2013

A Comparative Analysis Of Open Source Storage Area Networks With Esxi 5.1

Robert M. Trinkle
Purdue University

Recommended Citation

Trinkle, Robert M., "A Comparative Analysis Of Open Source Storage Area Networks With Esxi 5.1" (2013). *Open Access Theses.* Paper 97.

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Robert M. Trinkle

Entitled
A Comparative Analysis of Open Source Storage Area Networks With ESXi 5.1

For the degree of Master of Science

Is approved by the final examining committee:

P.T. Rawles
Chair

Raymond Hansen

Thomas Hacker

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): P.T. Rawles

Approved by: Jeffrey L. Whitten 11/25/2013
Head of the Graduate Program Date

A COMPARATIVE ANALYSIS OF OPEN SOURCE STORAGE AREA NETWORKS WITH ESXI 5.1

A Thesis

Submitted to the Faculty

of

Purdue University

by

Robert M. Trinkle

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

December 2013

Purdue University

West Lafayette, Indiana

TABLE OF CONTENTS

Page

APPENDICES

Page

LIST OF TABLES

Table Page

Appendix Table

LIST OF FIGURES

Figure Page

Appendix Figure

LIST OF ABBREVIATIONS

IEEE: Institute of Electrical and Electronics Engineers

IET: iSCSI Enterprise Target

IOPS: Input/ Output Operations Per Second

LUN: Logical Unit Number

MB/s: Mega Byte Per Second

MTU: Maximum Transmission Unit

NIC: Network Interface Card

OSI: Open Systems Interconnect

PERC: PowerEdge Raid Controller

RAID: Redundant Array of Independent Disks

SATA: Serial Advanced Technology Attachment

SCSI: Small Computer System Interface

SCST: Generic SCSI Target Subsystem for Linux

TCP/IP: Transmission Control Protocol/Internet Protocol

VM: Virtual Machine

vSwitch: Virtual Switch

GLOSSARY

Internet Small Computer Systems Interface (iSCSI) – A transport protocol which allows systems to communicate with storage devices over TCP/IP(Satran, Meth, Sapuntzakis, Chadalapaka, & Zeidner, 2004).

Iometer – Open source software which is capable of running multiple tests to benchmark IOPS of storage solutions.

IOPS – The amount of input and output operations on a storage disk per second. Theoretical maximum disk operations can be measured by utilizing a formula based on average latency and seek times (Lowe, 2010).

Maximum Transfer Unit (MTU) – The size of an Ethernet packet in which data can be sent within.

Storage Area Network (SAN) -- A device providing network attached block level data storage which appears as a local resource to an operating system. This storage is presented as a SCSI subsystem encapsulated in a TCP/IP connection(Aiken & Grunwald, 2003).

ABSTRACT

Trinkle, Robert M. M.S., Purdue University, December 2013. A Comparative Analysis of Open Source Storage Area Networks With ESXi 5.1. Major Professor: P.T. Rawles.

Storage Area Networks have increased in popularity with the advancement of virtualization technologies. SANs consist of a central repository of hard disks allowing multiple clients to access and share data over a computer network. In recent years, multiple commercial and open source SAN technologies have been introduced to the market. Unlike commercial products, the open source SAN technologies lack formal documentation and research making implementation best practices scarce. This research analyzed the performance of different SAN architectures and implementation strategies. In addition, this paper expands upon previous research by using current hardware and software technologies. The test results of prominent open source SAN technologies and an analysis of acquired data has added to the body of knowledge regarding SAN best practices.

CHAPTER 1. INTRODUCTION

This chapter is an introduction of the research conducted for this thesis. First, the problem and its significance are addressed. In addition the scope, questions pertaining to the research, assumptions, limitations, and delimitations are examined.

1.1 Background

Storage Area Networks (SAN) and virtualization have become a growing trend in datacenters. The combination of these two technologies can be used consolidate and enhance hardware. The fast adoption of SANs and virtualization has introduced new business practices among individuals and businesses alike.

The term SAN is synonymously referred to as IP SAN. IP SANs are block level storage area networks which communicate over TCP/IP using the iSCSI protocol (Yoder, Carlson, Thiel, Deel, & Hibbard, 2012). From a physical aspect, SANs are normally composed of multiple disks which are redundantly arranged for failover and throughput benefits. The redundant array of independent disks (RAID) is exposed to physical servers through the network fabric and appears as local storage to connected servers. Simplistically, SANs allow multiple servers to connect to a shared storage repository across a computer network.

IBM invented and commercialized mainframe Virtual Machines (VMs) many decades ago. However, VMs didn’t make the leap to commodity hardware until the late 1990s. During this period, VMware pioneered efficient virtualization on x86 platforms (Rosenblum & Waldspurger, 2011). Without much mainstream adoption until the early

2000’s, virtualization technology stayed relatively dormant until becoming popular in datacenters soon after. Virtualization allows for operating systems to be extracted from the hardware layer and run virtually as a VM. One of the largest benefits of virtualization technology is the ability to consolidate multiple energy inefficient physical servers into one powerful server utilizing virtualization. VMware has become one of the top contributors to this market and is referred to as the leading virtualization company by many IT professionals.

Other virtualization products also exist and are prominent in datacenters and workplaces. Microsoft’s virtualization solution, Hyper-V, can be implemented as bare-metal or installed in addition to an operating system. While powerful, Hyper-V lacks some of the robustness other solutions offer such as being limited to fewer resources per virtual machine. Xen, a bare-metal hypervisor, is used to power Amazon’s Elastic Compute Cloud and offers a robust virtualization solution for consumers. According to research, Xen is the leading virtualization platform behind VMware (Csaplar, 2012).

When considering virtualization technologies, Storage Area Networks are complimentary technology. SANs allow virtualization products to take advantage of robust features such migrating virtual machines across hardware without any disruption in service. This process can also be performed automatically in case of hardware failure. In addition, multiple hypervisors can connect to and share data from a single SAN target.

Major information technology manufactures have released commercial SAN solutions certified for use alongside virtualization software (Liu & Ai-shaikh, 2009). Some of the most prominent downsides to these manufactured SAN solutions are the hardware limitations as well as the high price tag. As with many products, commercial SANs are created with generalized hardware suitable for most implementations. This generalized hardware can introduce constraints when additional customizations are needed. Utilizing open source SAN software implemented on project specific hardware, a comparable SAN can be built for a much lower monetary cost (Intel, 2012). The open source SAN technologies, however, often lack instructions and documentation for installation and configuration. The lack of research and documentation regarding best practices and optimization of these open sourced technologies presents many issues when attempting to implement the most effective solution.

1.2 Statement of Problem

Open source Storage Area Networks have become increasingly popular with the advancement and adoption of VMware ESXi. SANs allow centralized storage to appear as local storage to virtual machines. With a wide variety of open source iSCSI target drivers and network protocols available, implementation best practices are scarce due to lacking benchmarks.

1.3 Significance of Problem

Virtualization technologies have been a primary concern in regards to Storage Area Network adoption in datacenters. Customarily, data centers have stored operating systems as well as pertinent data files locally on physical disks connected to a server. It is estimated the cost of powering the U.S.'s datacenters is expected to exceed $15 billion over the next decade (Ren, Wang, Urgaonkar, & Sivasubramaniam, 2012). Due to the increased demand to reduce carbon footprints and energy consumption, resource extensive servers utilizing virtualization have increased in popularity. These powerful servers are used to consolidate older servers which utilize more energy and provide less functionality. Alongside taking advantage of virtualization technologies, physical disks have also been removed from servers and replaced by SANs.

Implementing an open source based SAN introduces multiple configuration options. A multitude of open source iSCSI target engines have been created or altered in the last few years. The primary significance of the increase in iSCSI engines is the lack of published research and throughput specifications among the choices. These configuration options create different scenarios for throughput and data rates. Similar to new technologies, it is important these configuration options and the related effect on network throughput are measured.

Datacenters are constantly implementing IP-based services which can rely heavily on virtualization and storage technologies. Implementing storage and virtualization technologies effectively is necessary when considering services utilized by a vast majority of network users. Due to the variety of options, it is important to have data benchmarks in place to further measure and fine tune network performance.

1.4 Purpose of Research

The purpose of this research is to examine and analyze the effect different technologies such as virtualization and network protocols have on open source Storage Area Network throughput. The research will help datacenters determine the level of overhead virtualization introduces as well as which network metrics hold the most validity when implementing SAN solutions.

1.5 Research Question

This thesis will answer the primary research questions:

- What is the average throughput rates utilizing four different open source Storage Area Network target servers?
    - What effect on throughput rates does ESXi 5.1 introduce compared to a physical initiator?
- What effect does MTU frame size have on four different open source Storage Area Network target servers?
    - What effect does MTU frame size have on physical and virtual iSCSI initiators?

1.6 Assumptions

The assumptions of this project include:

- The SAS hardware and related hard drives are available and work correctly with Fedora.

- Metrics obtained by one initiator accessing a single target is relational to larger implementations.
- The underlying file systems are properly aligned.
- The test methodology is sufficient in representing a general use case

1.7 Limitations

The limitations of this research include:

- The performance testing will be limited to the built in functionality of Iometer.
- The research is limited to the hardware, software, and local area network used.

1.8 Delimitations

The delimitations of this research include:

- This research does not address other available open source iSCSI target servers apart from IET, SCST, LIO, and ISTGT.
- This research only examines the iSCSI protocol.
- This research does not address alternative throughput measuring tools other than Iometer.
- This research does not utilize other virtualization technologies other than ESXi 5.1.
- This research does not examine hardware iSCSI initiators.
- This research does not take into account iSCSI target security.
- This research does not take into account storage redundancy or failover.
- Only one iSCSI initiator will be used during each tests.

1.9 Summary

While the fundamental technologies behind SANs and virtualization have been examined extensively, the recent modernization and adoption in datacenters across the globe has sparked great interest in the technologies. Reducing energy use, becoming more efficient, and reducing total cost of ownership have been some of the primary factors pushing forth the virtualization movement. Due to the increased interest, it is important these technologies be evaluated and compared.

This chapter began by discussing the background of storage area networks and virtualization. The problem statement as well as the significance of the problem was then introduced and defined. In addition, the purpose of the experiment along with the research question was addressed. Finally, limitations, delimitations, assumption were addressed.

CHAPTER 2. LITERATURE REVIEW

During this review of literature, formal benchmarking experiments regarding current open source Storage Area Network (SAN) technologies were outdated. Due to the lack of current experiment information comparing modern open source iSCSI protocols, aspects important to obtaining credible data measurements for this research will be reviewed. Pertinent topics relating to this research are different SAN target servers, network construction, storage optimization, and virtualization, all of which have been reviewed considerably.

Although the focus of this review and research is open source products, previous studies of commercial SAN products were also analyzed when pertinent. The analyzed commercial SAN products studied are simply used for basic comparisons of underlying technologies. Although a clear difference is defined between commercial and open source based SANs, published research that examines commercial SAN products hold merit.

The analysis of literature has been completed using a variety of sources. The primary methods for research utilized Google Scholar and numerous scholarly databases including Compendex and IEEE Xplore. Due to the lack of published research around modern open source iSCSI target engines, various affiliated webpages belonging to these technologies were utilized. Websites of software vendors related to this study are also used.

2.1 iSCSI Target Servers

Internet Small Computer Systems Interface (iSCSI) is a protocol which utilizes TCP/IP. The iSCSI protocol aims to be fully compliant with the standardized SCSI architecture model (Satran et al., 2004). Compliance with the SCSI architecture model is important because iSCSI transports SCSI commands through TCP to the initiator and communicates with the underlying system as a local SCSI disk. Currently, iSCSI is one of the primary protocols used in commercial and open source SAN solutions. While many open source target server protocols are available, linux-iscsi.org (LIO) has emerged as the current Linux kernel standard (Torvalds, n.d.).

The iSCSI protocol works on the basis of an initiator and target. An initiator accesses a target and exchanges SCSI block data over an IP network. The iSCSI target exposes disks to the initiators addressed by logical unit numbers (LUN). An initiator is generally a client computer or server which sends iSCSI requests to the target SAN server. The iSCSI initiator requests are processed by software or hardware components of the system. Software initiators utilize the system kernel and resources to process the iSCSI traffic, while hardware initiators have separate physical offloading capabilities to process iSCSI traffic.

Target engines used in iSCSI have evolved throughout the advancement of Linux and Unix. In previous versions of the Linux and Unix kernel, SCSI Target Framework (STGT/TGT) was the default engine. Recently, major open source software distributions, such as Openfiler, natively utilized the iSCSI Enterprise Target (IET) engine in their open source SAN product. The pseudo kernel successor after IET was the SCSI Target Subsystem (SCST). SCST was a strong contender as the next standard Linux kernel target engine, but has ultimately been replaced by the current standard, LIO. LIO has been included in the Linux kernel beginning with version 2.6 (Linus Torvalds, 2011). FreeBSD, a Unix based operating system, has adopted a developing target named ISTGT.

2.1.1 iSCSI Enterprise Target

The iSCSI Enterprise Target (IET) was a result of splitting away from a previous target implementation, Ardis, because of certain shortcomings. Ardis lacked several functionalities which were solved by IET. The most notable issues corrected with the creation of IET was support for 64-bit architecture and Linux 2.6 kernel support, among others (“The iSCSI Enterprise Target Project,” n.d.). During the creation of IET, advanced Linux kernel support was important because IET was designed to run in user space as opposed to kernel space. Targets implemented in the kernel space allows for direct communication with the physical hardware and resources. Among other features, IET also supports multiple targets and initiators simultaneously. IET is able to provide regular files, block devices, and virtual block devices to initiators (“The iSCSI Enterprise Target Project,” n.d.).

2.1.2 SCST and LIO

The Generic SCSI Target Subsystem (SCST) is currently found in many Linux based storage solutions. SCST was a split of the previously discussed IET framework. SCST came to fruition because IET was thought to have violated many iSCSI standards. These violations presented critical issues such as possible data corruption, rendering the target engine unfit for production environments (“Generic SCSI Target Subsystem for Linux,” n.d.). During the course of SCST maturity, another target engine named linux-iscsi.org (LIO) was presented as an additional alternative to IET. Supporting most modern network fabrics, LIO and SCST share some similarities.

SCST and LIO both reside in the Linux kernel and support iSCSI (Rodrigues, 2011). The most significant similarity between the two targets is the fact both support persistent reservations. Persistent reservations in relationship to iSCSI encompass the ability for clustered storage solutions to participate in the takeover of failed network connections. Persistent reservations can maintain consistent throughput speeds during network failures and prevent issues with multiple hosts accessing the same Logical Unit Number (LUN).

The first difference between SCST and LIO is the way which the protocols handle communication between the initiator and target. According to the SCST home webpage, SCST allows for automatic session reassignment once communicated to the initiator (“Generic SCSI Target Subsystem for Linux,” n.d.). In addition, SCST has listed specifications stating the ability to dynamically adjust or protect against iSCSI commands with incorrect transfer sizes or directions (“Generic SCSI Target Subsystem for Linux,” n.d.). While the published characteristics listed for SCST are scarce, the protocol is described to handle varying transmission sizes better than other target engines which could lead to an increase in IOPS measurements using different network designs.

Second, LIO has the ability to have multiple connections per session (MC/S). Multiple connections, or multiple paths to an initiator, can be used in a variety of ways. MC/S can establish an additional connection through another network path in case of primary path failure. This additional connection is seamless to the prior connection which does not terminate the initial session. Also, MC/S is able to load balance traffic across multiple links that in turn can increase throughput with compatible hardware.

Finally, LIO and SCST handle errors within an iSCSI connection at different levels. According to RFC 3720, iSCSI connection issues can occur at the session, digest, or connection level (Satran et al., 2004). Different connection issues within iSCSI categorize the way target drivers handle errors sent from the initiator due to a broken connection or other error. Depending on the process handling, the error may be carried forth to the SCSI driver. LIO supports a maximum error recovery level of 2, which means it can recover from errors at all three previously mentioned areas (“The Linux SCSI target wiki,” 2013). In contrast, SCST only specifies a maximum error recovery of 0, which entails all connection errors are passed along to the SCSI driver (“Generic SCSI Target Subsystem for Linux,” n.d.).

2.1.3 ISTGT

ISTGT is a kernel level driver which shares many similarities with LIO and SCST. ISTGT supports persistent reservations and also has the ability to utilize MC/s and MPIO. Multiple iSCSI initiators have been tested with ISTGT including Windows Server 2008 R2 and ESXi 5.1, among many others (“An iSCSI target implementation for multipath failover cluster nodes,” n.d.). ISTGT is included by default in version 8.1 newer of the FreeNAS open source storage utility, however, documentation for this driver is lacking.

The main development site for this driver in in Japanese and the documentation is limited.

2.2 Storage Alignment

Storage alignment of file system volumes to underlying storage architecture can increase data transfer rates for certain types of applications. Unaligned formatted storage volumes can cause multiple storage data chunks to be access for a single disk read operation from the operating system. Storage alignment in regards to SANs and virtual machines can be aligned in three areas, SAN LUNs, VMFS volumes, and VMDK files.

Figure 2.1 below (adopted from VMware Figure 1) depicts an incorrectly aligned virtual file system in relationship to its underlying SAN architecture (VMware, 2009). The SAN LUN can be created using RAID or single disk.

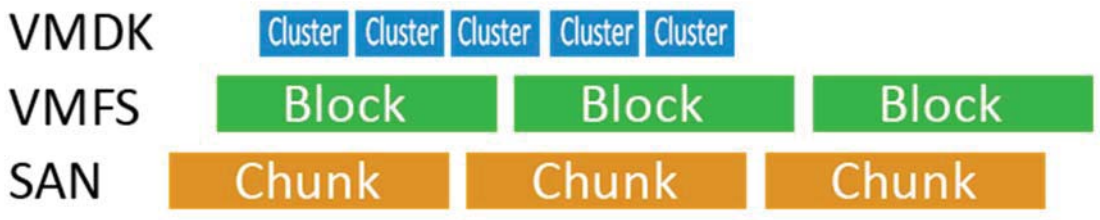


Figure 2.1 Unaligned Virtual File System

The SAN LUN is divided into chunks which each contain multiple sectors. I/O requests from file systems which only request a sector actually read an entire chunk in which the sector belongs. Figure 2.1 depicts an unaligned storage architecture where a read of the third cluster actually spans two VMFS blocks which request three SAN chunks. This request of multiple chunks for a single read can introduce measureable overhead depending on the application. Figure 2.2 below (adopted from VMware Figure 2) depicts a correctly aligned virtual file system in relation to its underlying SAN architecture (VMware, 2009).

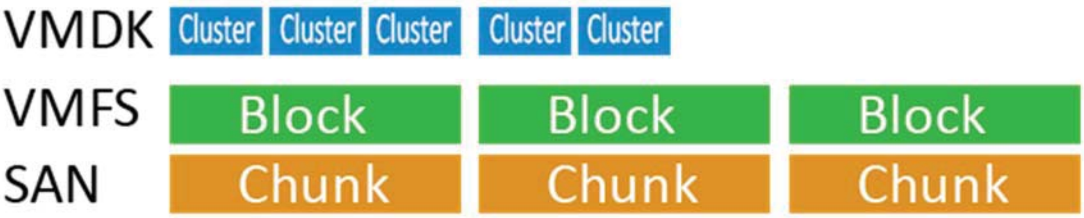


Figure 2.2 Aligned Virtual File System

This depiction shows properly aligned clusters, blocks, and chunks. Proper file system alignment ensures the request of a single cluster does not span multiple chunks of the underlying storage. In this case, accessing the third cluster only requests a single block and chunk optimizing efficiency..

In a storage alignment test from VMware, the results of sequential and random reads using an aligned and unaligned architecture were calculated using Iometer with varying I/O sizes. Correctly aligning the VMFS3 file system yielded an increase of roughly 20 MB/s during sequential reads for larger I/O sizes (VMware, 2009). Correctly aligning the VMFS3 file system yielded an increase of roughly 15-20 MB/s during sequential writes for larger I/O sizes.

