
2	**Jumbo**	Forced / on	Default	Default	Default	Jumbo frames enabled
3	Jumbo	Forced / on	**Maximum**	Default	Default	Baseline configuration to evaluate all subsequent settings

Table 5 Test case sequence for Intel X520 to evaluate other configuration options

Test case	Frame size	Flow Control	Rx / Tx buffers	Other adapter settings	RHEL 6.3 TCP stack setting	Comments
4	Jumbo	Forced / on	Maximum	Default	**Receive Window Scaling disabled**	
5	Jumbo	Forced / on	Maximum	Default	**Congestion control = reno**	
6	Jumbo	Forced / on	Maximum	Default	**TCP low latency enabled**	
7	Jumbo	Forced / on	Maximum	**TCP / Generic Segmentation Offload disabled**	Default	
8	Jumbo	Forced / on	Maximum	**Scatter Gather (DMA) disabled**	Default	Disabling Scatter Gather turns off TSO also
9	Jumbo	Forced / on	Maximum	**Large Receive Offload disabled**	Default	

4 Performance results

All test case performance results for each NIC mode and workload combination are presented in this section. For each NIC mode, the first chart displays the percentage difference in performance between standard frames and jumbo frames for each tested SAN I/O workload. The second chart displays the percentage difference in performance between the baseline configuration and each individual setting change for each tested SAN I/O workload. In the case of the Broadcom iSOE mode, the second chart shows the performance relative to software initiator mode.

Based on the results, recommended configurations for each NIC mode are given in Section 5.

Instructions for making configuration changes can be found in Appendix B.

> The performance results illustrated below may not reflect all EqualLogic PS Series SAN environments. It is recommended that each potential configuration change be evaluated in the environment prior to implementation.

4.1 Broadcom BCM57810 software initiator mode

Figure 3 and Figure 4 show the performance results for the Broadcom BCM57810 in software initiator mode.

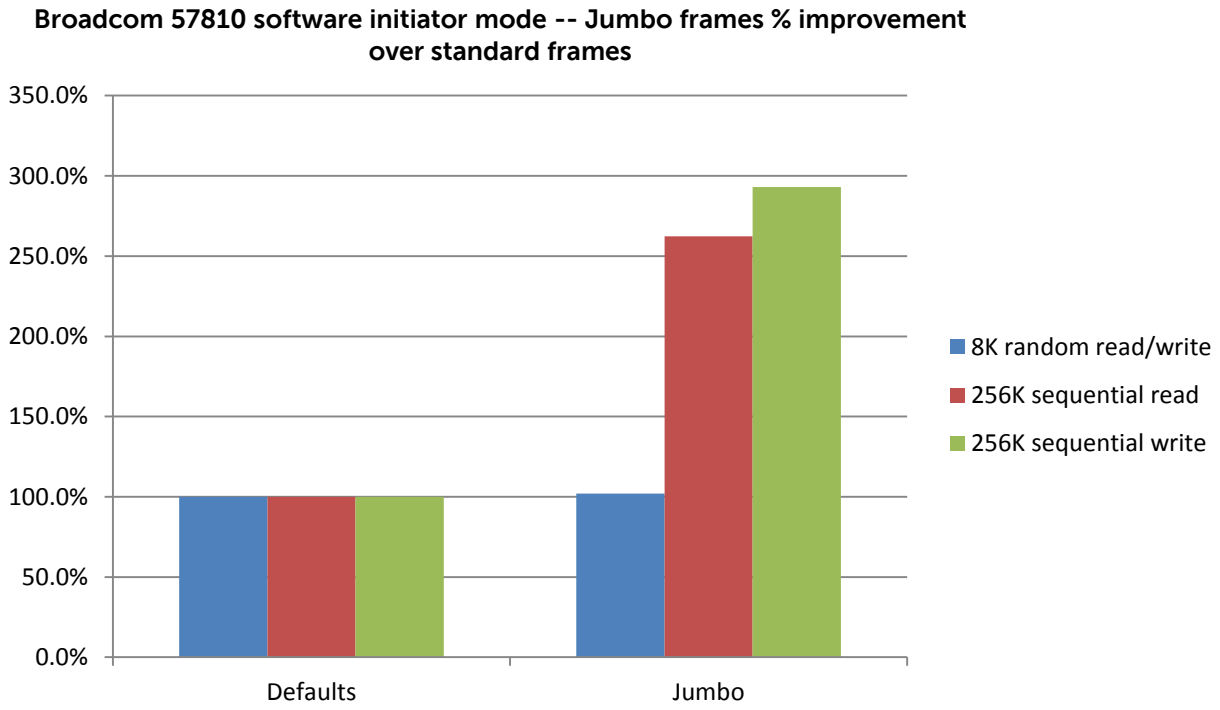

Figure 3 The performance effects of enabling jumbo frames when using Broadcom 57810 in software initiator mode