The current version of ESXi 5.1 properly aligns the VMFS volume blocks to the SAN chunks upon file system creation. ESXi 5.1 automatically aligns VMFS3 of VMFS5 partitions along the 1 MB boundary alleviating the previous need for manual user alignment (VMware, 2012). While previous versions of operating systems introduced cluster alignment issues, newer versions of Windows and Linux distributions automatically align the boot and data partitions to the underlying file system.

To summarize, modern operating systems along with current versions of VMware ESXi align file systems accordingly based on a default 1 MB boundary. Storage alignment fine tuning can be completed by manually adjusting VMFS block sizes and VMDK cluster sizes if needed, but is not recommended. Using the guided installation methods provided with most applications, the need for manual storage alignment is no longer required. In this research, ESXi 5.1, Windows Server 2008 R2, and the VMware

I/O Analyzer automatically align their respective partitions along a 1 MB boundary. The RAID 0 configuration on the SAN used a chunk size of 1024KB. Because the chunk size is divisible by the 1 MB VMFS and VMDK sizes, the storage is correctly aligned for maximum throughput values.

2.3 Network Construction

The standard method to transport iSCSI data is by utilizing Transmission Control Protocol / Internet Protocol (TCP/IP). Likewise, data communication between computers or devices on a Local Area Network (LAN) is normally handled utilizing TCP/IP. While TCP focuses on communication between applications, IP focuses on passing information between computers or devices.

Commonly discussed at the transport layer of the Open Systems Interconnect Model (OSI), TCP separates data in chunks depending on the Maximum Transmission Unit Size (MTU) of the TCP packet. After the data is separated into packets, it is passed along to the networking layer to be transported to its destination. An acknowledgement aspect is built into TCP which ensures reliable transmission of information sent over the network. Due to the reliability and wide adoption, iSCSI primarily utilizes TCP/IP to pass traffic.

Although not standardized by IEEE, jumbo frames have consistently been referred to as any Ethernet frame with a payload than the standard 1500 bytes. IEEE has determined to not support or define Jumbo frames due to concerns around vendor and equipment interoperability (Faustini et al., 2009). Because there is no standardization, frame sizes above 1500 bytes are only limited by the capabilities of all available hardware. Even though Jumbo Frame sizes are not standardized, common sizes referenced beyond 1500 bytes are 5000, 7200, and 9000 bytes.

In prior studies utilizing 500 MHz processors, MTU sizes have increased bandwidth rates in iSCSI throughput tests over Gigabit Ethernet (GbE) implementations as much as 60% These tests were performed using a prototype iSCSI target engine residing in the Linux kernel similar to LIO and SCST. While fibre channel was also used during some experiments, GbE was primarily the focused fabric.

2.4 Iometer

Developed in 2001 by Intel, Iometer is an open source application that can measure performance metrics of hard drives. These measurements from Iometer are normally presented in the form of IOPS and overall bandwidth. Iometer runs locally on a machine and utilizes a client and server model to measure storage devices. The server, or graphical user interface of Iometer, controls the threaded applications which perform the read and write operations on disks. These threaded applications are run by a service named Dynamo which carries out the simulated read and write operations. According to the Iometer user guide, the software application specializes in measuring system level hard drive and network performance as well as throughput of attached drives (“Iometer User’s Guide,” 2003).

Iometer, among other benchmarking tools, has been used multiple times by previous benchmarking studies. Iometer has software ports for most major operating systems including Linux, Windows, and OSX. According to the download statistics on SourceForge.net, Iometer was downloaded over 290,000 times in the year 2012.

Performance analyses of prior SAN test environments have primarily used Iometer. An experiment with commodity SAN systems utilized Iometer with varying seek and write schedules to measure CPU utilization and storage throughput (Aiken & Grunwald, 2003). This experiment determined the throughput difference between local SCSI operations and network iSCSI operations was visibly apparent. A similar experiment conducted using Iometer and a test environment also found iSCSI introduced significant overhead compared to local tests (Zhang, Yang, Guo, & Jia, 2005). In addition, another experiment also utilized Iometer and ESX 2.0 to depict the negligible difference between virtual machines and native machines using a variety of storage mediums including SANs (Ahmad, Anderson, Holler, Kambo, & Makhija, 2003).

Scott Drummonds (2008), a performance analyst at VMware, stated Iometer is the standard software utility and is recommended for IOPS measurement and analysis in a virtual environment. Measuring disk performance from a virtual machine on a hypervisor can introduce issues if guests generate high CPU utilization. Guests utilizing over 30% of the available CPU resources on the hypervisor can introduce time-based measurement inaccuracies (Drummonds, 2008). While the virtualization software is becoming more mature eliminating possible timing issues, it is important to keep virtual machine CPU utilization at a minimum when performing benchmarking tests in order to obtain the most accurate results. VMware has created a testing appliance built with Iometer called I/O Analyzer which addresses some of the potential timing issues and shortcoming when using Iometer from virtual machines. I/O Analyzer can efficiently generate I/O loads utilizing the VMware VI SDK to remotely collect storage performance statistics (VMware, 2013).

2.5 Iperf

Iperf is an open source network measuring tool to analyze the bandwidth between two endpoints. Iperf is a command line tool with software packages for Linux, Unix, and Windows, among others. This bandwidth monitoring tool has been analyzed and compared to other similar tools and used in many other throughput research studies (Kolahi, Narayan, Nguyen, & Sunarto, 2011). Because Iperf has open source packages for all operating systems in this research, it has been selected as the primary bandwidth monitoring tool for these experiments.

2.6 Summary

Multiple experiments have been conducted measuring performance analysis of virtualization and networking metrics. Although valid, some results from published research offer varying results. Iometer has been the pseudo standard when measuring disk utilization among many different storage areas. Although many open source iSCSI target engines are available, the amount of comparative measurements among the different iSCSI targets is lacking.

In summary this chapter provided a review of four iSCSI target engines and the similarities and differences each share. Also, jumbo frames were examined in addition to their effect on network traffic. Finally, Iometer and associated best practices were summarized. In general, this review of literature explains key technologies which will be used throughout this research.

CHAPTER 3. METHODOLOGY

This chapter discusses the research methodology which was used to address the primary research questions:

- What is the average throughput rates utilizing four different open source Storage Area Network target servers?
    - What effect on throughput rates does ESXi 5.1 introduce compared to a physical initiator?
- What effect does MTU frame size have on four different open source Storage Area Network target servers?
    - What effect does MTU frame size have on physical and virtual iSCSI initiators?

As an overview, the technologies used in this experiment, the framework of the methodology, and the intended acquired data will be discussed.

The research conducted and the metrics obtained from this methodology are quantitative in nature. The primary metric obtained and analyzed throughout the experiment is recorded in IOPS and throughput (MB/s) values. Each independent storage disk and RAID array has a theoretical maximum IOPS value. The purpose is to create an environment and examine different iSCSI protocols while measuring the effects networking protocols and virtualization have on a set of baseline storage values.

3.1 Framework

In order to accurately obtain throughput metrics for this experiment, it was necessary to create a practical test environment representative of what is found in datacenters. The standard testing environments have been created and duplicated for each iSCSI target server. The test environments consisted of an iSCSI target server, a physical and virtual client machine, and a network switch. It was necessary to create multiple environments to separate iSCSI target server implementations. It was also determined only a single iSCSI initiator would be connected to an iSCSI target at a time. The networking equipment utilized was 1 GbE interfaces and Cat5e rated Ethernet cables. In Figure 3.1 below, the test environment consisting of a physical and virtual environment are shown. Environment 1 shown in Figure 3.1 depicts the topology created and used to examine a physical iSCSI initiator implementation. Environment 2 shown in Figure 3.1 was used to examine a virtual machine alongside an iSCSI initiator.

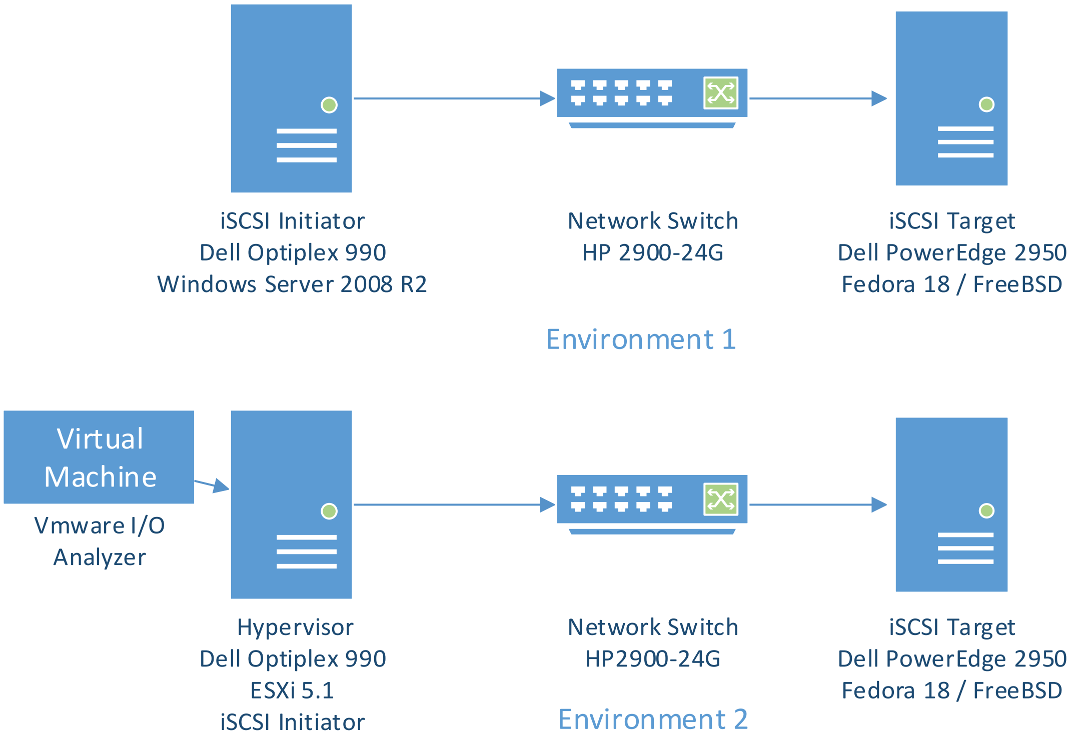


Figure 3.1 Logical Test Enviornment

The complete network topology and architecture can be found in Appendix A.

For IET, SCST, and LIO iSCSI Target servers, the Linux distribution Fedora 18 was installed and configured with default settings on a Dell PowerEdge 2950. The PowerEdge 2950 was configured with a single 500 GB SATA Seagate Barracuda hard drive used for the operating system installation. In addition, four additional Barracuda SAS 500 GB were implemented in a RAID 0 configuration for use as the iSCSI backing store. The hardware RAID configuration was configured using the PERC 6/I controller in the PowerEdge 2950. Additionally, an Intel Gigabit Ethernet NIC was installed in the PowerEdge 2950. It was necessary to use separate NICs for management traffic and iSCSI traffic on separate VLANS. Table 3.1 below shows detailed specifications for the Dell PowerEdge 2950 used.

Table 3.1 Dell Optiplex 2950 Specifications

Processor	Intel R Xeon L5335 @ 2.00 GHz
Networking	Dual embedded Broadcom NetXtreme II 5708 Gigabit Ethernet NIC Additional PCI-Express Intel 893647 Ethernet 10/100/1000
Memory	16 GB 533MHz
Hard Drive(s)	System: 500 GB SATA Seagate Barracuda ST500DM002 RAID: 4 X Barracuda ES.2 SAS 500 GB ST3500620SS
RAID Controller	PERC 6/i

Fedora 18 was chosen as the Linux distribution because natively it supports two of the three Linux based iSCSI target servers which were tested. On a single installation of Fedora 18, the latest kernel available at the time, linux-3.9.4, was implemented and modified to support SCST. Modifying the kernel was performed due to recommendations from the SCST documentation (“Generic SCSI Target Subsystem for Linux,” n.d.) The last tested target server, ISTGT, was implemented on a separate SATA hard drive. ISTGT was configured using FreeNAS 9, an open source storage utility built on FreeBSD. The pertinent configuration files modified for each iSCSI target server can be referenced in Appendix B through Appendix E.

The iSCSI initiators used in the architecture were built using Dell OptiPlex 990's. The first iSCSI initiator configured utilized the software iSCSI initiator built into Windows Server 2008 R2. The Windows Server operating system was installed directly on the local hard drive of the Optiplex 990. The Optiplex 990 was modified with an additional Intel Gigabit Ethernet network interface card to separate management traffic from iSCSI traffic. In order to measure throughput metrics on Windows Server 2008 R2, it was necessary to install the latest version of Iometer 2006.07.27. This environment is referred to in later portions of this thesis as the physical initiator environment. Additional specifications of the Optiplex 990s are shown in Table 3.2 below.

Table 3.2 Dell Optiplex 990 Specifications

Processor	Intel 2nd Generation Core i7 2600 with Intel vPro Technology
Networking	Integrated Intel 82579LM Ethernet 10/100/1000 Additional PCI-Express Intel 893647 Ethernet 10/100/1000
Memory	4 X 4GB Non-ECC dual-channel 1333MHz DDR3 SDRAM
Hard Drive(s)	3.5" 250GB 7200 RPM SATA 3.0 Gb/s

A second Optiplex 990 with the same resources shown in Table 3.2 was utilized for the second iSCSI initiator, referred to in later portions of this thesis as the virtual initiator environment. VMware ESXi 5.1u1 was installed to the hard drive of the Optiplex 990 with default settings. It should be noted that a custom version of ESXi was utilized to support the onboard Intel82579LM NIC chipset on the Optiplex 990. After installation, ESXi was configured to utilize two separate vSwitches. One vSwitch was used primarily for machine management while the second vSwitch was used solely for iSCSI traffic. Because the native installation of ESXi 5.1u1 cannot accurately generate iSCSI traffic to measure throughput, it was necessary to configure a virtual machine to generate workload. In order to create workload, VMware's I/O analyzer 1.5.1 was deployed on the hypervisor. This virtual machine was deployed with the default settings initially and then modified to match the resources of the physical iSCSI initiator. The actual amount of RAM, number of CPU cores, and virtual hard drive space for the VM was slightly less than its physical counterpart due to resource allocations needed to run ESXi 5.1u1. The VMware I/O analyzer was configured with one virtual CPU with eight cores and 14 GB of RAM. Additionally, a second thick provisioned 60 GB hard drive was added to the virtual machine. This second hard drive was necessary to ensure the metric tests were not cached while running on the SAN (VMware, 2013). Both virtual drives were created on the iSCSI target.

The network switch connecting the initiator and targets remained constant throughout the architecture as a HP ProCurve 2900-24G. The ProCurve switch was chosen because of hardware availability which supported jumbo frames. The ProCurve 2900 was divided into two separate VLANS. Ports 1-12 were configured on VLAN 304 used for management traffic during the experiments. Ports 13-24 were configured on VLAN 900 which was used solely as iSCSI traffic. Each iSCSI port was configured to accept a maximum frame size of 9014 bytes. The complete configuration for the HP Procurve 2900-G switch is shown in Appendix F.

3.2 Testing Methodology

It was first necessary to determine the maximum throughput values on the iSCSI RAID array. To measure the maximum disk throughput, hdparm was used locally on the SAN servers and ran against the iSCSI RAID array. In addition, it was important to verify network connectivity from initiator to target was running as expected. To obtain maximum network throughput values, Iperf was ran from the client initiators to the target servers.

To compare initiators and different targets, it was necessary to create a standard set of tests. The VMware I/O Analyzer has preset tests to simulate different applications. Each test has a predetermined block size, read/write percentage, and random/sequential percentage to reflect characteristics of different scenarios or applications. A variety of tests were selected from the available presets as the testing methodology. These tests were also replicated on the physical Windows Server 2008 R2 machine running Iometer and saved as an .icf file for continued use. The series of tests performed during each experiment are shown below in Table 3.3. The methodology remained consistent throughout the course of each experiment.

Table 3.3 Iometer Test Methodology

Test Description	Block Size	Read Percentage	Write Percentage	Random Percentage	Outstanding I/O
Maximum Throughput	512k	100	0	0	32
Maximum IOPS	512b	100	0	0	32
Maximum Write Throughput	512k	0	100	100	32
Maximum Write IOPS	512b	0	100	100	32
Exchange 2003	4k	60	40	80	12
Exchange 2007	8k	55	45	80	12
SQL 16K	16k	66	34	100	16
SQL 64K	64k	66	34	100	16
Web Server	8k	95	5	75	4
Workstation	8k	80	20	80	4

3.2.1 Experiments

Each iSCSI SAN target server was evaluated using a standard testing procedure. To begin, Iometer running on the physical iSCSI initiator was tested while connected to the IET iSCSI target server. Each test from the methodology outlined previously in Table 3.3 was performed for five minutes. Each test in the methodology was performed three times and the results were averaged. After the physical iSCSI initiator tests were completed, the steps were reproduced using the VMware I/O analyzer on the initiator running ESXi 5.1u1. After both initiators were tested the MTU size was altered on the ESXi iSCSI vSwitch, the iSCSI network interface on Windows Server 2008 R2, and on the iSCSI network interface of the IET iSCSI target server. The MTU size on the vSwitch and IET iSCSI interface was set to 9000 bytes while the Windows Server 2008 R2 iSCSI NIC was set to a value of 9014 bytes. The same methodology was performed again on both iSCSI initiators.

Once all tests utilizing standard and jumbo frames were completed, the series of tests were replicated using the SCST, LIO, and ISTGT iSCSI target servers for a total of four different experiments. The primary metrics obtained from each set of tests were recorded in IOPS and throughput measured in MBps.

3.3 Analyzing Data

After the series of experiments were completed on each iSCSI target server, the data collected was analyzed to answer the research questions. First, the data collected for each test was examined and compared to the theoretical maximum network values and disk values obtained from the iPerf and hdparm tests previously conducted. Next, the values obtained from the physical Windows server and the VMware virtual machines were compared to these values. Finally, the physical and virtual metrics obtained using standard frames were compared to results from jumbo frames. This series of comparative analysis was repeated for each experiment conducted. The successfulness of each SAN was determined from the test sets. A successful SAN or test set is defined as having a majority of higher throughput or IOPS values compared to other tests.

3.4 Summary

This chapter explained the importance of creating an accurate framework architecture to test four different iSCSI target engines. In addition, the testing methodology was discussed which explained the test set used. Finally, a description of the experiments conducted was explained and detailed.

To summarize, the methodology used to test different architectures was explained. This methodology consists of a series of 10 tests. The test set was performed from two different iSCSI initiators using the network throughput measuring tool, Iometer. Each initiator test set was performed three times. The values of each group of test sets were averaged to obtain the final values.

CHAPTER 4. RESULTS AND DISCUSSIONS

During the course of the experiments, a large amount of data was collected. Data sets include three trials of each testing methodology performed four times per experiment. The multiple sets of data were averaged together to provide finalized values which were used to compare scenarios proposed during the research questions. This chapter analyzes the collected data and infers conclusions from the data trends.

While collecting data, it was observed certain tests from the methodology are better compared when grouped together. To measure throughput (MBps), the subset of tests which do not reflect maximum IOPS measurements are grouped and compared. The throughput measuring tests have also been logically subdivided into two different categories. The categories include maximum throughput tests and application specific throughput tests.

The maximum throughput tests simply measure the highest iSCSI initiator obtainable value. The application specific test set provides an overview of how the initiators handle application specific traffic. When observing obtainable IOPS, the two tests which discretely measure maximum IOPS values are grouped and compared. The tests which were used to compare throughput are shown below in Table 4.1. Additionally, the tests which were used to compare IOPS are shown below in Table 4.2.