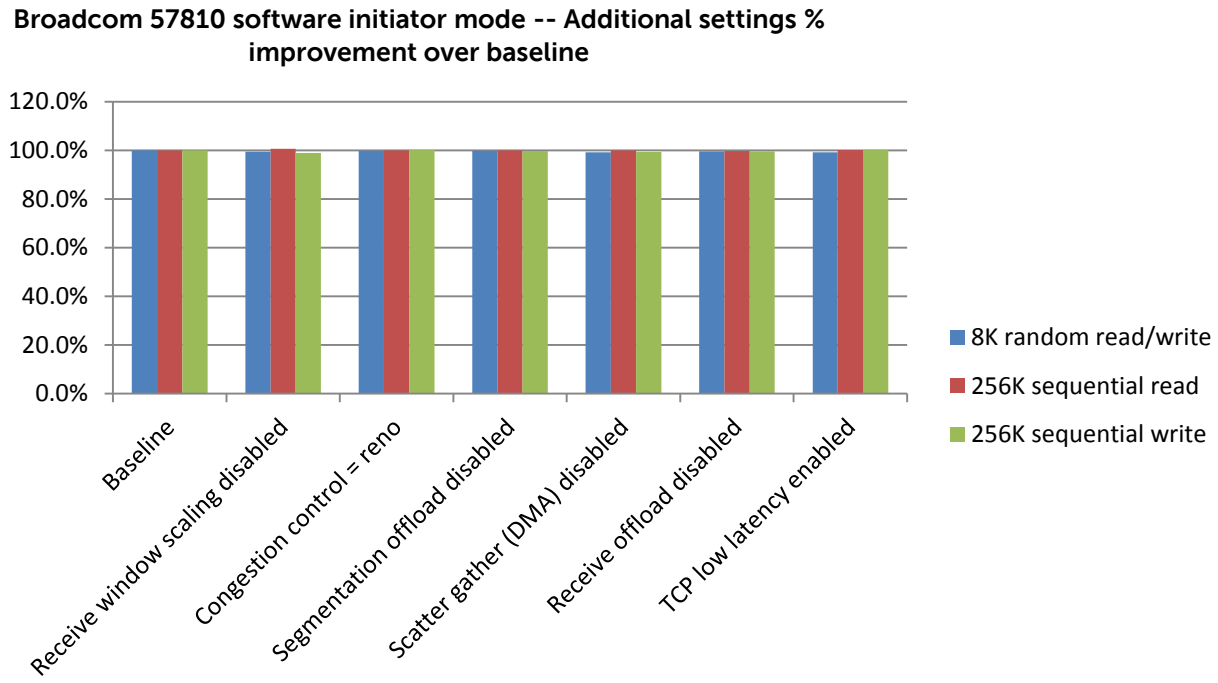

Figure 4 Performance effects of individual configuration changes relative to the baseline configuration on a Broadcom 57810 in software initiator mode

4.2 Broadcom BCM57810 iSOE mode

Figure 5 and Figure 6 show the performance results for the Broadcom BCM57810 in iSOE mode.