Table 4.1 Throughput Tests

Test Description	Block Size	Read Percentage	Write Percentage	Random Percentage	Outstanding Workers
Maximum Throughput	512k	100	0	0	32
Maximum Write Throughput	512k	0	100	100	32
Exchange 2003	4k	60	40	80	12
Exchange 2007	8k	55	45	80	12
SQL 16K	16k	66	34	100	16
SQL 64K	64k	66	34	100	16
Web Server	8k	95	5	75	4
Workstation	8k	80	20	80	4

Table 4.2 IOPS Tests

Test Description	Block Size	Read Percentage	Write Percentage	Random Percentage	Outstanding Workers
Maximum IOPS	512b	100	0	0	32
Maximum Write IOPS	512b	0	100	100	32

Each series of experiments and test sets have been used to analyze and compare the iSCSI target servers. In addition, the data has been used to compare the effect virtualization has on iSCSI traffic in comparison to a physical machine utilizing iSCSI. Finally the effect of different MTU sizes and the benefit or degradation of throughput values have been examined.

Each comparative analyses performed was based on tests with the highest values. After a comparative analysis of each iSCSI target was completed, all four experiments were compared and analyzed in an attempt to determine the fastest server in regards to the testing environment and methodology used.

4.1 Local RAID and Network Results

The iPerf utility was used to determine the network speed between initiator and target. It was discovered the average network throughput speed between initiator and target was 112 MB/s. This value is near the maximum theoretical bandwidth of Gigabit Base Ethernet at 125 MB/s. It was determined 112 MB/s would stand as the maximum theoretical throughput for each experiment.

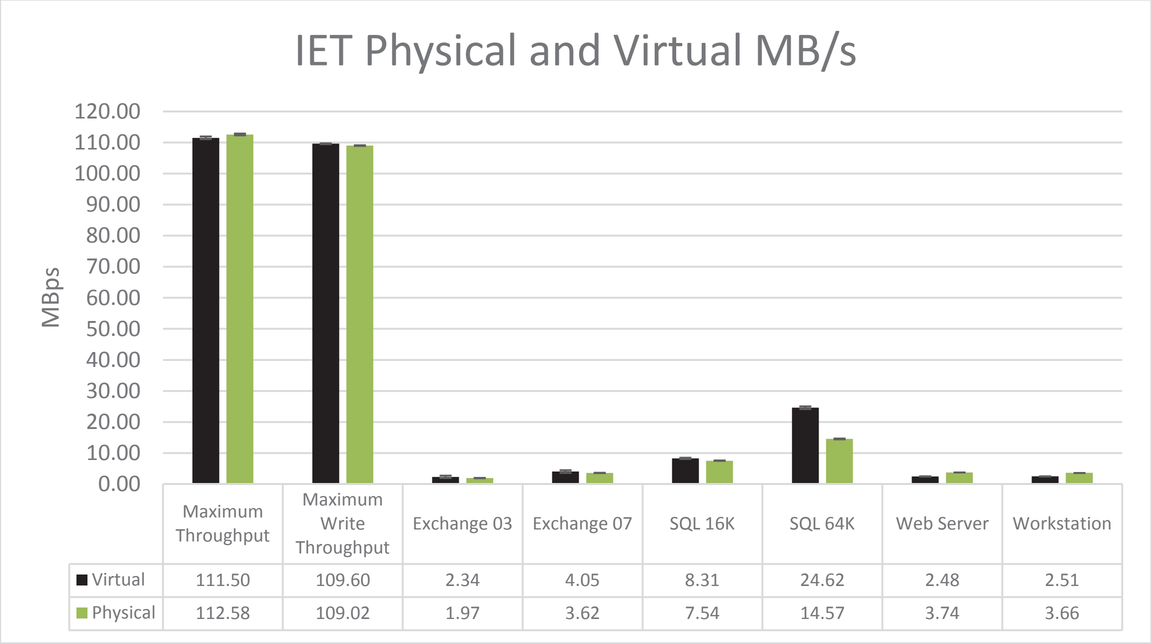
To ensure the Dell PowerEdge 2950 which was the underlying hardware for the iSCSI target servers was not a bottleneck, hdparm was used to measure the read and write speeds of the local RAID array. Using hdparm, it was discovered the maximum total MB/s speed of the RAID 0 iSCSI array was 430 MB/s. This data confirms the SAN server hardware was performing as expected and throughput speeds are limited by the network fabric.

4.2 IET iSCSI Target Results

To compare physical and virtual architectures against the IET iSCSI target server, a series of tests were performed. First, the throughput and IOPS from both initiator architectures were gathered and compared. Then, the same process was performed with 9000 byte MTU sizes and compared against the initial results.

4.2.1 Physical and Virtual Comparison

The throughput differences between physical and virtual iSCSI initiators in regards to the IET iSCSI target server were negligible. Figure 4.1 below displays the results of the test set in throughput.

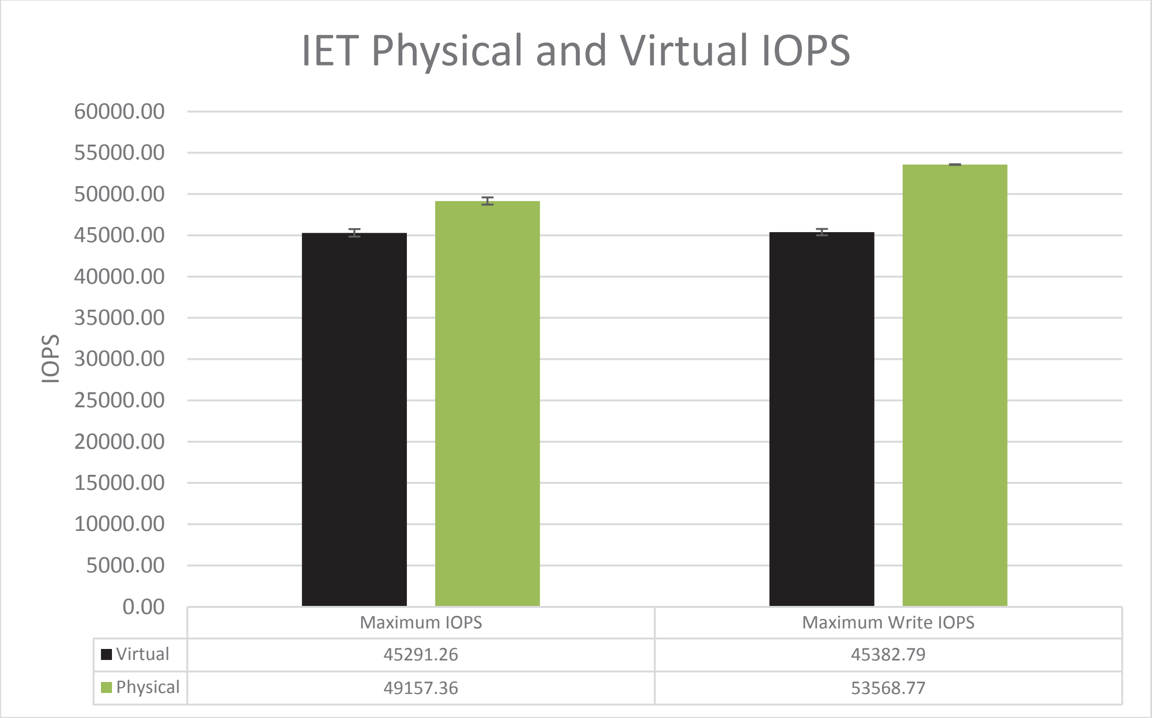


	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
■ Virtual	111.50	109.60	2.34	4.05	8.31	24.62	2.48	2.51
🟩 Physical	112.58	109.02	1.97	3.62	7.54	14.57	3.74	3.66

Figure 4.1 IET Physical and Virtual MB/s

The results for both tests regarding the maximum level of read/write throughput for this target server were near the maximum fabric rate found in previous iPerf results. In five of the eight results shown, the virtual machine appliance yielded slightly higher throughput rates, however, the actual rate difference in most results are negligible. The greatest difference between physical and virtual architectures was during the SQL 64K test which resulted in about 10 MB/s difference, or 68%.

During the same test set, IOPS were also measured and compared. Figure 4.2 below displays the results in IOPS obtained from the test set.



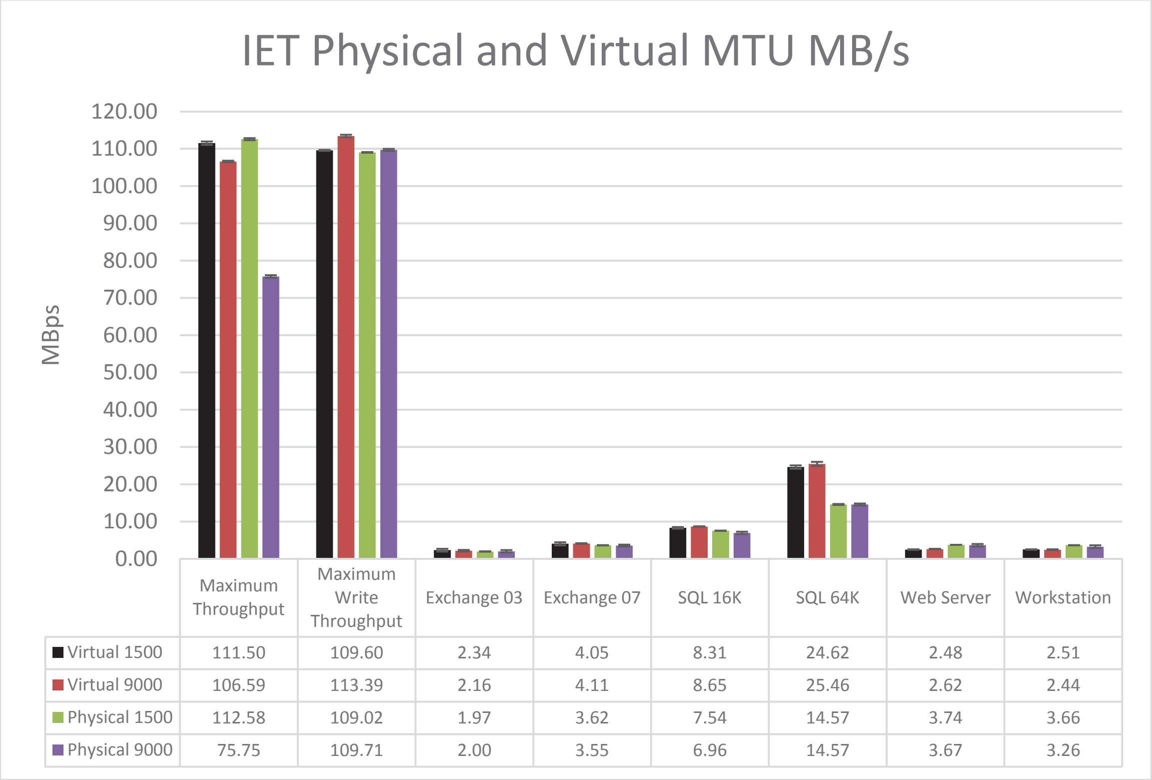
	Maximum IOPS	Maximum Write IOPS
■ Virtual	45291.26	45382.79
🟩 Physical	49157.36	53568.77

Figure 4.2 IET Physical and Virtual IOPS

As shown above, the physical operating system with software iSCSI initiator yielded higher IOPS values. Physical read IOPS values were higher by about 3800 IOPS, or about 8.5%. Physical maximum write IOPS values were higher by about 8000 IOPS, or 18%. The difference between physical and virtual initiators in this case is significant because the average of the virtual initiator does not overlap the standard deviation of the physical initiator.

4.2.2 MTU Size Comparison

After comparing physical and virtual architectures, the MTU size was changed on each initiator and IET target server NIC. The results comparing throughput (MB/s) between initiators are shown below in Figure 4.3. The figure graphically depicts the previous results of physical and virtual iSCSI initiators using standard frame sizes as well as the same testing architecture utilizing jumbo frames.

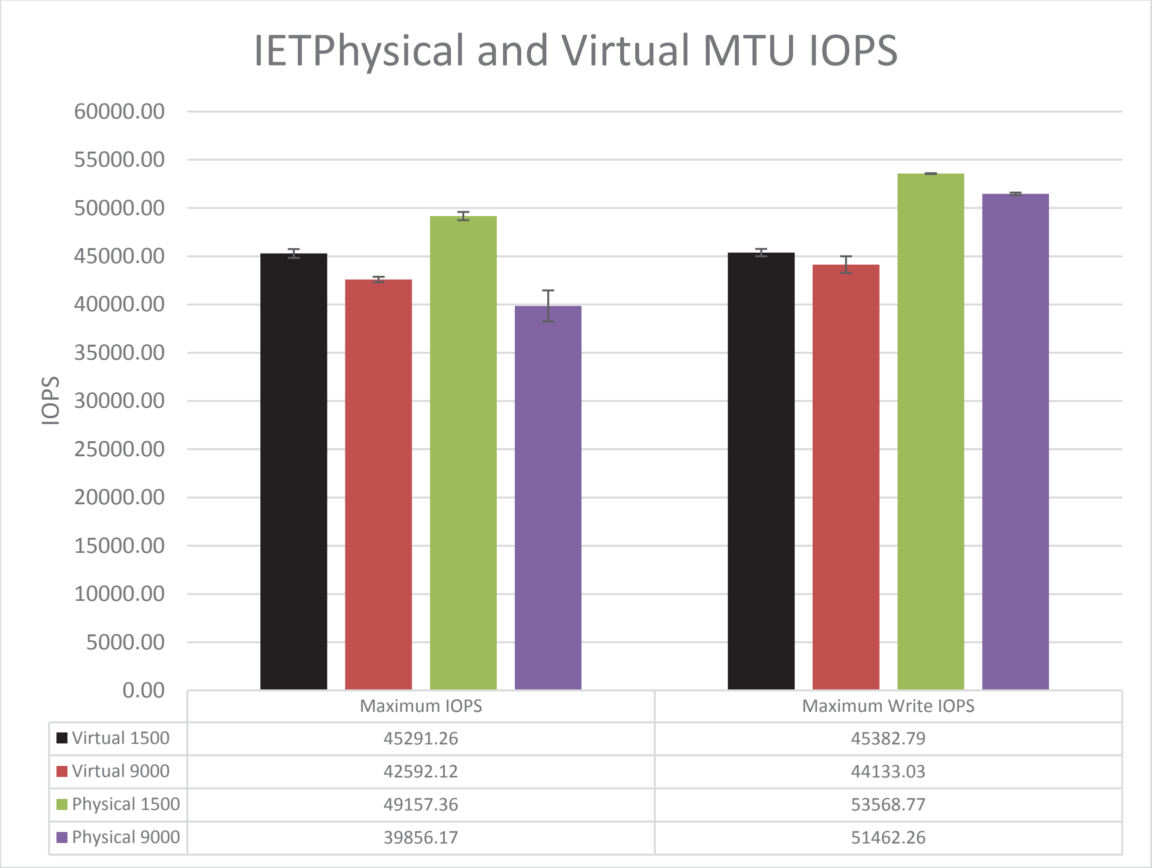


	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
⬛ Virtual 1500	111.50	109.60	2.34	4.05	8.31	24.62	2.48	2.51
🟥 Virtual 9000	106.59	113.39	2.16	4.11	8.65	25.46	2.62	2.44
🟩 Physical 1500	112.58	109.02	1.97	3.62	7.54	14.57	3.74	3.66
🟪 Physical 9000	75.75	109.71	2.00	3.55	6.96	14.57	3.67	3.26

Figure 4.3 IET Physical and Virtual MTU

The first two tests in Figure 4.3 yielded varying results. The first test measuring maximum read throughput favored standard MTU sizes in physical and virtual architectures. The introduction of 9000 byte MTU sizes degraded the read throughput performance. The maximum write throughput performance resulted in line speeds across all architectures. The remaining application specific test varied between architectures. Virtual architectures resulted in higher throughput values when using jumbo frames in four of the six tests. Physical architectures, however, resulted in higher application specific throughput values when using jumbo frames in two of the six tests.

The final test performed analyzed maximum IOPS values of standard frame and jumbo frame sizes. The data from the test performed is shown below in Figure 4.4.



	Maximum IOPS	Maximum Write IOPS
⬛ Virtual 1500	45291.26	45382.79
🟥 Virtual 9000	42592.12	44133.03
🟩 Physical 1500	49157.36	53568.77
🟪 Physical 9000	39856.17	51462.26

Figure 4.4 IET Physical and Virtual MTU IOPS

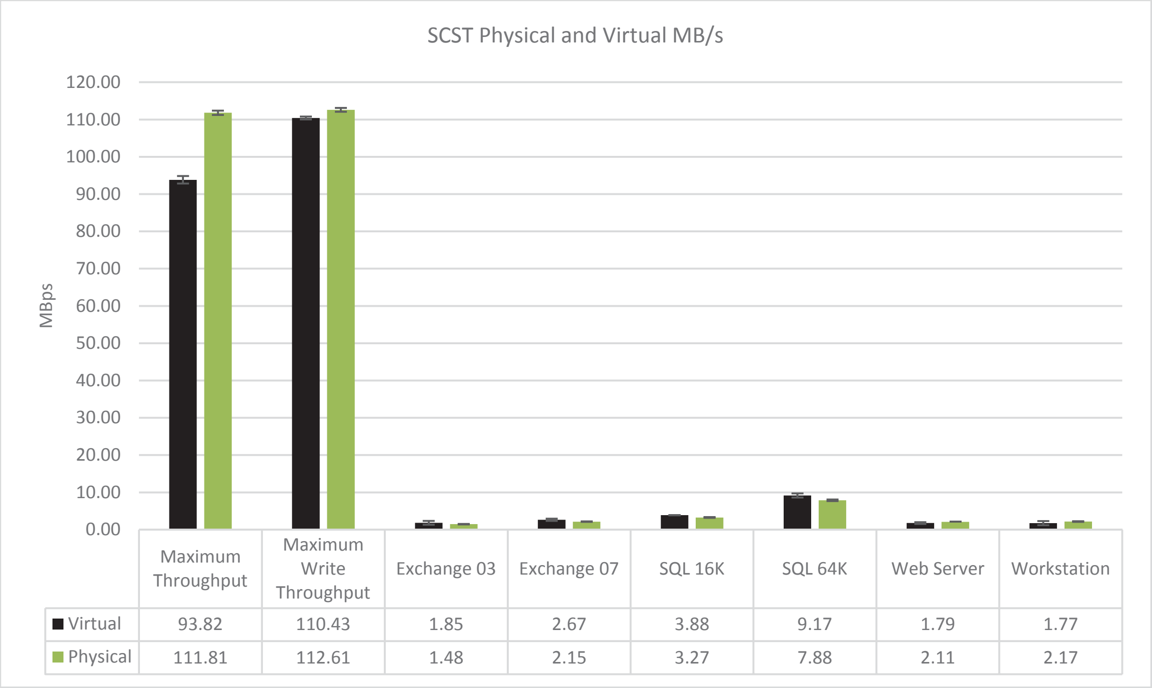
In the MTU tests above, the previous results utilizing standard frames are compared to the results obtained using jumbo frames. Standard frame sizes yielded higher values regardless of the architecture examined. This pattern is similar to the results found when comparing throughput and varying MTU sizes. Standard frame sizes in a virtual environment exceeded jumbo frame results by 6.3% and 2.6% for read and write values, respectively. The difference between standard and jumbo frame sizes for virtual initiators concerning maximum write IOPS is about 1200 IOPS, or 3%. Additionally, the difference between standard and jumbo frame sizes for physical initiators in regards to maximum read and write values were 23% and 4.0%, respectively.

4.3 SCST iSCSI Target Results

To compare physical and virtual architectures against the SCST iSCSI target server, a standard series of tests were performed to obtain values. First, the throughput and IOPS of both systems were compared. Then, the same process was performed with 9000 byte MTU sizes and compared against the initial results.

4.3.1 Physical and Virtual Comparison

The test set utilizing physical and virtual architectures with SCST iSCSI provided mixed results. Maximum throughput values were higher using the physical architecture during initial tests. The virtual architecture application specific results were higher in four of the six tests. The results from this test set are shown below in Figure 4.5.

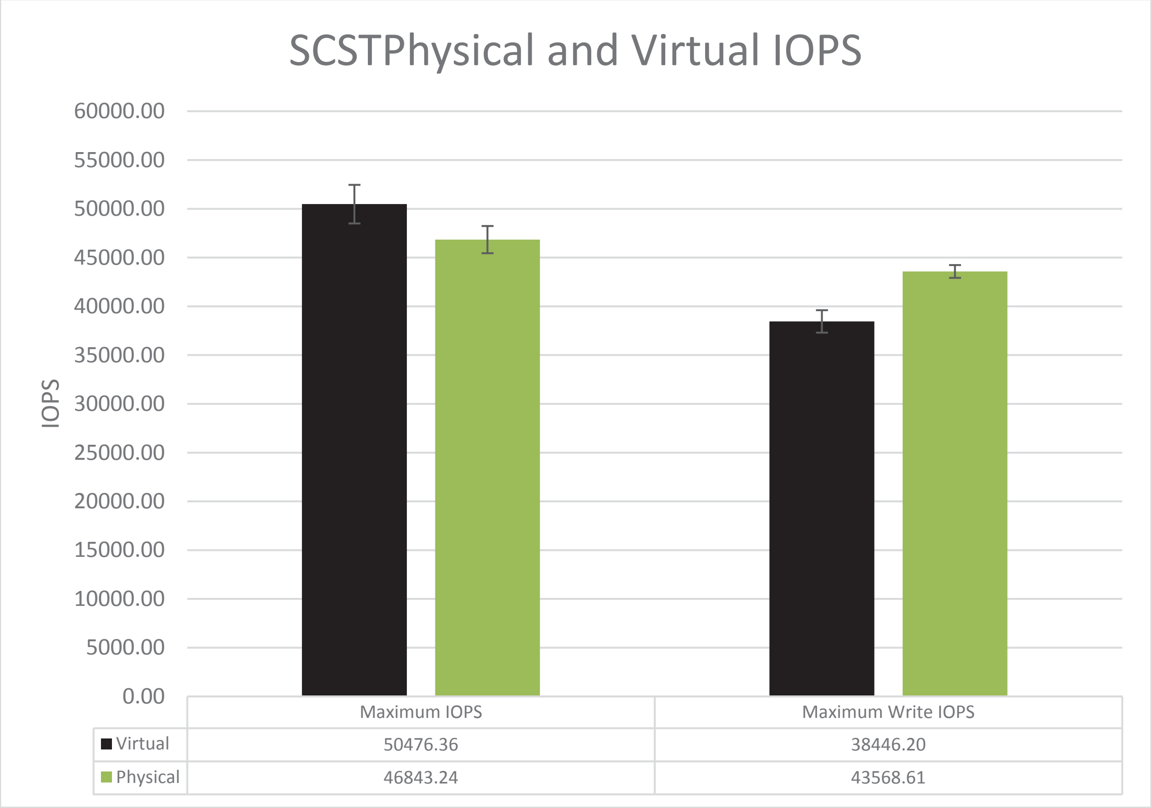


	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
■ Virtual	93.82	110.43	1.85	2.67	3.88	9.17	1.79	1.77
🟩 Physical	111.81	112.61	1.48	2.15	3.27	7.88	2.11	2.17

Figure 4.5 SCST Physical and Virtual MB/s

The data from Figure 4.5 shows only slight throughput differences for the last six tests. However, the first two tests concerning maximum read and write throughput favor the physical machine over the virtual appliance. Throughput read speeds differ by about 18 MB while write speeds only differ by about 2 MB.

The results in Figure 4.6 below also show varying results. Read IOPS values favor virtual architectures while write IOPS speeds favor physical architectures. Observing the results from the data shows a clear difference between the architectures. The maximum read IOPS between architectures differ by about 36,000 IOPS while maximum write IOPS differ by about 51,000 IOPS. The percentage differences between physical and virtual initiators for read and write IOPS are 7.7% and 13.3%.

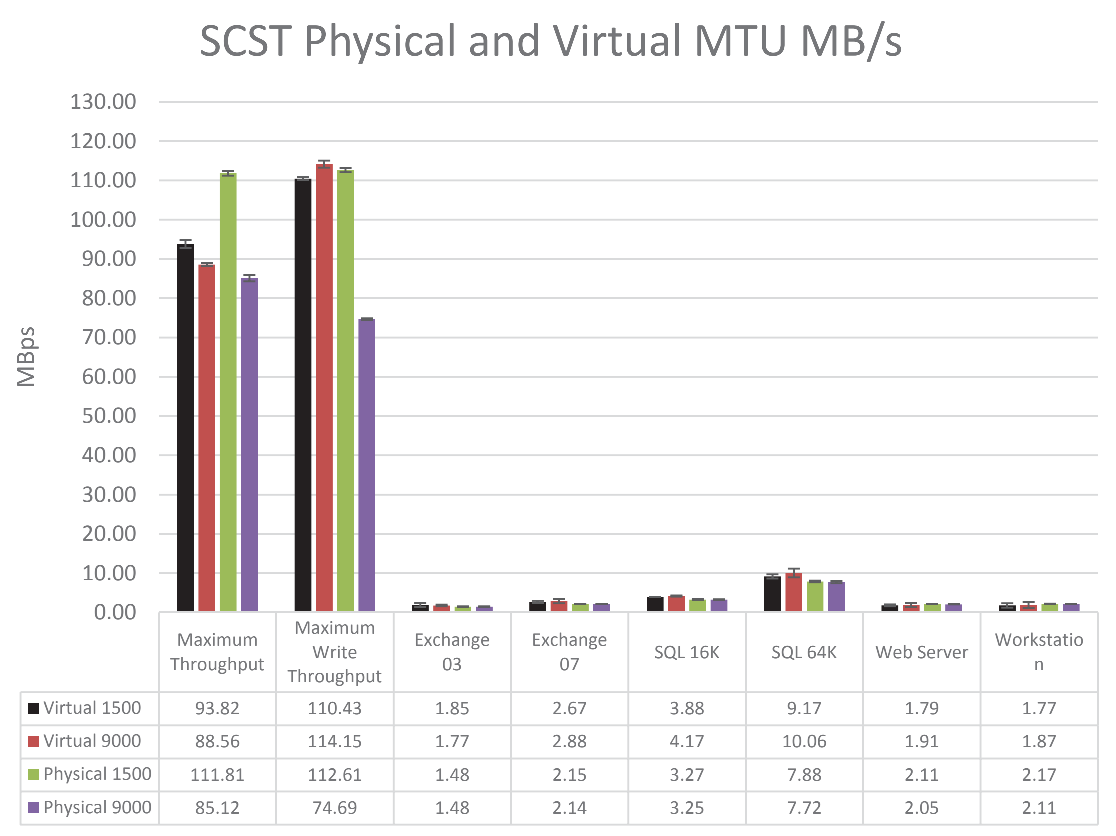


	Maximum IOPS	Maximum Write IOPS
■ Virtual	50476.36	38446.20
🟩 Physical	46843.24	43568.61

Figure 4.6 SCST Physical and Virtual IOPS

4.3.2 MTU Size Comparison

The MTU size of standard and jumbo frames were tested using the same SCST iSCSI SAN. Once the data was collected, the results were compared to determine if jumbo frames increased throughput speeds for virtual and physical architectures. The data obtained is shown in Figure 4.7 below.

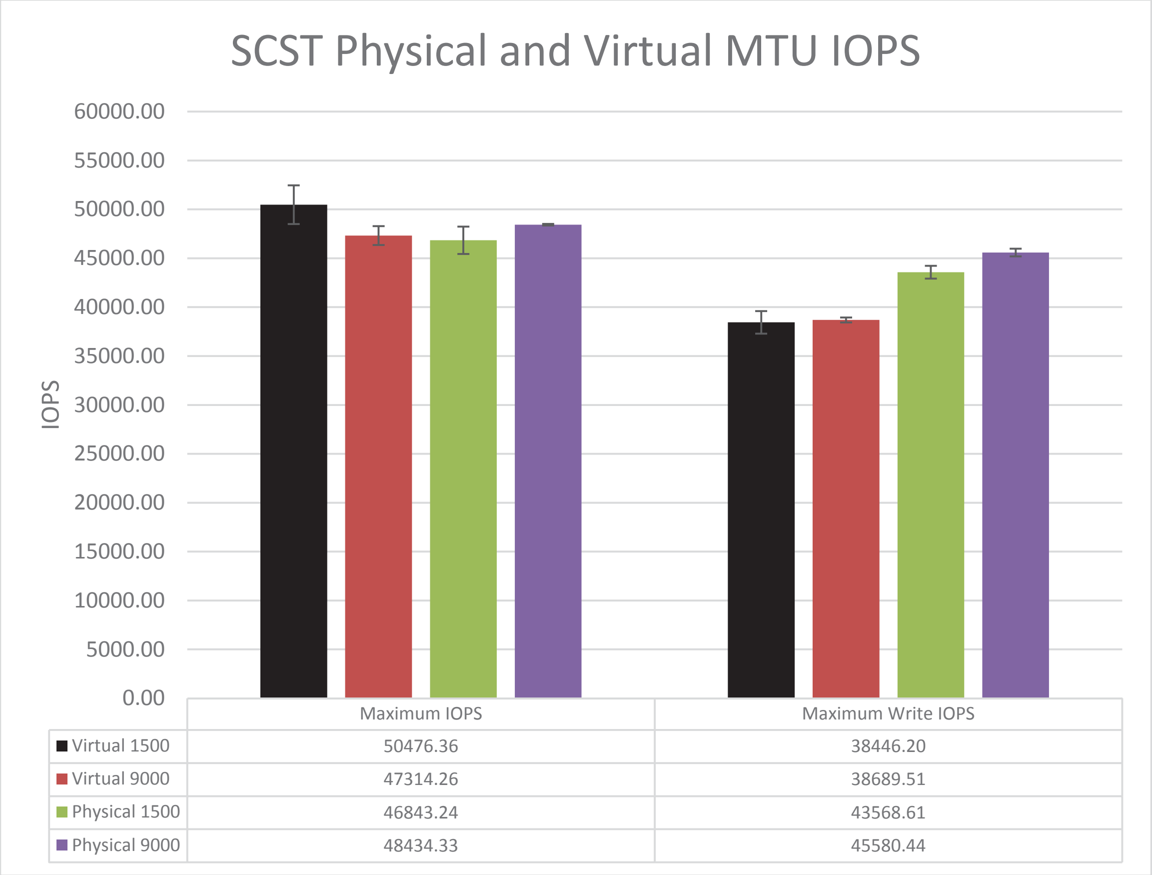


	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
⬛ Virtual 1500	93.82	110.43	1.85	2.67	3.88	9.17	1.79	1.77
🟥 Virtual 9000	88.56	114.15	1.77	2.88	4.17	10.06	1.91	1.87
🟩 Physical 1500	111.81	112.61	1.48	2.15	3.27	7.88	2.11	2.17
🟪 Physical 9000	85.12	74.69	1.48	2.14	3.25	7.72	2.05	2.11

Figure 4.7 SCST Physical and Virtual MTU

The data in Figure 4.7 provides varying results. In the first two tests comparing maximum throughput scenarios, standard frame sizes produced higher values in all maximum physical architecture instances. The virtual architecture resulted in higher values with standard frames during maximum read throughput tests and higher values with jumbo frames in maximum write throughput tests. The only test using jumbo frames which saturated the fabric link was the virtual maximum write throughput test. The remaining application specific tests produced mixed results for each architecture. The physical architecture favored standard frames in all six tests. The virtual architecture, however, resulted in higher values in five of six tests utilizing jumbo frames. It should be noted the actual differences between standard and jumbo frame sizes are negligible.

Following throughput tests, IOPS were also measured using standard and jumbo frame sizes. The virtual architecture using standard frame sizes yielded higher results when measuring read IOPS values and lower results when measuring write IOPS values. Alternately, the physical architecture tested slightly favored jumbo frame sizes in read and write tests. The complete data set is shown graphically in Figure 4.8 below.



	Maximum IOPS	Maximum Write IOPS
⬛ Virtual 1500	50476.36	38446.20
🟥 Virtual 9000	47314.26	38689.51
🟩 Physical 1500	46843.24	43568.61
🟪 Physical 9000	48434.33	45580.44

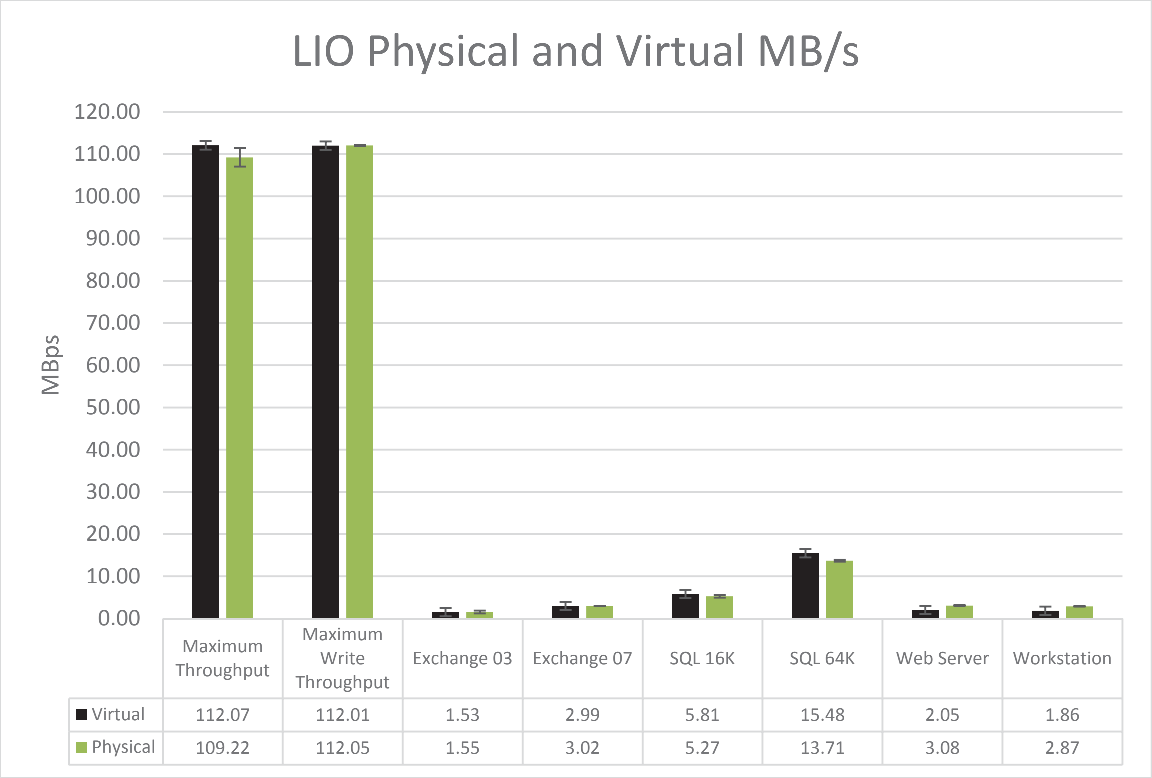
Figure 4.8 SCST Physical and Virtual MTU IOPS

4.4 LIO iSCSI Target Results

To compare physical and virtual architectures against the LIO iSCSI target server, a series of tests were performed to obtain and compare values. First, the throughput and IOPS of both systems were compared. Then, the same process was performed with 9000 byte MTU sizes and compared against the initial results.

4.4.1 Physical and Virtual Comparison

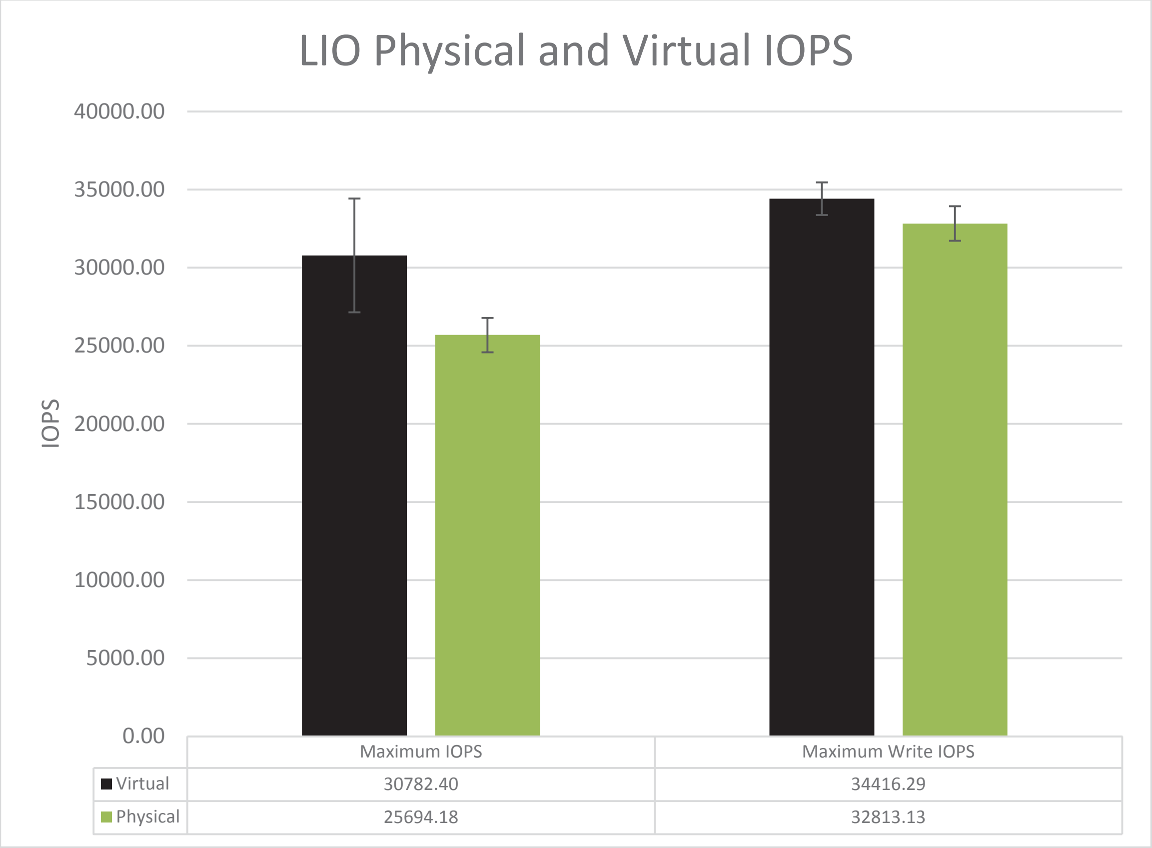
After comparing the physical and virtual initiator results from the test set, it was determined the amount of difference between the two architectures were negligible. The maximum throughput tests, both read and write, resulted in near line speeds for both architectures. In addition, the last six tests simulating different applications did not exhibit a notable difference. The results are depicted graphically in Figure 4.9 below.



	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
■ Virtual	112.07	112.01	1.53	2.99	5.81	15.48	2.05	1.86
🟩 Physical	109.22	112.05	1.55	3.02	5.27	13.71	3.08	2.87

Figure 4.9 LIO Physical and Virtual MB/s

The results measuring IOPS between virtual and physical initiators are shown below in Figure 4.10. The results show the virtual architecture with higher values. The actual difference between physical and virtual architectures for read and write values is 5000 IOPS and 1600 IOPS. Read differences equate to nearly 19%, while write differences were minimal at 5%.