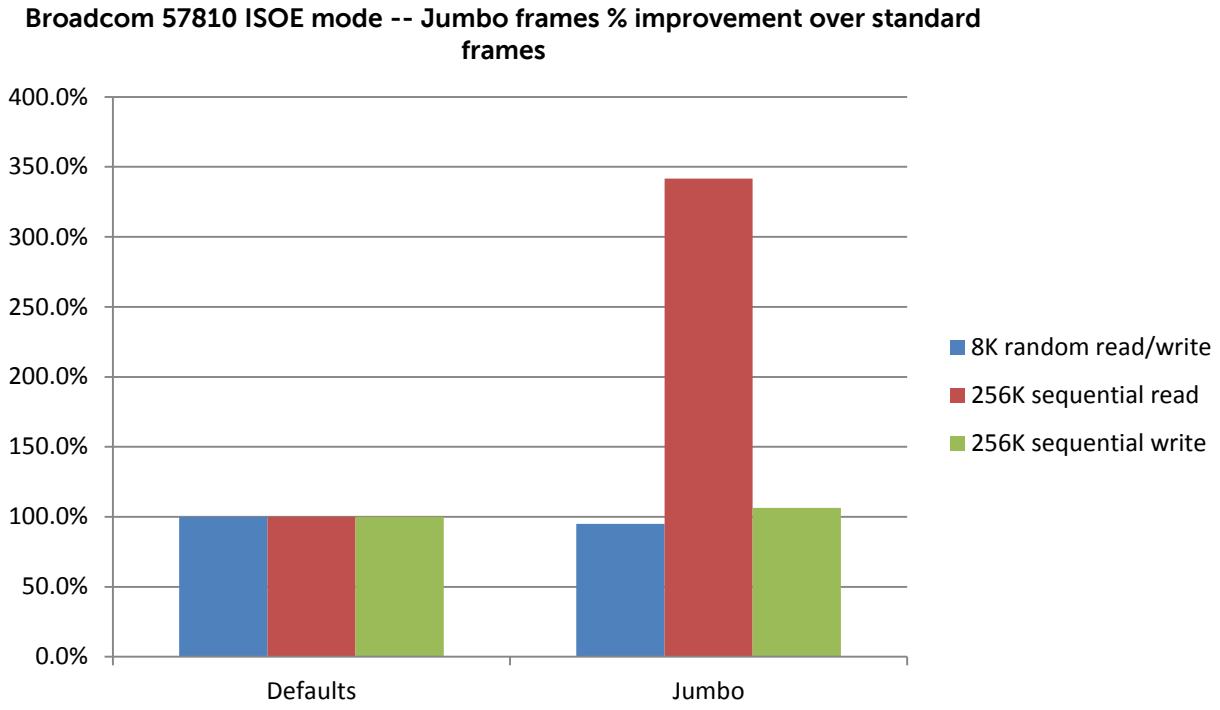

Figure 5 The performance effects of enabling jumbo frames when using Broadcom 57810 in iSOE mode

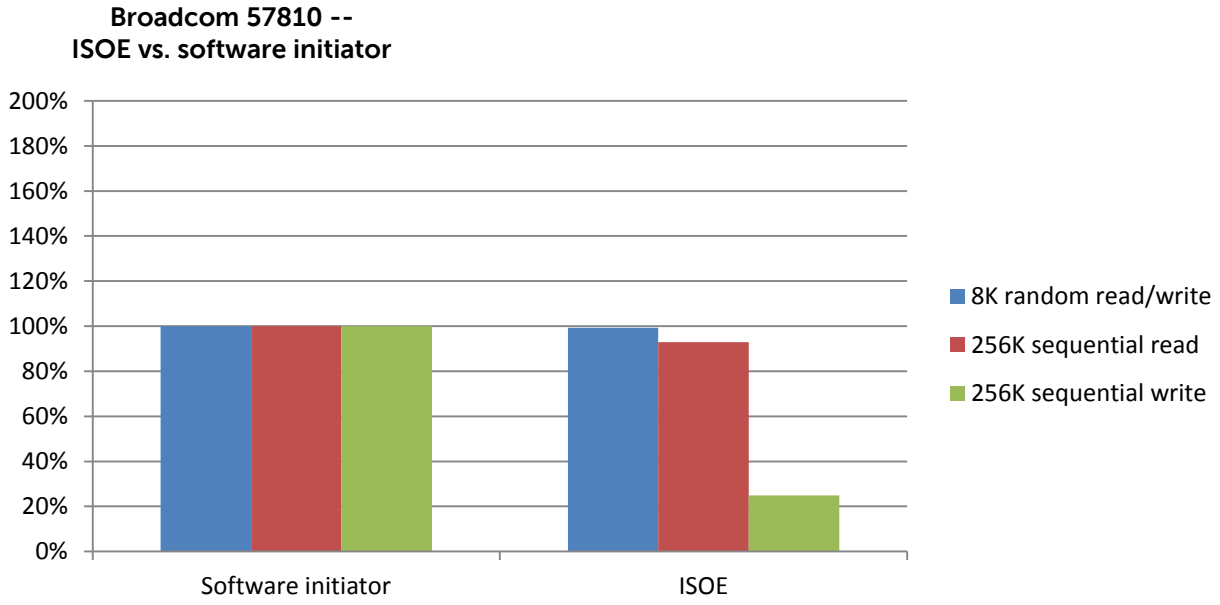

Figure 6 The performance effects of Broadcom iSOE mode relative to software initiator mode

4.3 Intel X520 software initiator mode performance results

Figure 7 and Figure 8 show the performance results for the Intel X520 in software initiator mode.

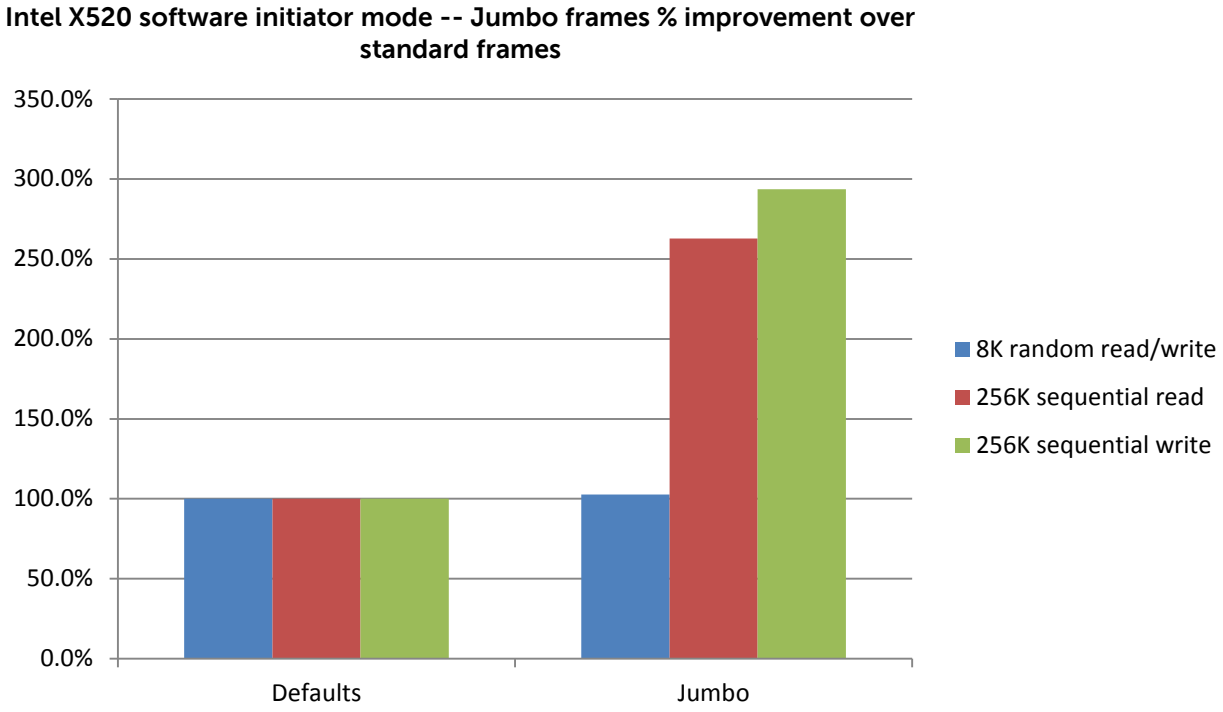

Figure 7 The performance effects of enabling jumbo frames when using Intel X520 in software initiator mode