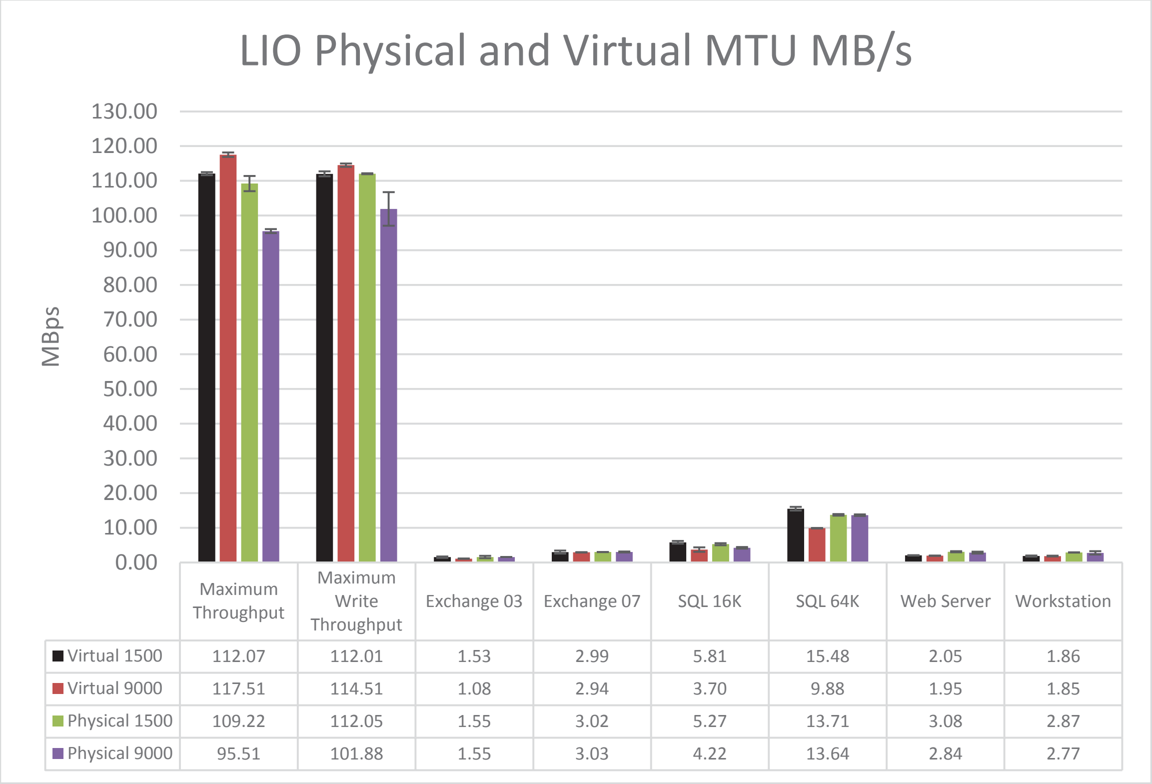


	Maximum IOPS	Maximum Write IOPS
■ Virtual	30782.40	34416.29
🟩 Physical	25694.18	32813.13

Figure 4.10 LIO Physical and Virtual IOPS

4.4.2 MTU Size Comparison

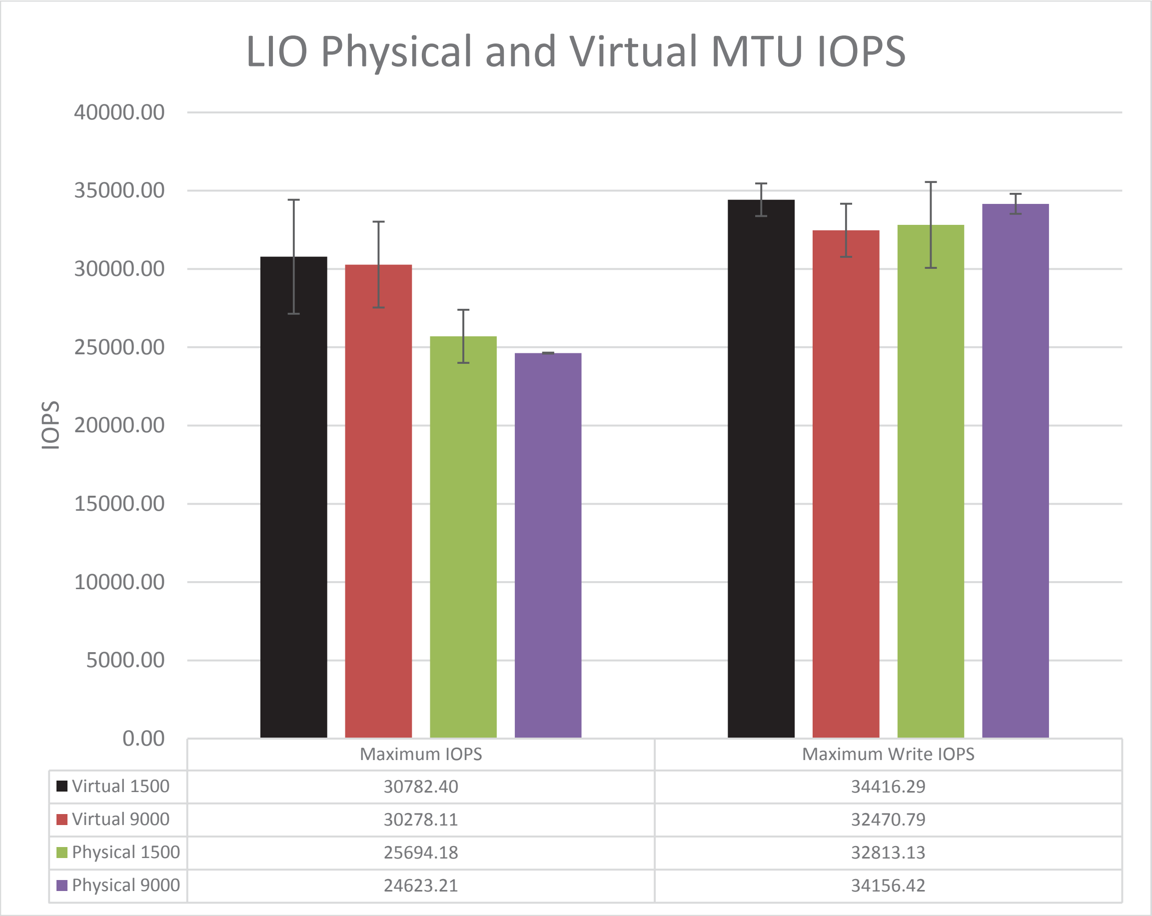
Altering the MTU frame size provided minimally varying results. Maximum throughput tests resulted in higher virtual architecture values when using jumbo frames. The physical architecture, however, resulted in higher values using standard frames for both maximum throughput tests. It should be noted the maximum throughput values for the virtual initiator are all at line speed. The physical environment, however, exhibited a decrease in throughput by over 10 MB/s when using jumbo frames. From a virtual and physical initiator aspect, standard frames yielded slightly higher values in all application specific tests. The complete set of data comparing MTU sizes and architectures is depicted below in Figure 4.11.



	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
■ Virtual 1500	112.07	112.01	1.53	2.99	5.81	15.48	2.05	1.86
🟥 Virtual 9000	117.51	114.51	1.08	2.94	3.70	9.88	1.95	1.85
🟩 Physical 1500	109.22	112.05	1.55	3.02	5.27	13.71	3.08	2.87
🟪 Physical 9000	95.51	101.88	1.55	3.03	4.22	13.64	2.84	2.77

Figure 4.11 LIO Physical and Virtual MTU MB/s

Following throughput tests, IOPS were then measured and compared. In three out of four comparisons, standard frame sizes resulted in higher IOPS values. The physical initiator test utilizing jumbo frames was the only test which jumbo frame throughput values were slightly higher than the standard frame size results. Due to averages overlapping the standard deviations, the differences between frame sizes are negligible. The complete set of results from standard and jumbo frame sizes measuring IOPS are depicted graphically below in Figure 4.12.



	Maximum IOPS	Maximum Write IOPS
■ Virtual 1500	30782.40	34416.29
🟥 Virtual 9000	30278.11	32470.79
🟩 Physical 1500	25694.18	32813.13
🟪 Physical 9000	24623.21	34156.42

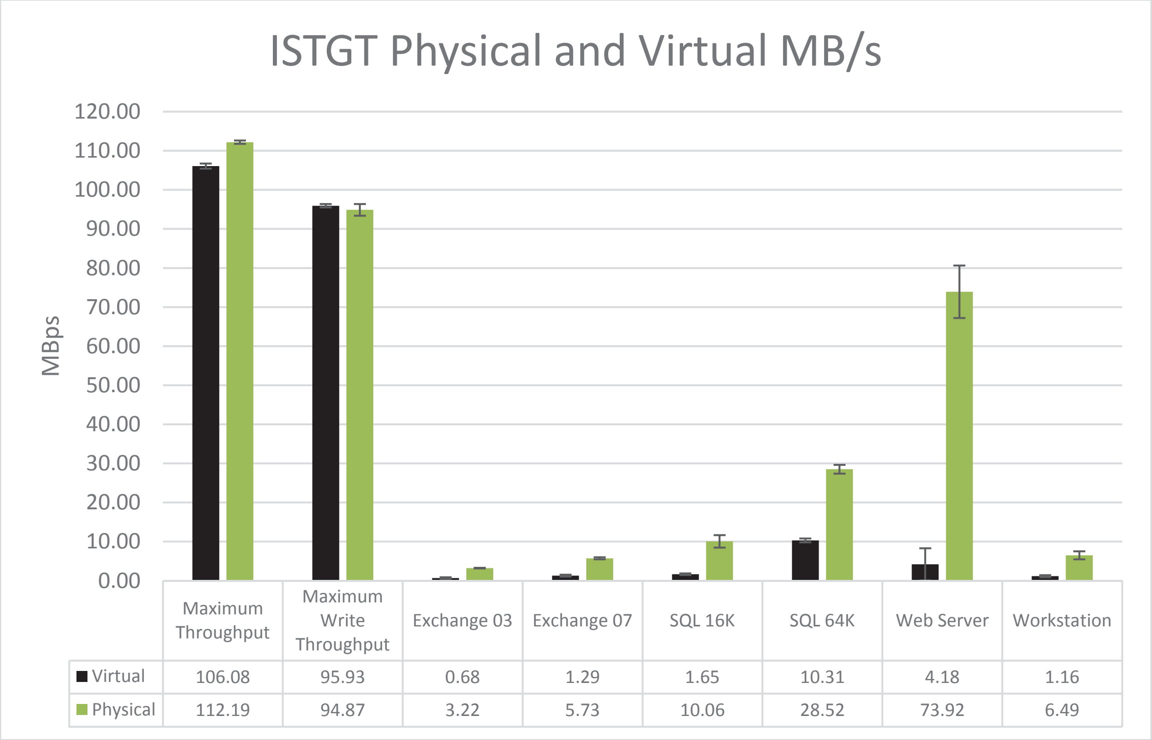
Figure 4.12 LIO Physical and Virtual MTU IOPS

4.5 <u>ISTGT iSCSI Target Results</u>

The final experiment tested the ISTGT iSCSI target server. To compare physical and virtual architectures, a series of tests were performed to obtain metrics. To begin, the throughput and IOPS of both architectures were compared. Then, the same process was performed with 9000 byte MTU sizes and compared against the initial results.

4.5.1 Physical and Virtual Comparison

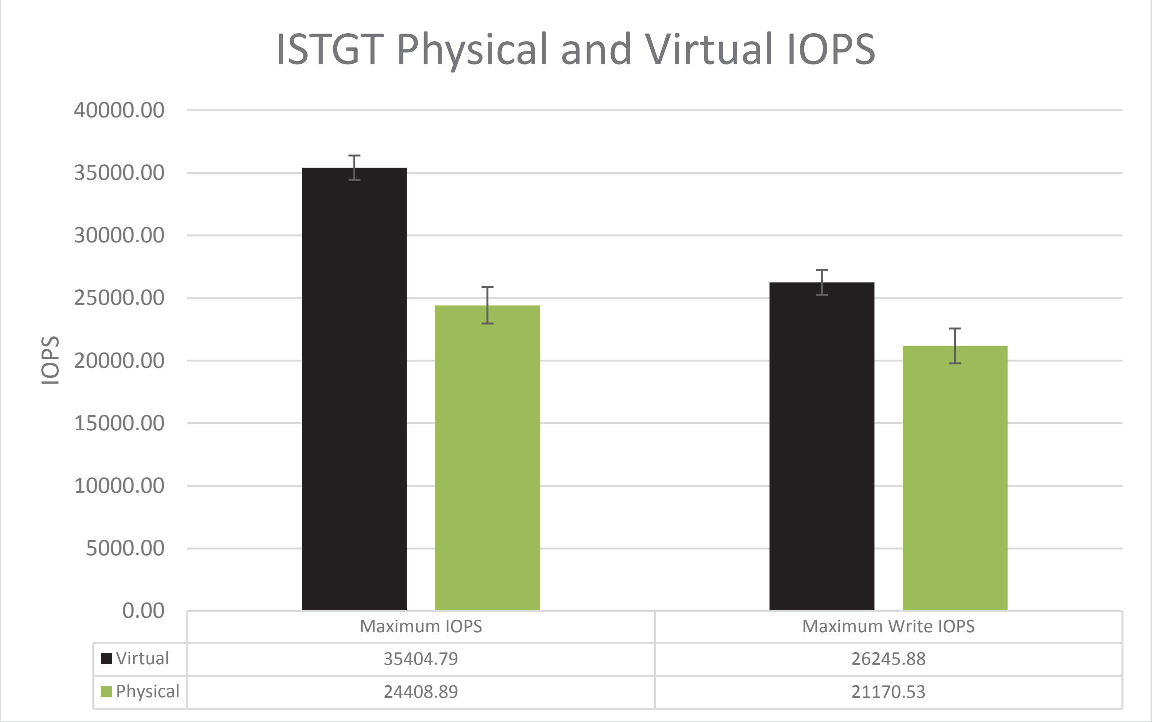
The comparison of physical and virtual architectures utilizing the ISTGT SAN are depicted below in Figure 4.13. The results favor the physical architecture in application specific tests with notable throughput differences. While application specific tests have varying differences, the first two tests which measuring maximum throughput speeds are comparable in both architectures. The actual difference between physical and virtual maximum read throughput is about 6 MB/s, or 6%. The actual difference between virtual and physical maximum write throughput is neglible. It should be noted the physical initiator produced application specific results with skewed values do to caching. The physical initiator and Iometer encountered a limitation which did not allow Iometer to properly saturate the memory in the SAN. These cached results are apparent in the remaining physical initiator application specific results referencing ISTGT.



	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
■ Virtual	106.08	95.93	0.68	1.29	1.65	10.31	4.18	1.16
■ Physical	112.19	94.87	3.22	5.73	10.06	28.52	73.92	6.49

Figure 4.13 ISTGT Physical and Virtual MB/s

Tests comparing IOPS values resulted in virtual architectures with higher results. In both read/write IOPS tests, the virtual architecture was notably higher than the physical initiator. The difference in maximum read values is about 11,000 IOPS, or 45%. Write values differed by about 5000 IOPS, or 23%. The large variance in values suggest a marginal difference between physical and virtual initiators. The results are depicted graphically in Figure 4.14 below.

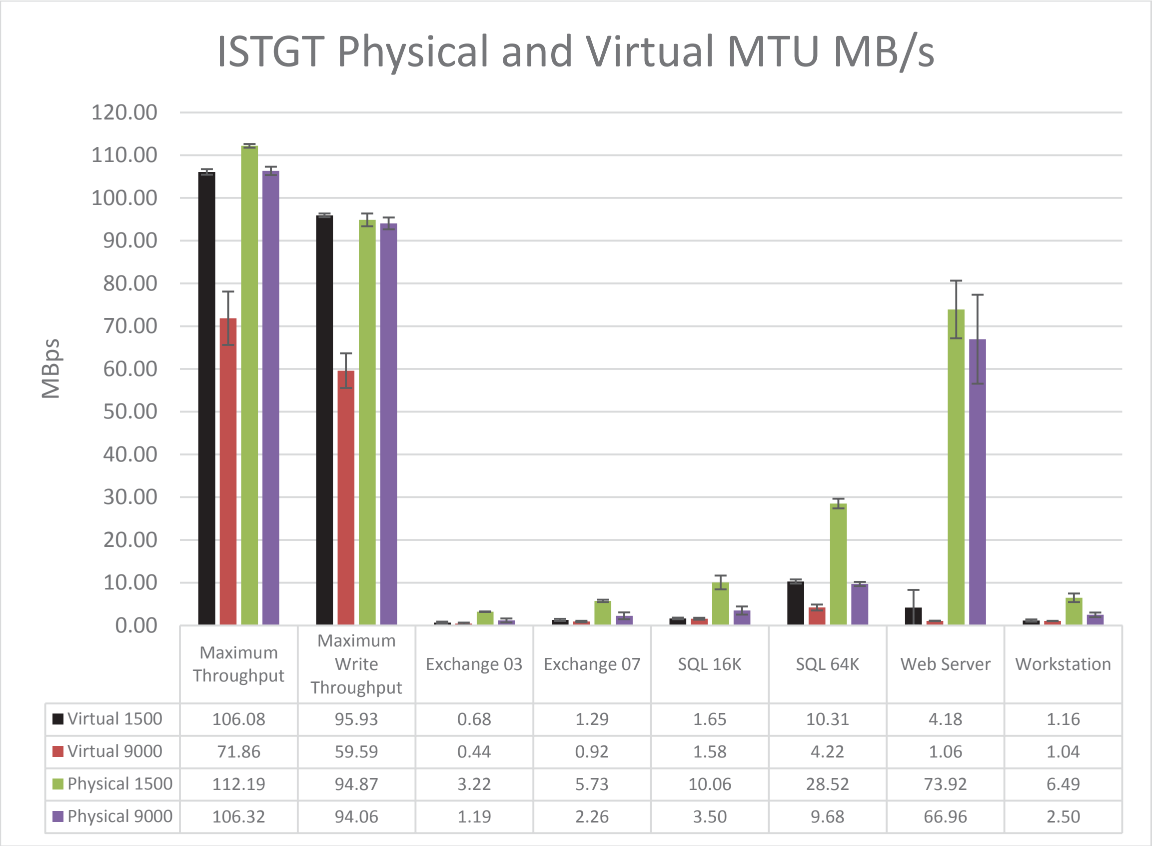


	Maximum IOPS	Maximum Write IOPS
■ Virtual	35404.79	26245.88
🟩 Physical	24408.89	21170.53

Figure 4.14 ISTGT Physical and Virtual IOPS

4.5.2 MTU Size Comparison

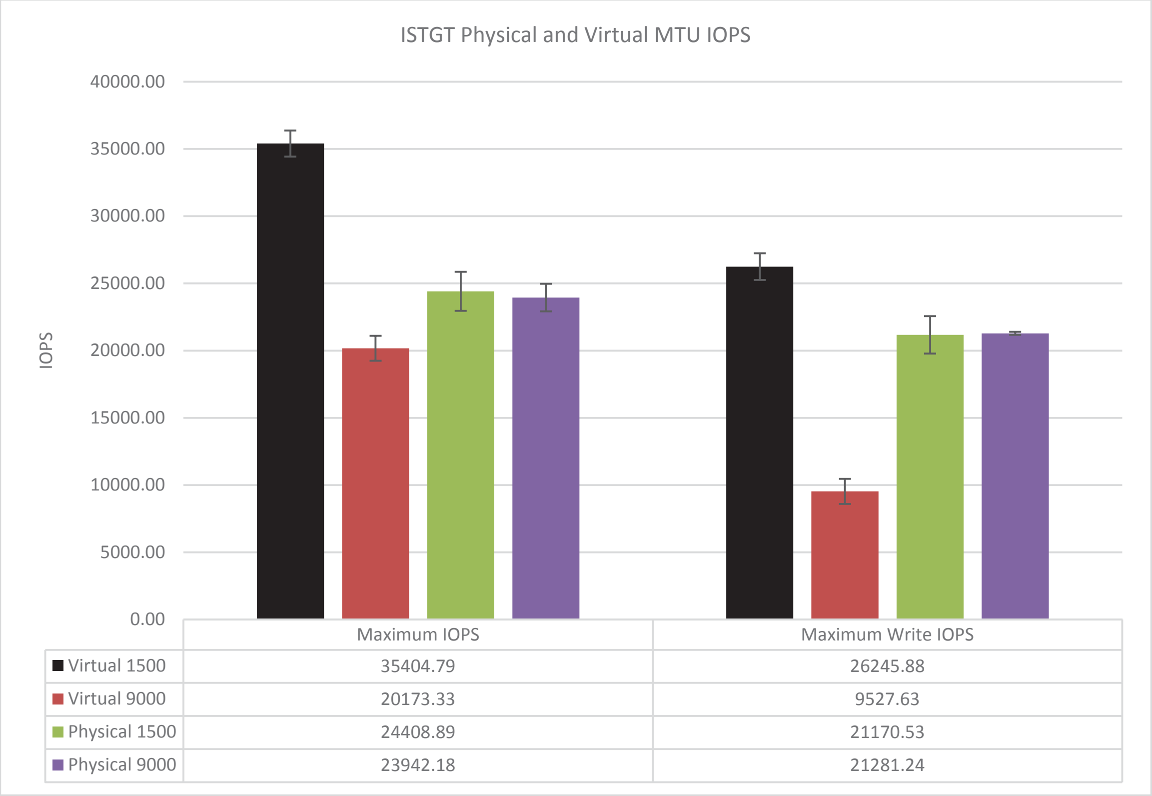
Increasing the MTU size and performing the test set resulted in degraded performance compared to standard frames. In general, tests measuring maximum read/write throughput concluded standard frame sizes resulted in higher values compared to jumbo frames. In some cases, such as the maximum throughput tests, jumbo frames degraded performance values significantly in virtual architectures. Notable differences specifically come from the virtual architecture in which maximum read and write throughput speeds differ by almost 35 MB/s, or 47%. In addition, standard frame sizes in virtual and physical architectures resulted in higher values during application specific tests.The complete results are shown below in Figure 4.15.



	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
⬛ Virtual 1500	106.08	95.93	0.68	1.29	1.65	10.31	4.18	1.16
🟥 Virtual 9000	71.86	59.59	0.44	0.92	1.58	4.22	1.06	1.04
🟩 Physical 1500	112.19	94.87	3.22	5.73	10.06	28.52	73.92	6.49
🟪 Physical 9000	106.32	94.06	1.19	2.26	3.50	9.68	66.96	2.50

Figure 4.15 ISTGT Physical and Virtual MTU MB/s

Similarly, tests measuring IOPS values also favored standard frame sizes over jumbo frames. The difference between frame sizes using virtual architectures varied by over 15,000 IOPS, or 75%, in maximum read tests. Maximum write tests varies by over 20,000 IOPS which is a significant difference. The results for physical initiators were much more consistent and the differences were negligible. The complete results are shown below in Figure 4.16.



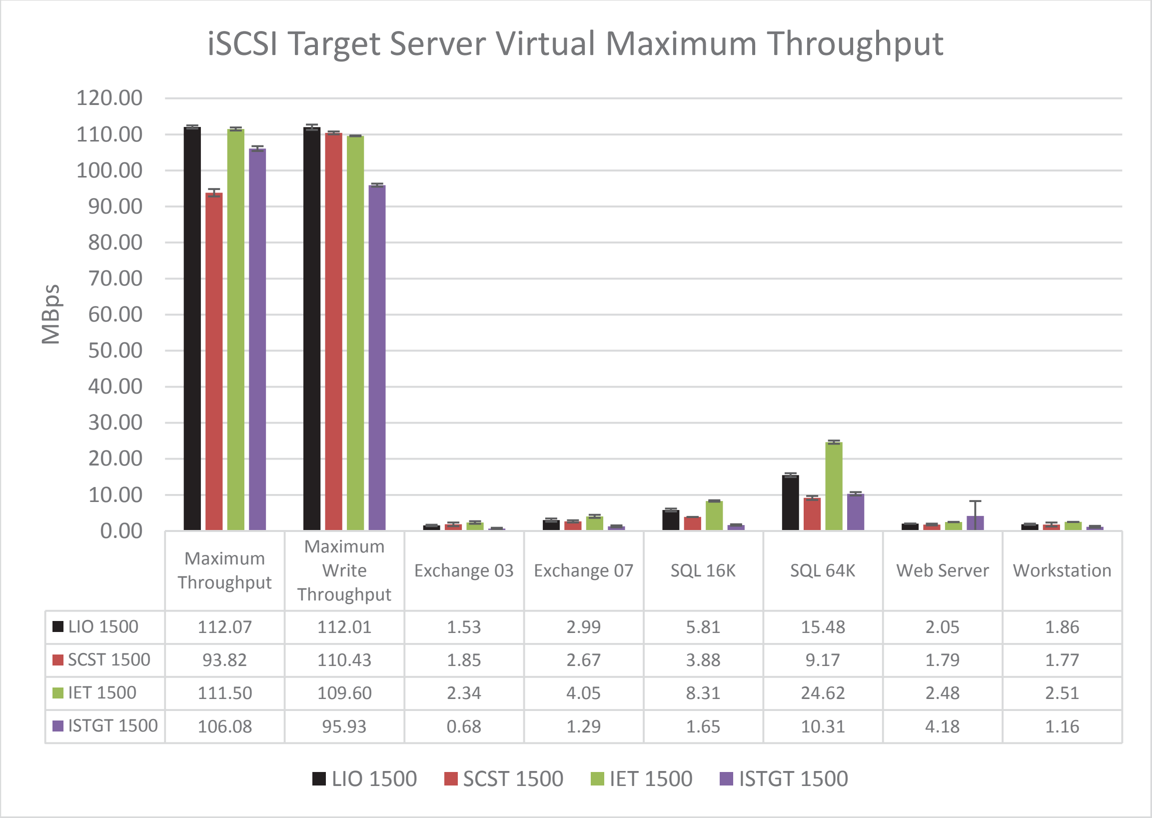
	Maximum IOPS	Maximum Write IOPS
■ Virtual 1500	35404.79	26245.88
■ Virtual 9000	20173.33	9527.63
■ Physical 1500	24408.89	21170.53
■ Physical 9000	23942.18	21281.24

Figure 4.16 ISTGT Physical and Virtual MTU IOPS

4.6 iSCSI Target Server Comparisons

After all results were obtained from the four experiments and relative test sets, the results from each SAN Target were compared. It was determined the tests used to compare SANs consist only of standard frame sizes. Standard frame sizes were selected due to the consistent resultant values of previous tests compared to the results from jumbo frames. In addition, some test sets exhibited notable differences between physical and virtual architectures with standard frames. Due to the variation in values, physical and virtual architectures are both used to compare SANs. Final comparisons consist of only comparing maximum throughput and IOPS values among physical and virtual architectures.

The results from standard frame sizes measuring virtual iSCSI initiator throughput are depicted graphically in Figure 4.17 below. Although most values do not exhibit much variation, the highest values in maximum throughput tests were part of the LIO iSCSI target. LIO, SCST, and IET all had values which were close to the line speed of the network fabric. ISTGT nearly approached line speeds at 106.08 MB/s. The LIO iSCSI target did not yield the highest values compared to other SANs in application specific test. Additionally, the results from LIO were comparable to the highest application specific values which were from IET.



	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
■ LIO 1500	112.07	112.01	1.53	2.99	5.81	15.48	2.05	1.86
■ SCST 1500	93.82	110.43	1.85	2.67	3.88	9.17	1.79	1.77
■ IET 1500	111.50	109.60	2.34	4.05	8.31	24.62	2.48	2.51
■ ISTGT 1500	106.08	95.93	0.68	1.29	1.65	10.31	4.18	1.16

Figure 4.17 iSCSI Target Service Virtual Maximum Throughput

The next comparison evaluated the IOPS values of a virtual environment among the four different SANs. The results show IET and SCST as the targets with highest IOPS values. The complete comparison is shown below in Figure 4.18. The largest separation in read IOPS values was between LIO and SCST at nearly 20,000 IOPS, or 64%. The largest separation between write IOPS values was between IET and ISTGT at nearly 42,000 IOPS, or 72%.

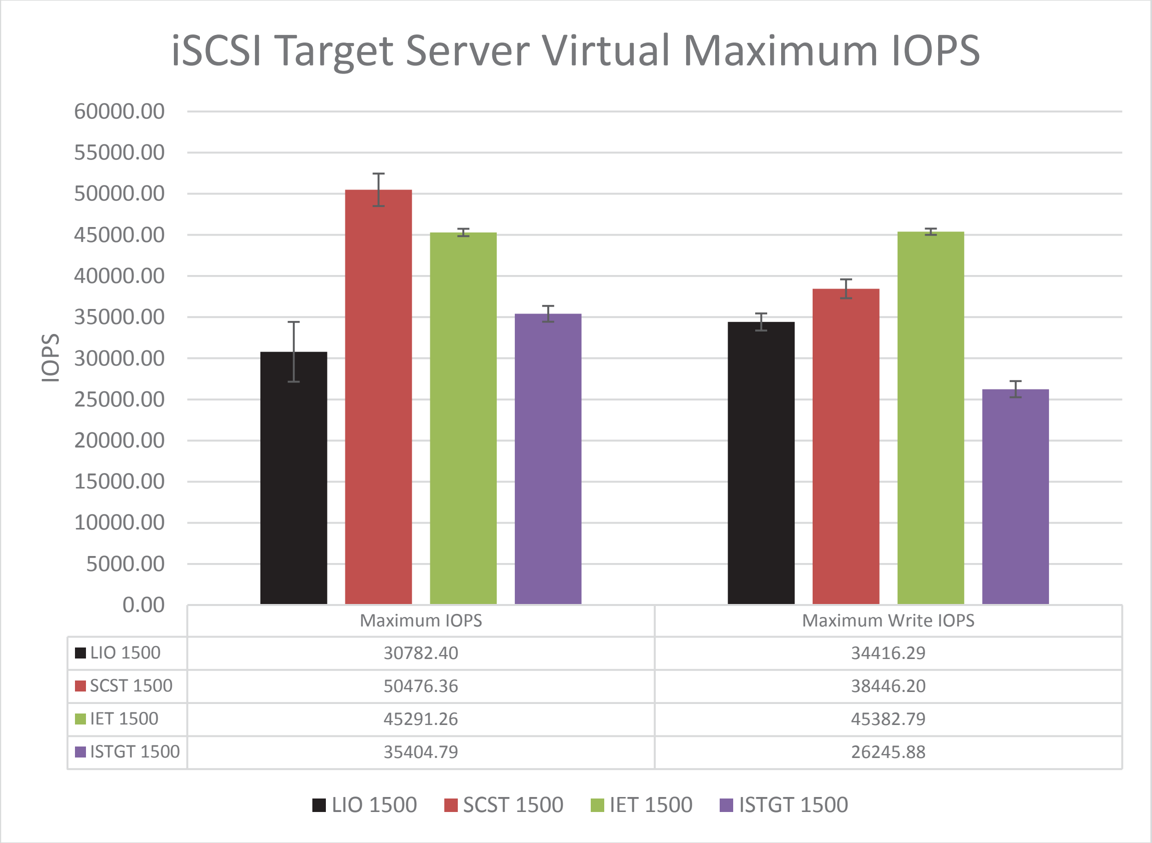


Figure 4.18 iSCSI Target Server Virtual Maximum IOPS

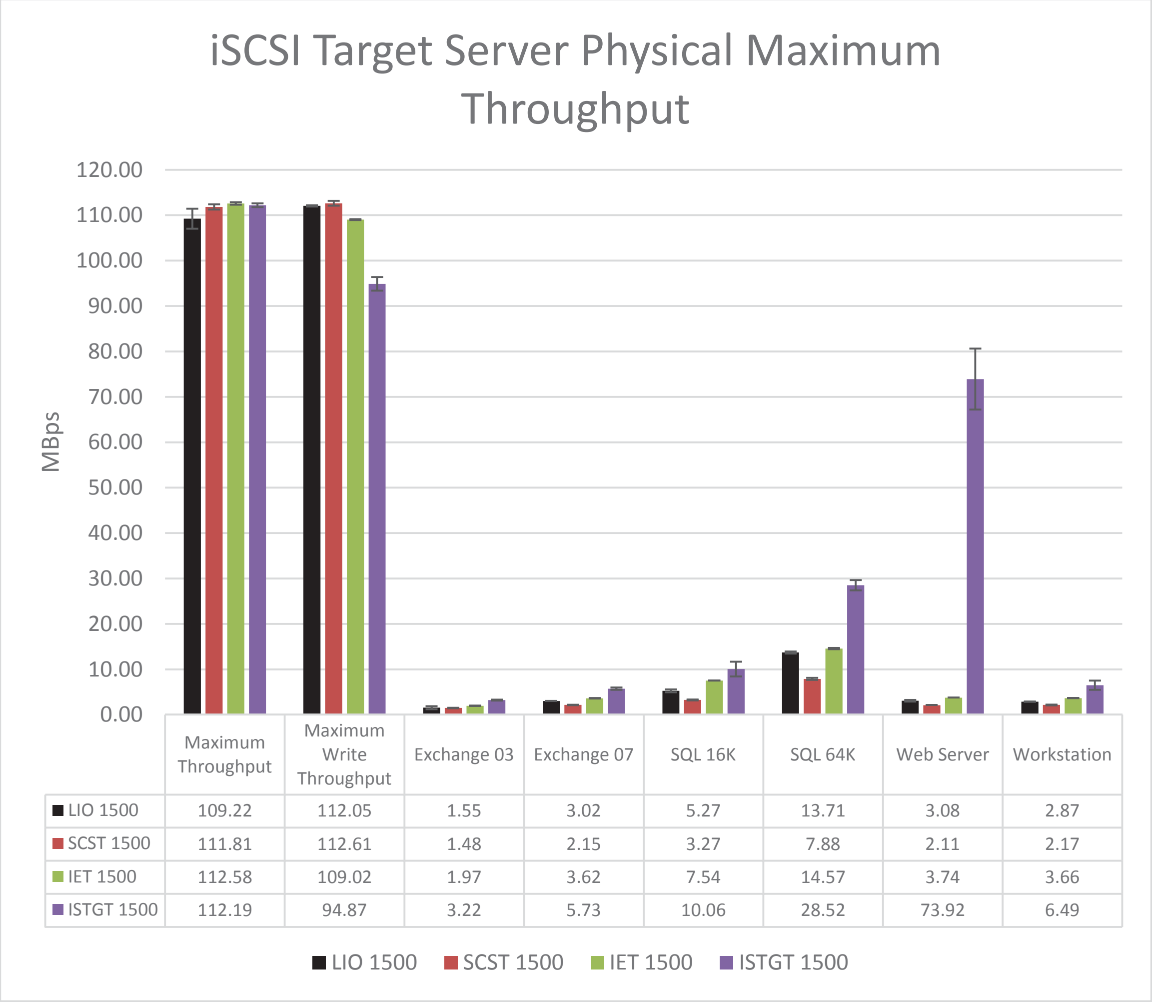
A complete reference of actual and percentage changes are shown below in Table 4.3.

Table 4.3 Virtual Maximum Read and Right IOPS Differences

Virtual Maximum Read IOPS Comparison	Actual Difference	Percentage Difference
SCST to IET	5185.10	11%
SCST to LIO	19693.96	64%
SCST to ISTGT	15071.58	43%

Virtual Maximum Write IOPS Comparison	Actual Difference	Percentage Difference
IET to SCST	6936.60	18%
IET to LIO	10966.51	32%
IET to ISTGT	19136.91	73%

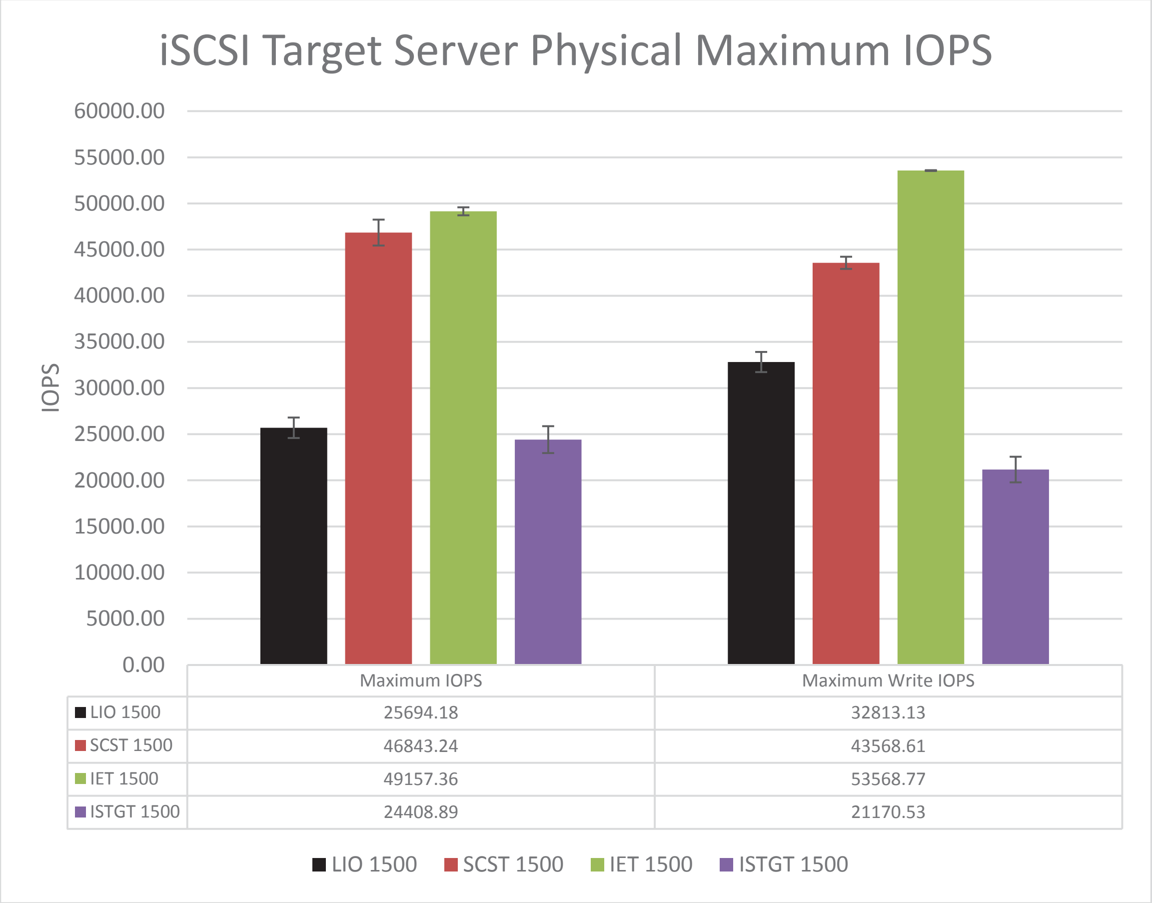
Additional comparisons consisted of throughput and IOPS values of the physical architecture among all SANS tested. The data for throughput comparison is shown in Figure 4.19 below. This figure shows the maximum values for read throughput were from IET,SCST, and ISTGT. In addition, the maximum throughput values for write tests were from LIO, SCST, and IET. The application specific tests clearly favored ISTGT, however, many results such as the web server infers the results were due to cached values.