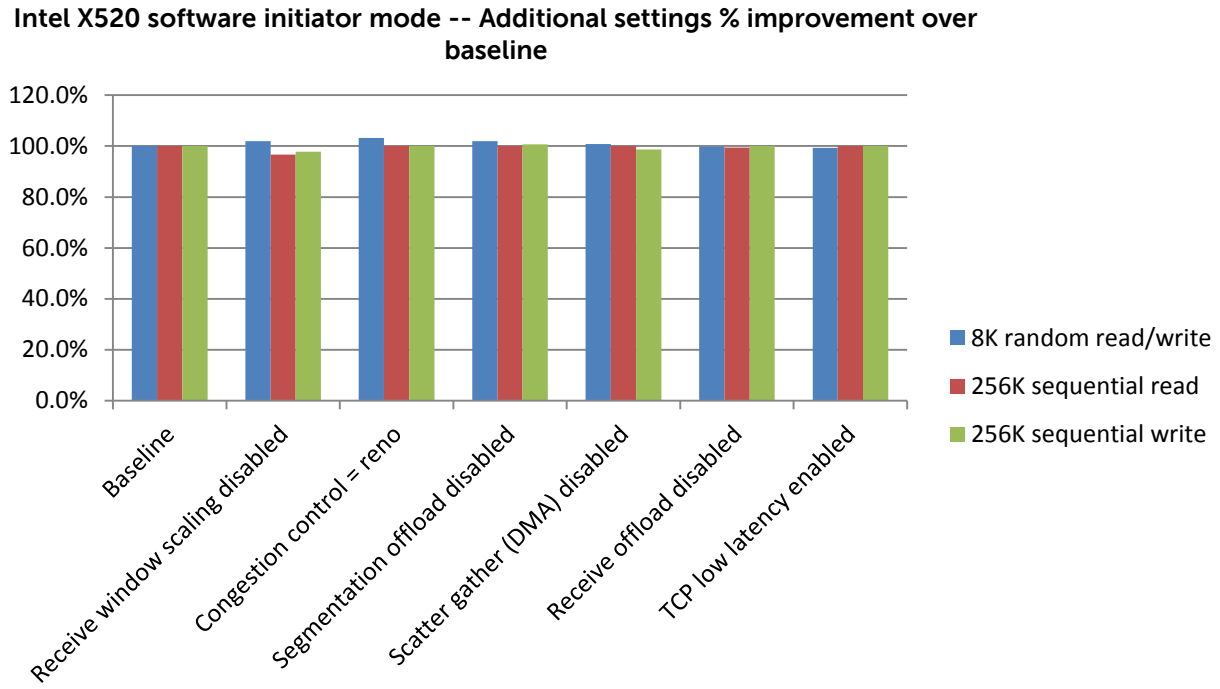

Figure 8 Performance effects of individual configuration changes relative to the baseline configuration on an Intel X520 in software initiator mode.

5 Recommended configurations

In this section, recommended configurations will be given based on the performance results and analysis detailed in Section 4_Results_and_analysis. Non-default settings are recommended only when a compelling difference in performance for one or more workloads was observed, or when a setting is known to provide benefit during network congestion or heavy CPU utilization. *Only the non-default settings are listed.*

For a complete list of tested options and default values, as well as instructions on making configuration changes to the storage host, see Appendix B.

5.1 Broadcom BCM57810 software initiator mode

Based on the performance results and analysis for each workload, the following NIC and OS configuration *changes* are recommended.

Table 6 Broadcom BCM57810 software initiator mode recommended adapter configuration changes

Setting	Default value	Recommended value
Flow Control	Autonegotiate = on	Autonegotiate = off
Jumbo packet	1500	9000
Receive buffers	407	4078

5.2 Broadcom BCM57810 iSOE mode

Based on the performance results and analysis for each workload, the following NIC configuration *changes* are recommended.

Table 7 Broadcom BCM57810 iSOE mode recommended adapter configuration changes

Setting	Default value	Recommended value
MTU	1500	9000

5.3 Intel X520 software initiator mode

Based on the performance results and analysis for each workload, the following NIC configuration *changes* are recommended.

Table 8 Intel X520 software initiator mode recommended adapter configuration changes

Setting	Default value	Recommended value
Jumbo packet	1500	9000
Receive Buffers	512	4096

Transmit Buffers	512	4096

6 Conclusion

Configuration changes beyond the baseline configuration were not justified by an analysis of the performance data from either the base or the congested SAN environments. Therefore, the recommended configuration for network adapters in RHEL 6.3 is the following:

- Jumbo frames enabled
- Flow control on, auto-negotiation off
- Maximum transmit and receive buffers
- All else default

While iSOE performance was comparable to the software initiator during the 8 KB random read/write and the 256 KB sequential read workloads, iSOE throughput during the 256K sequential write workload was 75% less than when using software initiator mode as seen in Figure 6.

Furthermore, during both 256K sequential read and write workloads, CPU I/O wait percentages of over 40% (as reported by the Linux iostat utility) were observed when using iSOE mode, more than double the I/O wait percentage when using software initiator mode. This means that the CPU is waiting for I/O processes to complete and is an indication of an I/O bottleneck.

Further investigation revealed that the Expected Data Transfer Length of the iSCSI writes was 256 KB when using software initiator mode but only 16 KB when using Broadcom iSOE mode. This would explain the performance difference relative to software initiator mode of the larger block size sequential write workload. It is possible that this behavior could cause iSOE to exhibit better relative performance in certain SAN environments.

Broadcom iSOE performance should be evaluated prior to any implementation in Linux.

A Test configuration details

Hardware	Description
Blade enclosure	Dell PowerEdge M1000e chassis: • CMC firmware: 4.45

Blade server	Dell PowerEdge M620 server:
	• Red Hat Enterprise Linux 6.3 x86_64
	• BIOS version: 1.7.6
	• iDRAC firmware: 1.40.40
	• (2) Intel® Xeon® E5-2650
	• 64GB RAM
	• Dual Broadcom 57810S-k 10GbE CNA
	  • Base driver: 1.78.80
	  • ISOE driver: 2.7.8.2b
	  • Firmware: 7.8.53
	• Dual Intel x520-k 10GbE CNA
	  • Driver: 3.6.7-k
	  • Firmware: 14.5.9
	• Dell EqualLogic Host Integration Tools for Linux 1.2.0

Blade I/O modules	(2) Dell 10Gb Ethernet Pass-through module
SAN switches	(2) Dell Force10 s4810 • Firmware: 9.2.0.0
SAN array members	(2) Dell EqualLogic PS6110XV • (2) 10GbE controllers • Firmware: 6.0.6 H2

B Network adapter and TCP stack configuration details

This section provides more detail about the configuration options and default settings of the network adapter properties and the RHEL 6.3 TCP stack.

B.1 Software initiator mode adapter options

The following tables list the tested adapter options for the Broadcom BCM57810 NetXtreme II 10 GigE NIC and the Intel X520 in software initiator mode along with the default value for each.

Table 9 Broadcom BCM57810 software initiator mode adapter options

Setting	Default value
Flow control	Autoneg Tx/Rx Enable
TCP / Generic segmentation offload	On
MTU	1500
Receive buffer descriptors	407
Large receive offload	On
Scatter gather (DMA)	On
Transmit buffers descriptors	4078

Table 10 Intel X520 software initiator mode adapter options

Setting	Default value
Flow control	Forced Tx/Rx Enable
TCP / Generic segmentation offload	On
MTU	1500
Receive buffer descriptors	512
Large receive offload	On
Scatter gather (DMA)	On
Transmit buffers descriptors	512

B.2 Configuring adapter options in software initiator mode

Adapter properties for the Broadcom BCM57810 and Intel X520 can be set and confirmed with native Linux utilities.

To view current settings use the following methods.

- To view the MTU size for each adapter interface.
  - `ifconfig`
- To view flow control, ring buffer size, and offload settings.
  - Flow control
    - `ethtool -a <interface>`
  - Ring buffer size
    - `ethtool -g <interface>`
  - For offload settings
    - `ethtool -k <interface>`

To configure recommended non-default settings persistently across reboots, place the following lines in each adapter's /etc/sysconfig/network-scripts/ifcfg-<interface> configuration file.

- For Broadcom:
  - `MTU=9000`
  - `ETHTOOL_OPTS="-G <interface> rx 4078; -A <interface> autoneg off"`
- For Intel
  - `MTU=9000`
  - `ETHTOOL_OPTS="-G <interface> rx 4096 tx 4096"`

B.3 Broadcom BCM57810 iSOE mode adapter options

The following table lists the tested adapter options for the Broadcom BCM57810 NetXtreme II 10 GigE NIC in iSOE mode along with the default value.

Table 11 Broadcom BCM57810 iSOE mode adapter options

Setting	Default value
MTU	1500

B.4 Configuring Broadcom BCM57810 adapter properties in iSOE mode

The Broadcom iSOE adapter inherits the MTU setting from the corresponding Ethernet adapter. As long as the corresponding Ethernet adapter is administratively active and configured for jumbo MTU the iSOE adapter will initiate iSCSI sessions using jumbo frames.

Note: Every step that is described in this section must be applied to every Broadcom CNA port that will be configured in iSOE mode to be connected to the SAN. For example **p1p1** and **p1p2**

1. Install the `iscsi-initiator-utils` package.

2. Ensure bnx2i kernel module is loaded.
   ```
   lsmod | grep bnx2i
   ```

3. If not loaded, then load the kernel module.
   ```
   modprobe -a bnx2i
   ```

4. Use ifconfig to find the MAC address of the Broadcom adapter ports. The iSOE MAC address should be +1 from the base NIC MAC address. For example:
   a. Base adapter = E0:DB:55:10:46:72
   b. iSOE adapter = E0:DB:55:10:46:73

5. To find the iSOE interface name, run the following iscsiadm command:
   ```
   iscsiadm –m iface
   ```

6. Assign an IP address and subnet to each iSOE interface to be configured.
   ```
   iscsiadm -m iface -I <iSOE-interface> -o update -n iface.ipaddress -v <IP-address>
   iscsiadm -m iface -I <iSOE-interface> -o update -n iface.subnet_mask -v <subnet mask>
   ```

7. If a VLAN other than the default VLAN ID is being used, change the following parameters. If the default VLAN ID is used, skip this step.
```
iscsiadm -m iface -I <iSOE-interface> -o update -n iface.vlan_id -v <vlan id>
iscsiadm -m iface -I <iSOE-interface> -o update -n iface.iface_num -v <vlan id>
```

   The corresponding Ethernet adapter must also be administratively active and configured for the proper VLAN ID using the following instructions.

   a. Configure your physical interface in `/etc/sysconfig/network-scripts/ifcfg-ethX`, where *X* is a unique number corresponding to a specific interface, as follows:
```
DEVICE=ethX
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MTU=9000
```

   b. Configure the VLAN interface in `/etc/sysconfig/network-scripts`. The configuration filename should be the physical interface, a period and the VLAN ID number. For example, if the VLAN ID is 192, and the physical interface is `eth0`, then the configuration filename would be `ifcfg-eth0.192`:
```
DEVICE=ethX.192
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
NETMASK=255.255.255.0
USERCTL=no
NETWORK=192.168.1.0
VLAN=yes
```

Note: If a second VLAN needs to be configured, add a new file for the VLAN configuration details. For example, VLAN ID 193, on the same interface, `eth0` needs the file name ifcfg-`eth0.193`.

   c. Restart the networking service, in order for the changes to take effect, as follows:
```
service network restart
```

8. Set Jumbo frames to each iSOE interface to be configured.
```
iscsiadm -m iface -I <iSOE-interface> -o update -n iface.mtu -v 9000
```

9. Confirm iSOE interface settings.
```
iscsiadm -m iface -I <iSOE-interface>
```

B.5 RHEL 6.3 TCP stack options

The following table lists the tested TCP stack options for RHEL 6.3 along with the default value.