	Maximum Throughput	Maximum Write Throughput	Exchange 03	Exchange 07	SQL 16K	SQL 64K	Web Server	Workstation
■ LIO 1500	109.22	112.05	1.55	3.02	5.27	13.71	3.08	2.87
■ SCST 1500	111.81	112.61	1.48	2.15	3.27	7.88	2.11	2.17
■ IET 1500	112.58	109.02	1.97	3.62	7.54	14.57	3.74	3.66
■ ISTGT 1500	112.19	94.87	3.22	5.73	10.06	28.52	73.92	6.49

Figure 4.19 iSCSI Target Server Physical Maximum Throughput

Regarding IOPS values between SANS from a physical initiator, IET and SCST again yielded the highest values. The largest separation of read IOPS values was between IET and ISTGT at nearly 24,500 IOPS, over a 100% difference. The largest separation of write IOPS values was between IET and ISTGT at nearly 32,000 IOPS, over a 100% difference. The differences between the two highest SANS, SCST and IET, were much smaller. Maximum read IOPS values between the targets differed by 2300 IOPS, or 4.9%. The difference between IET and SCST write values was 10,000 IOPS, or 22%. The complete comparison of IOPS values between SANs can be seen below in Figure 4.20.



	Maximum IOPS	Maximum Write IOPS
⬛ LIO 1500	25694.18	32813.13
🟥 SCST 1500	46843.24	43568.61
🟩 IET 1500	49157.36	53568.77
🟪 ISTGT 1500	24408.89	21170.53

Figure 4.20 iSCSI Target Server Physical Maximum IOPS

A complete reference of actual and percentage changes are shown below in Table 4.4.

Table 4.4 Physical Maximum Read and Right IOPS Differences

Physical Maximum Read IOPS

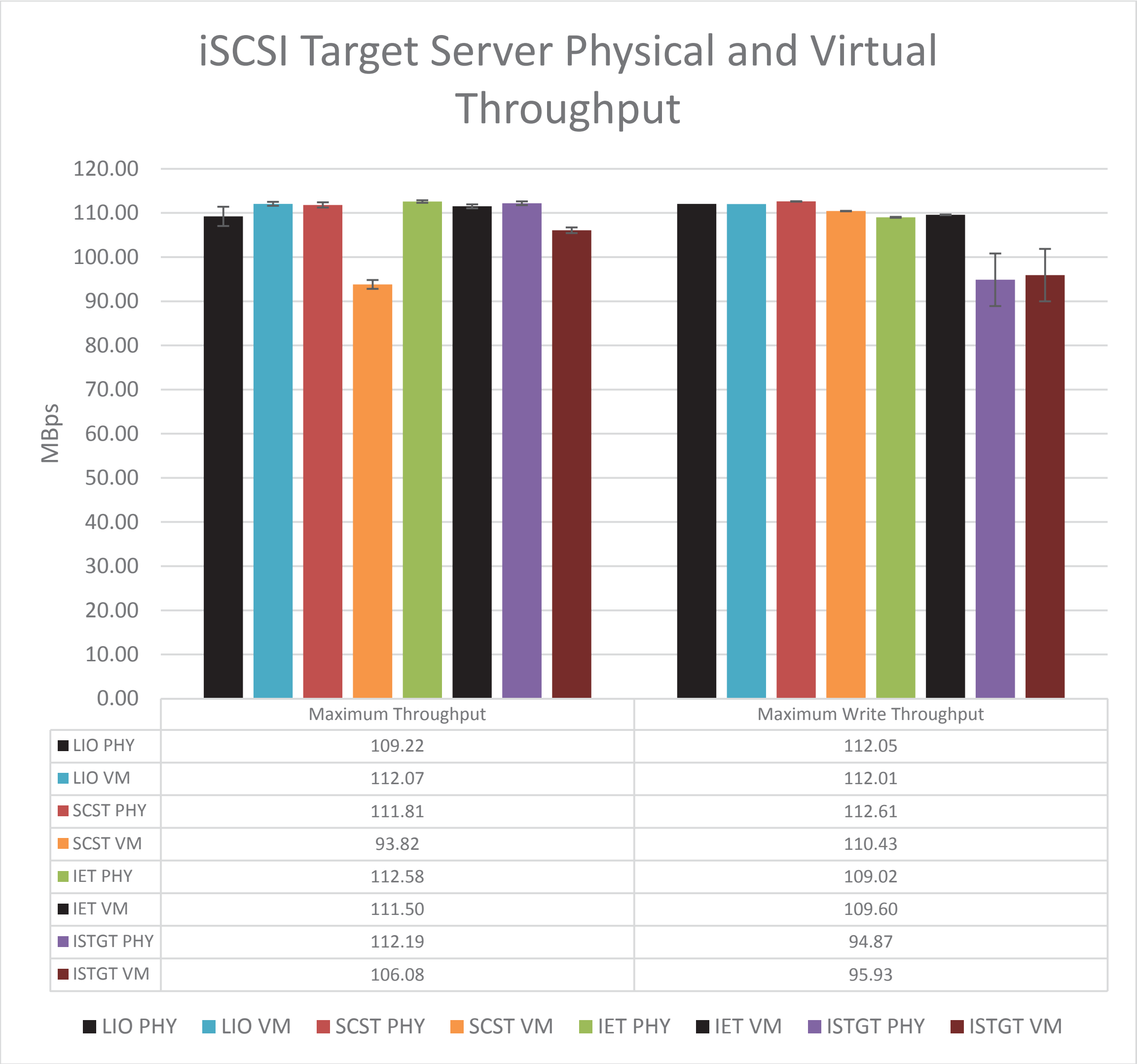
Comparison	Actual Difference	Percentage Difference
IET to SCST	2314.12	5%
IET to LIO	23463.18	91%
IET to ISTGT	24748.47	101%

Physical Maximum Write IOPS

Comparison	Actual Difference	Percentage Difference
IET to SCST	10000.16	23%
IET to LIO	20755.65	63%
IET to ISTGT	32398.25	153%

Comparing the overall data between iSCSI targets and both initiators has been shortened to the maximum throughput and IOPS test sets. The maximum read and write throughput is compared as well as the maximum read and write IOPS. This data can be used to infer which iSCSI SAN target is the most consistent and yielded the highest values.

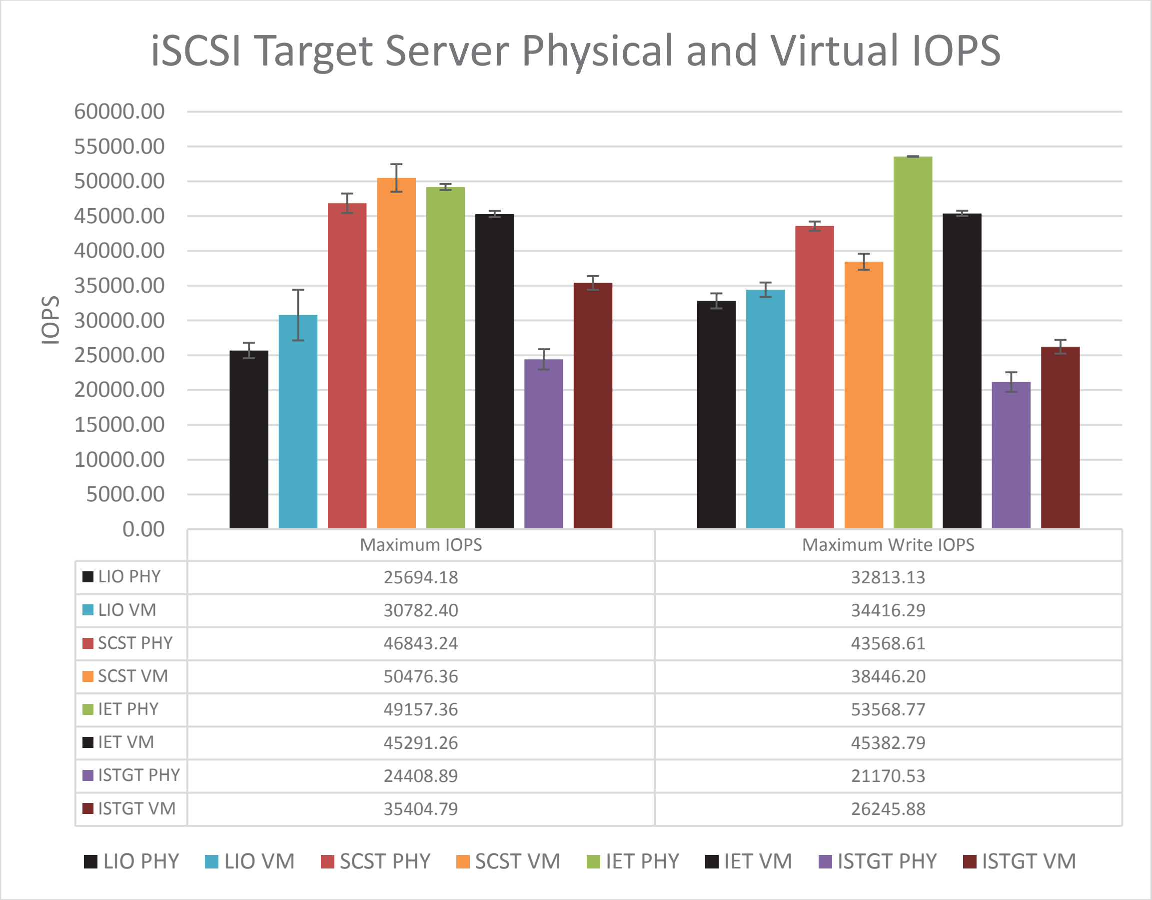
In Figure 4.21 below, physical and virtual iSCSI initiators are shown from all four experiments. The data shows throughput speeds in MB/s for all initiator and target combinations are mostly comparable. All test values are near the maximum line speed of 112 MB/s previously determined with the exception of ISTGT results and SCST virtual read results.



	Maximum Throughput	Maximum Write Throughput
⬛ LIO PHY	109.22	112.05
🟦 LIO VM	112.07	112.01
🟥 SCST PHY	111.81	112.61
🟧 SCST VM	93.82	110.43
🟩 IET PHY	112.58	109.02
⬛ IET VM	111.50	109.60
🟪 ISTGT PHY	112.19	94.87
🟫 ISTGT VM	106.08	95.93

Figure 4.21 iSCSI Target Server Physical and Virtual Throughput

Finally, IOPS were compared with the same test sets of physical, virtual, and SAN architectures. The data points in Figure 4.22 below show a large variation between SAN targets. While previous throughput tests showed most iSCSI initiators as comparable with line speed, the IOPS values are more diverse. IET and SCST had the highest values in both tests with ISTGT being the lowest. LIO, SCST, and ISTGT resulted in higher virtual values as opposed to the respective physical results for maximum read IOPS. The only SAN target yielding higher physical read IOPS results was IET. LIO and ISTGT resulted in higher virtual values as opposed to physical values while SCST and IET both had higher physical values.



	Maximum IOPS	Maximum Write IOPS
LIO PHY	25694.18	32813.13
LIO VM	30782.40	34416.29
SCST PHY	46843.24	43568.61
SCST VM	50476.36	38446.20
IET PHY	49157.36	53568.77
IET VM	45291.26	45382.79
ISTGT PHY	24408.89	21170.53
ISTGT VM	35404.79	26245.88

Figure 4.22 iSCSI Target Server Physical and Virtual IOPS

4.7 Summary

Observing and analyzing the data acquired from the experiments, the architecture with the overall highest throughput values was not defined. Dependent on the SAN target analyzed, physical and virtual architectures produced varying results. In general, no single architecture consistently produced higher values than the others across all four experiments. The entire data set from all experiments conducted is shown in Appendix G. The data collected infers each SAN target handles initiators differently, regardless of architecture.

Likewise, the data collected and analyzed regarding IOPS metrics did not infer a clear pattern to accurately conclude which architecture was most consistent. In five of the eight total IOPS tests, the virtual architecture produced higher IOPS values than the physical architecture. In some IOPS measuring tests, differences in IOPS among architectures ranged from about 1,800 to 30,000 IOPS. Analyzing the associated MB/s accompanied by the respective variances in IOPS values, 20,000 IOPS is between 9-10 MB/s throughput difference.

The MTU size of frames also did not produce uniform results. Dependent on the SAN target tested, the two initiator architectures produced varying throughout and IOPS results. In most cases, jumbo frames produced results which were more sporadic and less consistent. In addition, there were instances where jumbo frames produced lower values than standard frames.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

5.1 iSCSI Target Server Conclusions

While some iSCSI targets resulted in higher throughput or IOPS values for certain tests, an overly successful target was not depicted by the data. The analyzed data and series of experiments, however, provided insight into iSCSI overhead and SAN performance. Similarly to previous discussed experiments conducted by Aiken & Grunwald, iSCSI introduced a noticeable difference between local SCSI traffic due to network bottlenecks.

While using a single iSCSI initiator, the difference between physical and virtual architectures did not produce consistent results. In most cases, the maximum throughput measured only differed by one or two MB/s. The minor variation between the two architectures was unexpected. The implementation of the entire virtual operating system being ran from the iSCSI target was expected to produce more overhead opposed to the physical counterpart. The exception was SCST which had an overall read virtual throughput value approximately 18 MB/s, or 1.9% less than the physical implementation.

Additionally, IOPS results obtained during the experiments were higher in virtual instances as opposed to physical instances, in most cases. LIO and ISTGT consistently resulted in higher IOP values in read and write tests, while SCST only had higher read IOP values. Although a physical to virtual difference was established in IOPS values, the actual difference between architectures in most cases were again minimal. The most noticeable difference between architectures involved ISTGT for maximum read IOPS. The difference between the architectures was nearly 11,000 IOPS, or 45%.

In regards to the research question proposed previously, virtualization did not introduce a measureable amount of overhead compared to the physical initiator. Previous research and testing suggested virtualization would introduce a measureable amount of overhead in regards to iSCSI throughput. While some virtualization throughput values exceeded their physical counterparts, the results were largely dependent and relational to the type of SAN tested.

The second research question referenced the effect jumbo frames had on a given architecture. Altering the MTU size did not produce a measurable difference in most cases. This is contrary to previous experiments discussed in the literature review. In experiments from Simitci evaluating iSCSI performance, jumbo frames increased throughput values by 60% (Simitci, Malakapalli, & Gunturu, 2001). During the set of experiments in this thesis with modern iSCSI targets and hardware, jumbo frames only marginally increased some application specific tests. Additionally the overall throughput decreased in some instances. Implementing jumbo frames introduces additional overhead when configuring network equipment and hardware. As previously stated, jumbo frames are not an IEEE standard so different hardware manufacturers can have different implementations. While MTU size generally did not introduce large throughput differences, if any, some iSCSI targets such as ISTGT slightly benefited from jumbo frames during application specific tests.

The trend of data infers older iSCSI targets, such as IET and SCST, produced higher throughput values. The more modern iSCSI targets, LIO and ISTGT, produced comparable values in throughput tests but lower values in IOPS tests.. While implementing the most practical iSCSI target server for an environment, it is necessary to observe other factors besides throughput metrics.

As discussed previously, the newer iSCSI target servers such as LIO have advanced error handling. Although not tested, SANs generally have multiple initiators connected at a single time. These multiple initiators increase the chance for iSCSI transport errors. LIO is stated to have the most advanced error handling logic which may provide higher throughput rates during errors among other targets.

ISTGT also has advanced error handling and hardware acceleration support for virtual environments. Out of the 4 targets tested, ISTGT was the only target which allowed for hardware acceleration. Hardware acceleration uses the SAN hardware to decrease the time certain virtualization tasks such as cloning takes. This relieves network traffic and decreases the amount of time needed for virtualization tasks.

All iSCSI targets tested, besides ISTGT, were created from a standard installation of Fedora 18. These basic implementations of iSCSI targets were extremely limited in support and documentation. These targets are often built into software bundles such as OpenFiler (IET) and OpenE (SCST). These products have a support community which can provide troubleshooting assistance as well as paid support. The bundled SAN solutions also provide easier installation and implementation with graphical or text based installers.

Datacenters implementing open source SAN solutions have a variety of metrics to take into account. In most instances, modern SAN targets suggest jumbo frames introduce more overhead with minimal gain. Additionally, a virtual environment alone does not introduce a significant amount of overhead. From the acquired data, implementation best practices are dependent on the primary use case intended for a SAN.

The results of this research add to the body of knowledge of open source SAN implementations. As previously discussed, open source SAN documentation and installation methods can be scarce. The results contained in this research can assist others who seek to implement one of the tested SAN targets. Also, the metrics tested such as architecture and MTU size can be applied to SAN targets other than IET, SCST, LIO and ISTGT. Additionally, the research has reinforced the fact that virtualization introduces minimal overhead compared to physical machines and in some instances can outperform physical machines. Tests utilizing modern SAN targets and computer hardware have determined altering the MTU size may be beneficial in certain application specific implementations but generally create more overhead and varied results. Using this research as a guide, open source SAN implementations can be better planned for optimal performance. Finally, this research introduces additional areas for others to expand upon and research further.

5.2 Future Work

Throughout the course of these experiments, a large amount of data was obtained and analyzed. By analyzing the data, other areas for future work became apparent. Criteria for future work is defined in this sense as additional relevant areas which have the possibility of expanding on the previous research. The additional areas discussed in this section include multiple iSCSI initiators, additional virtualization tasks, failover times, and increased hardware specifications.

The first area for future work is the idea of using multiple initiators in a single test as opposed to only one single initiator. While the tests in this research focused around maximum obtainable speeds, SANs normally have multiple initiators passing data at the same time. Multiple iSCSI initiators passing traffic could introduce additional overhead which targets could handle differently. Additional research with up to three initiators was done after the primary experiments previously discussed and insinuated a slight decrease in overall throughput. Providing additional stress on iSCSI targets may impact overall performance.

The next area for future work involves focusing solely on additional virtualization aspects. As discussed previously in this thesis, virtualization has often been associated with and complimented by SANs. VMware and other virtualization products introduce additional aspects which could affect overall throughput and IOPS. Such aspects include cloning virtual machines and virtual machine suspension times. Performing similar throughput tests or timed tests regarding these factors are additional areas of concern.

The final area of additional research revolves around different equipment which would not limit throughput rates or introduce bottlenecks. The data collected in the previous experiments was limited to the throughput speed of the network. Additional research completed which utilized 10 GbE network fabric or MPIO could be used to determine if iSCSI targets have more variance in maximum throughput values. Similarly to the OpenFiler tests achieving 100,000 random IOPS performed by Intel, utilizing available equipment which was only limited by the actual protocol could provide more insight into iSCSI target differences (Intel, 2012).

REFERENCES

REFERENCES

Ahmad, I., Anderson, J. M., Holler, A. M., Kambo, R., & Makhija, V. (2003). An analysis of disk performance in VMware ESX server virtual machines. *2003 IEEE International Conference on Communications (Cat. No.03CH37441)*, 65–76. doi:10.1109/WWC.2003.1249058

Aiken, S., & Grunwald, D. (2003). A performance analysis of the iSCSI protocol. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1194849

An iSCSI target implementation for multipath failover cluster nodes. (n.d.). Retrieved from http://www.peach.ne.jp/archives/istgt/

Csaplar, D. (2012). Is the hypervisor market expanding or contracting? Retrieved from http://blogs.aberdeen.com/it-infrastructure/is-the-hypervisor-market-expanding-or-contracting/

Drummonds, S. (2008). VMware communities: Storage system performance analysis with Iometer. Retrieved April 14, 2013, from http://communities.vmware.com/docs/DOC-3961

Faustini, A., Solder, C., Scheibe, T., Law, D., Ayandeh, S., Kohl, B., & Multanen, E. (2009). Ethernet Jumbo Frames, 1–10.