Table 12 RHEL 6.3 TCP stack options

Setting	Default value
TCP low latency	Off
Congestion control	cubic
Receive window scaling	On

B.6 Configuring the RHEL 6.3 TCP stack

To persistently set non-default settings to the RHEL 6.3 TCP stack add the following lines to /etc/sysctl.conf. Note that the following instructions are for reference only and are not recommended based on the performance results illustrated in this paper.

1. To disable TCP receive window scaling:
   ```
   net.ipv4.tcp_window_scaling = 0
   ```
2. To change the TCP congestion control algorithm to Reno:
   ```
   net.ipv4.tcp_congestion_control = reno
   ```
3. To enable TCP low latency:
   ```
   net.ipv4.tcp_low_latency = 1
   ```

B.7 Installing Host Integration Tools for Linux

Host Integration Tools for Linux provides EqualLogic recommended multi-path (MPIO) functionality, command-line tools for discovering and connecting to EqualLogic volumes, and a performance tuning system check. To install Host Integration Tools for Linux follow the instructions below.

1. Download the Host Integration Tools for Linux ISO from the Dell EqualLogic support site (login required).
   a. https://eqlsupport.dell.com/secure/login.aspx

2. Mount the ISO image from within Linux.

3. Change to the directory of the ISO mount point, for example:
   ```
   cd /media/CDROM
   ```

4. Run the HIT for Linux installer script.
   ```
   ./install --nogpgcheck
   ```

5. Follow the instructions, choosing to include only the SAN interface subnets.

6. If using the Broadcom iSOE adapter, choose **bnx2i** as the iSCSI initiator.

7. Note that eqltune, the EqualLogic performance tuning utility, will be run automatically by the HIT for Linux installer. This utility will detect and fix problematic settings for block devices, Ethernet adapters, sysctl tunable options, and more. Eqltune will record and can, if necessary, restore the original configuration. Run equltune from the command line for further information.

8. Once complete, include into the shell the HIT bash configuration file for command line completion of EqualLogic tools. Note the space between the period and the full path.
   ```
   . /etc/bash_completion.d/equallogic
   ```

B.8 Configuring the Host Integration Tools for Linux

1. Depending on whether you configured the SAN interfaces prior to installing the Host Integration Tools for Linux, you may need to adjust the subnets/interfaces included in the MPIO settings. For example:
   ```
   rswcli --mpio-exclude --adapter=<non-SAN-interface>
   rswcli --mpio-include --ip-address=<iSOE IP>
   ```

B.9 Connecting to iSCSI targets

After configuring the SAN interfaces and installing Host integration Tools for Linux, the last step is to connect to the EqualLogic iSCSI volumes.

1. Confirm that Host integration Tools for Linux is using the correct adapters and initiator by checking the adapter list in the status output.
   ```
   ehcmcli status
   ```
2. Discover the iSCSI target volumes.
   ```
   rswcli --add-group-access --group-ip=<group-IP-address> --group-name=<group-name>
   ```
3. Confirm target volume discovery
   ```
   iscsiadm -m node | sort -u
   ```
4. Login to each iSCSI target volume once. After that, HIT MPIO will create additional sessions as necessary. Sessions will be automatically started at every subsequent boot.
   ```
   ehcmcli login --target <volume-name> --portal <Group-IP>
   ```
5. Confirm session count. Piping the output into "wc -l" returns the number of lines/sessions.
   ```
   iscsiadm -m session
   iscsiadm -m session | wc -l
   ```

For more information on Host integration Tools for Linux, download the Dell EqualLogic Host Integration Tools for Linux Installation and User's Guide version 1.2.0 from the Dell EqualLogic support site (login required).
https://eqlsupport.dell.com/secure/login.aspx

C I/O parameters

Vdbench SAN workloads were executed using the following parameters in the parameter file.

Common parameters:

```
hd=default
hd=one,system=localhost
```

iSCSI volumes (random IO):

```
sd=sd1,host=*,lun=/dev/eql/v1,openflags=o_direct,size=102400m,threads=5
sd=sd2,host=*,lun=/dev/eql/v2,openflags=o_direct,size=102400m,threads=5
sd=sd3,host=*,lun=/dev/eql/v3,openflags=o_direct,size=102400m,threads=5
sd=sd4,host=*,lun=/dev/eql/v4,openflags=o_direct,size=102400m,threads=5
sd=sd5,host=*,lun=/dev/eql/v5,openflags=o_direct,size=102400m,threads=5
sd=sd6,host=*,lun=/dev/eql/v6,openflags=o_direct,size=102400m,threads=5
sd=sd7,host=*,lun=/dev/eql/v7,openflags=o_direct,size=102400m,threads=5
sd=sd8,host=*,lun=/dev/eql/v8,openflags=o_direct,size=102400m,threads=5
```

iSCSI volumes (sequential IO on two arrays):

```
sd=sd1,host=*,lun=/dev/eql/v1,openflags=o_direct,size=30m,threads=5
sd=sd2,host=*,lun=/dev/eql/v2,openflags=o_direct,size=30m,threads=5
sd=sd3,host=*,lun=/dev/eql/v3,openflags=o_direct,size=30m,threads=5
sd=sd4,host=*,lun=/dev/eql/v4,openflags=o_direct,size=30m,threads=5
sd=sd5,host=*,lun=/dev/eql/v5,openflags=o_direct,size=30m,threads=5
sd=sd6,host=*,lun=/dev/eql/v6,openflags=o_direct,size=30m,threads=5
sd=sd7,host=*,lun=/dev/eql/v7,openflags=o_direct,size=30m,threads=5
sd=sd8,host=*,lun=/dev/eql/v8,openflags=o_direct,size=30m,threads=5
```

iSCSI volumes (sequential IO on four arrays):

```
sd=sd1,host=*,lun=/dev/eql/v1,openflags=o_direct,size=45m,threads=5
sd=sd2,host=*,lun=/dev/eql/v2,openflags=o_direct,size=45m,threads=5
sd=sd3,host=*,lun=/dev/eql/v3,openflags=o_direct,size=45m,threads=5
sd=sd4,host=*,lun=/dev/eql/v4,openflags=o_direct,size=45m,threads=5
sd=sd5,host=*,lun=/dev/eql/v5,openflags=o_direct,size=45m,threads=5
sd=sd6,host=*,lun=/dev/eql/v6,openflags=o_direct,size=45m,threads=5
sd=sd7,host=*,lun=/dev/eql/v7,openflags=o_direct,size=45m,threads=5
sd=sd8,host=*,lun=/dev/eql/v8,openflags=o_direct,size=45m,threads=5
```

8KB random 67% read workload:

```
wd=wd1,sd=(sd1-sd8),xfersize=8192,rdpct=100,skew=67
wd=wd2,sd=(sd1-sd8),xfersize=8192,rdpct=0,skew=33
```

256KB sequential read workload:

```
wd=wd1,sd=(sd1-sd8),xfersize=262144,rdpct=100,seekpct=sequential
```

256KB sequential write workload:

```
wd=wd1,sd=(sd1-sd8),xfersize=262144,rdpct=0,seekpct=sequential
```

Runtime options:

```
rd=rd1,wd=wd*,iorate=max,elapsed=1200,interval=5
```

Additional resources

Support.dell.com is focused on meeting your needs with proven services and support.

DellTechCenter.com is an IT Community where you can connect with Dell Customers and Dell employees for the purpose of sharing knowledge, best practices, and information about Dell products and your installations.

Referenced or recommended Dell publications:

- EqualLogic Configuration Guide: http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/19852516/download.aspx
- EqualLogic Compatibility Matrix (ECM): http://en.community.dell.com/techcenter/storage/w/wiki/2661.equallogic-compatibility-matrix.aspx
- EqualLogic Switch Configuration Guides: http://en.community.dell.com/techcenter/storage/w/wiki/4250.switch-configuration-guides-by-sis.aspx
- The latest EqualLogic firmware updates and documentation (site requires a login): http://support.equallogic.com

Force10 Switch documentation:

- http://www.force10networks.com/CSPortal20/KnowledgeBase/Documentation.aspx

For EqualLogic best practices white papers, reference architectures, and sizing guidelines for enterprise applications and SANs, refer to Storage Infrastructure and Solutions Team Publications at:

- http://dell.to/sM4hJT

Other recommended publications:

- TCP Implementation in Linux: A Brief Tutorial http://www.ece.virginia.edu/cheetah/documents/papers/TCPlinux.pdf
- Tuning 10Gb network cards on Linux http://landley.net/kdocs/ols/2009/ols2009-pages-169-184.pdf