Generic SCSI Target Subsystem for Linux. (n.d.). Retrieved April 04, 2013, from http://scst.sourceforge.net/target_iscsi.html

Intel. (2012). *Meeting the demands of virtual storage in the cloud* (p. 3). Retrieved from https://download.openfiler.com/Intel_FrankenSAN_Case_Study_3-16-12.pdf

Iometer User's Guide. (2003). Retrieved from http://iometer.cvs.sourceforge.net/viewvc/iometer/iometer/Docs/Iometer.pdf

Kolahi, S. S., Narayan, S., Nguyen, D. D. T., & Sunarto, Y. (2011). Performance monitoring of various network traffic generators. *2011 UkSim 13th International Conference on Computer Modelling and Simulation*, 501–506. doi:10.1109/UKSIM.2011.102

Linus Torvalds. (2011). No Title. Retrieved from http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=commitdiff;h=38567333a6dabd0f2b4150e9fb6dd8e3ba2985e5

Liu, E., & Ai-shaikh, R. (2009). SAN performance evaluation testbed (pp. 1–5).

Lowe, Sc. (2010). Calculate IOPS in a storage array. *Tech Republic*. Retrieved February 26, 2013, from http://www.techrepublic.com/blog/datacenter/calculate-iops-in-a-storage-array/2182

Ren, C., Wang, D., Urgaonkar, B., & Sivasubramaniam, A. (2012). Carbon-Aware energy capacity planning for datacenters, 391–400. Retrieved from http://ieeexplore.ieee.org.ezproxy.lib.purdue.edu/xpl/articleDetails.jsp?arnumber=6298199

Rodrigues, G. (2011). A tale of two SCSI targets. Retrieved from http://lwn.net/Articles/424004/

Rosenblum, M., & Waldspurger, C. (2011). Virtualization I/O: Decoupling a logical device from its physical implementation offers many compelling advantages., *9*(11), 30. doi:10.1145/2063166.2071256

Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., & Zeidner, E. (2004). Internet Small Computer Systems Interface (iSCSI) - RFC 3720.

Simitci, H., Malakapalli, C., & Gunturu, V. (2001). Evaluation of SCSI over TCP/IP and SCSI over fibre channel connections, *55344*, 87–91. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=946698&tag=1

The iSCSI Enterprise Target Project. (n.d.). Retrieved April 04, 2013, from http://iscsitarget.sourceforge.net/

The Linux SCSI target wiki. (2013). Retrieved June 04, 2013, from http://linux-iscsi.org/wiki/Target

Torvalds, L. (n.d.). No Title. Retrieved from http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=38567333a6dabd0f2b4150e9fb6dd8e3ba2985e5

VMware. (2009). Recommendations for aligning VMFS partitions. *2009—01—01)[2o12-04—01].*, 1–10. Retrieved from http://www.vmware.com/pdf/esx3_partition_align.pdf

VMware. (2012). Performance best practices for VMware vSphere 5.1. *2010 —O1—01)[2012—04—01].* Retrieved from http://www.vmware.com/pdf/Perf_Best_Practices_vSphere5.0.pdf

VMware. (2013). VMware I/O Analyzer Installation and User's Guide, 1–34. Retrieved from http://download3.vmware.com/software/vmw-tools/ioanalyzer/IOAnalyzerGuide_1.5.1_20130412.pdf

Yoder, A. G., Carlson, M., Thiel, D., Deel, D., & Hibbard, E. (2012). The 2012 SNIA dictionary, *v.2012.1.E*.

Zhang, J., Yang, B., Guo, J., & Jia, H. (2005). Study and performance analysis of IP-based storage systems. (D. Xu, K. A. Schouhamer Immink, & K. Shono, Eds.), *5643*, 150–159. doi:10.1117/12.571020

APPENDICES

Appendix A iSCSI Network Topology

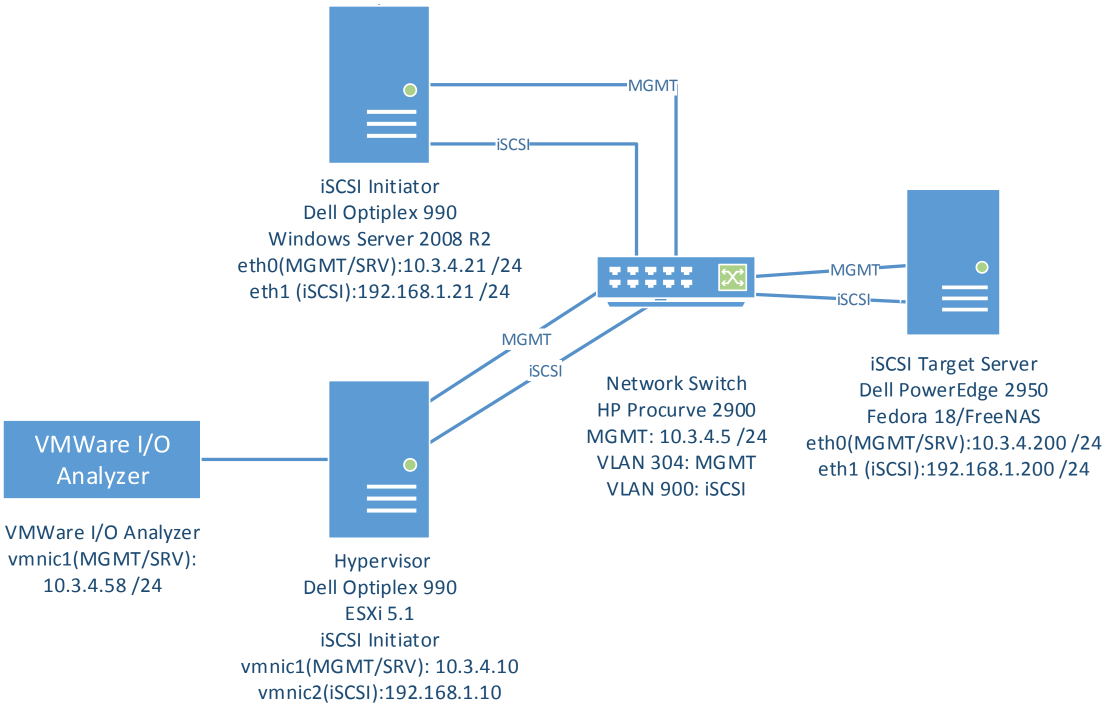


Figure A.1 IP Network Topology iSCSI Architecture

Appendix B IET SAN Configuration File

```
[root@IETsan ~]# cat /etc/tgt/targets.conf
# This is a sample config file for tgt-admin.
# The "#" symbol disables the processing of a line.
# Set the driver. If not specified, defaults to "iscsi".
default-driver iscsi
# Set iSNS parameters, if needed
#iSNSServerIP 192.168.111.222
#iSNSServerPort 3205
#iSNSAccessControl On
#iSNS On

# Continue if tgtadm exits with non-zero code (equivalent of
# --ignore-errors command line option)
#ignore-errors yes

<target iqn.2013-01.test.lcl:iet-phy>
   # provided devicce as a iSCSI target
   backing-store /dev/vg_target00/lv_target00
   # iSCSI Initiator's IP address you allow to connect
   initiator-address 192.168.1.0/24
```

```
</target>

<target iqn.2013-03.test.lcl:iet-vm>
    # provided devicce as a iSCSI target
    backing-store /dev/vg_target03/lv_target03
    # iSCSI Initiator's IP address you allow to connect
    initiator-address 192.168.1.0/24

</target>
```

Appendix C SCST SAN Configuration File

```
# /etc/scst.conf Configuration File

HANDLER vdisk_fileio {

        DEVICE disk01 {

                filename /dev/sda1

                nv_cache 1

        }

        DEVICE disk02 {

                filename /dev/sda2

                nv_cache 1

        }

}


TARGET_DRIVER iscsi {

        enabled 1


        TARGET iqn.2013-01.test.lcl:scst-vm {

                LUN 1 disk01

                enabled 1

        }


        TARGET iqn.2013-02.test.lcl:scst-phy {
```

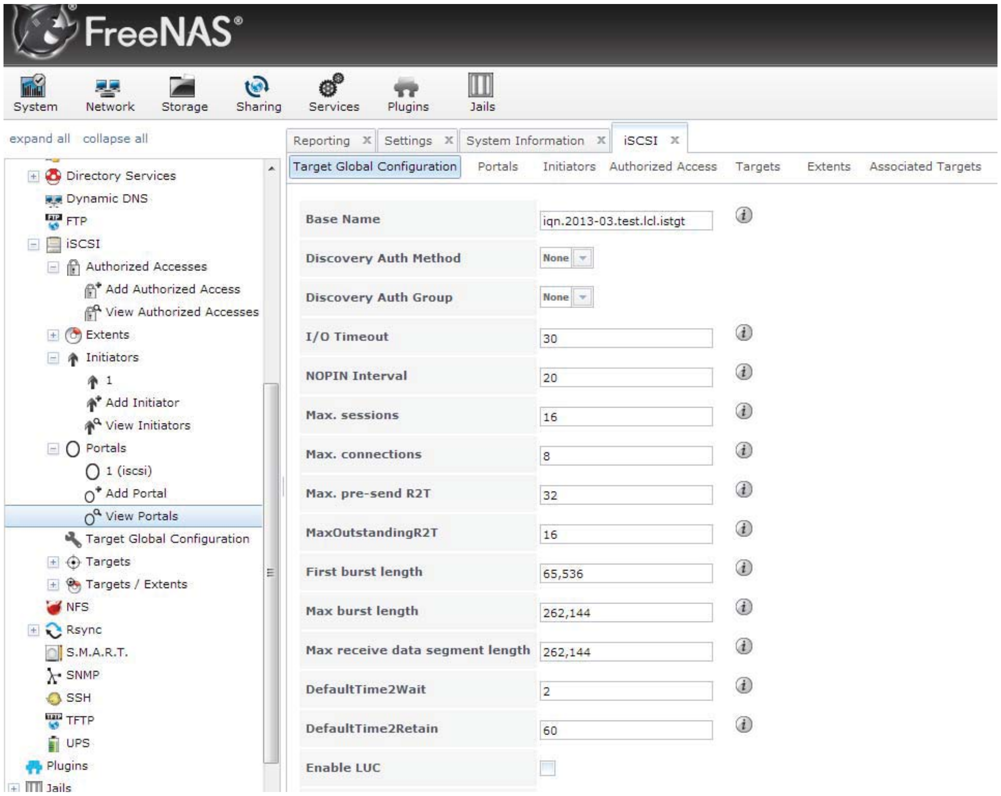
```
                LUN 2 disk02

                enabled 1

        }



}
```

Appendix D LIO SAN Configuration

```
[root@LioSAN]# targetcli

targetcli shell version 2.1.26

Copyright 2011 by RisingTide Systems LLC and others.

For help on commands, type 'help'.

/> ls

o- / ......................................................................... [...]

  o- backstores .............................................................. [...]

  | o- block .................................................. [Storage Objects: 2]

  | | o- lio1 ....................................... [/dev/sda1 (200.0GiB) write-thru activated]

  | | o- lio2 ....................................... [/dev/sda2 (200.0GiB) write-thru activated]

  | o- fileio ................................................. [Storage Objects: 0]

  | o- pscsi .................................................. [Storage Objects: 0]

  | o- ramdisk ................................................ [Storage Objects: 0]

  o- iscsi ............................................................ [Targets: 2]

  | o- iqn.2013-01.test.lcl:Lio2950-vm ........................................ [TPGs: 1]

  | | o- tpg1 ........................................................... [enabled]

  | |   o- acls ........................................................... [ACLs: 1]

  | |   | o- iqn.1998-01.com.vmware:esxi-592ddc91 ............................. [Mapped LUNs: 1]

  | |   |   o- mapped_lun0 ............................................ [lun0 block/lio1 (rw)]

  | |   o- luns ........................................................... [LUNs: 1]

  | |   | o- lun0 ..................................................... [block/lio1 (/dev/sda1)]

  | |   o- portals ....................................................... [Portals: 1]
```

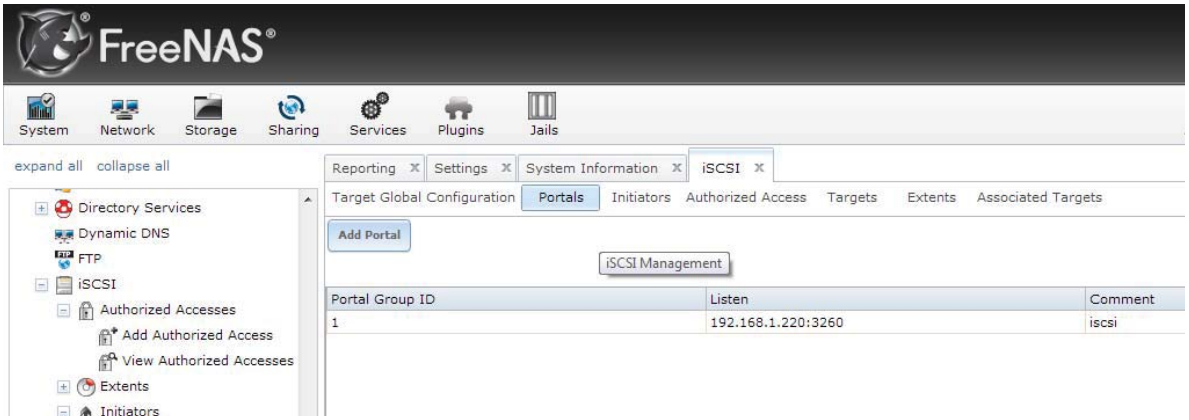
```
| |     o- 192.168.1.220:3260 ............................................................. [OK]
| o- iqn.2013-02.test.lcl:lio2950-phy ................................................ [TPGs: 1]
|   o- tpg1 .......................................................................... [enabled]
|     o- acls ........................................................................ [ACLs: 1]
|     | o- iqn.1991-05.com.microsoft:win-v32tlnb77vj .......................... [Mapped LUNs:
1]
|     |   o- mapped_lun0 ................................................ [lun0 block/lio2 (rw)]
|     o- luns ........................................................................ [LUNs: 1]
|     | o- lun0 ...................................................... [block/lio2 (/dev/sda2)]
|     o- portals .................................................................. [Portals: 1]
|       o- 192.168.1.220:3260 ............................................................. [OK]
o- loopback ....................................................................... [Targets: 0]
o- vhost ............................................................................ [Targets: 0]
```

Appendix E ISTGT SAN Configuration



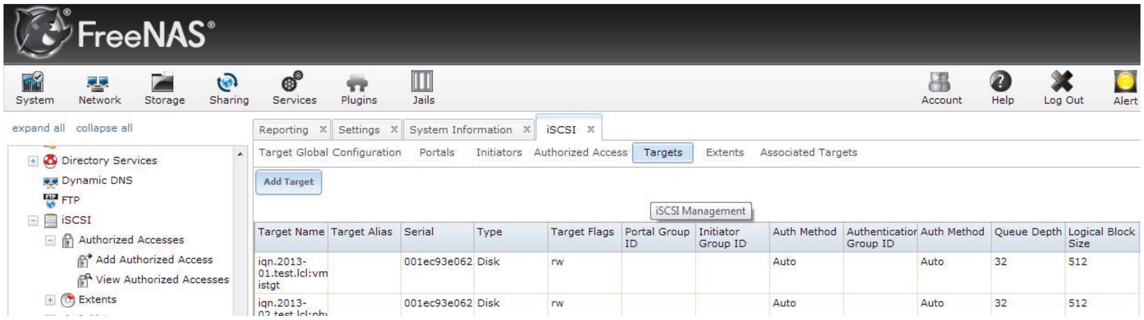
Base Name	iqn.2013-03.test.lcl.istgt
Discovery Auth Method	None
Discovery Auth Group	None
I/O Timeout	30
NOPIN Interval	20
Max. sessions	16
Max. connections	8
Max. pre-send R2T	32
MaxOutstandingR2T	16
First burst length	65,536
Max burst length	262,144
Max receive data segment length	262,144
DefaultTime2Wait	2
DefaultTime2Retain	60
Enable LUC	☐

Figure E.1 FreeNAS Target Global Configuration



Portal Group ID	Listen	Comment
1	192.168.1.220:3260	iscsi

Figure E.2 FreeNAS Portals Configuration

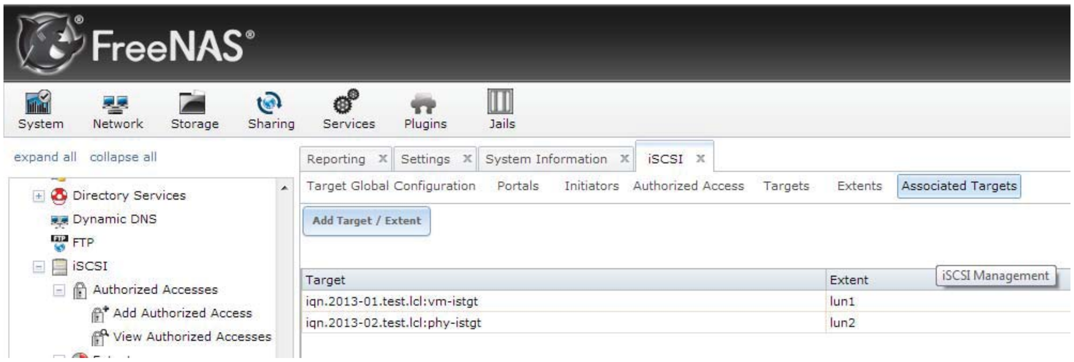


Target Name	Target Alias	Serial	Type	Target Flags	Portal Group ID	Initiator Group ID	Auth Method	Authenticatior Group ID	Auth Method	Queue Depth	Logical Block Size
iqn.2013-01.test.lcl:vmistgt		001ec93e062	Disk	rw			Auto		Auto	32	512
iqn.2013-02.test.lcl:phy		001ec93e062	Disk	rw			Auto		Auto	32	512

Figure E.3 FreeNAS Targets Configuration

Extent Name	Extent Type	Path to the extent	Comment
lun1	ZVOL	/dev/zvol/iSCSI/vm	vm
lun2	ZVOL	/dev/zvol/iSCSI/phy	phy

Figure E.4 FreeNAS Extents Configuration



Target	Extent
iqn.2013-01.test.lcl:vm-istgt	lun1
iqn.2013-02.test.lcl:phy-istgt	lun2

Figure E.5 FreeNAS Associated Targets

Appendix F HP Procurve 2950 Configuration

```
ProCurve Switch 2900-24G# show run
Running configuration:
; J9049A Configuration Editor; Created on release #T.13.71
hostname "ProCurve Switch 2900-24G"
module 1 type J86xxA
module 3 type J90XXA
ip default-gateway 10.3.4.1
vlan 1
   name "DEFAULT_VLAN"
   untagged A1-A4
   ip address dhcp-bootp
   tagged 24
   no untagged 1-23
   exit
vlan 304
   name "Management"
   untagged 1-12
   ip address 10.3.4.5 255.255.255.0
   exit
vlan 900
   name "iSCSI"
```

```
  untagged 13-24

  no ip address

  jumbo

  exit

jumbo max-frame-size 9018

ip route 10.0.0.0 255.0.0.0 10.3.4.1

snmp-server community "public" Unrestricted
```

Appendix G Raw iSCSI Average Values

Table G.1 IET Virtual IOPS With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	44730.55	45297.46	45845.78	45291.26	455.31
Maximum Write IOPS	45240.57	45906.26	45001.55	45382.79	382.80

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	42216.12	42897.59	42662.66	42592.12	282.64
Maximum Write IOPS	45128.85	44255.27	43014.98	44133.03	867.30

Table G.2 IET Physical IOPS With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	49754.35	48725.28	48992.45	49157.36	436.00
Maximum Write IOPS	53516.81	53623.37	53566.15	53568.77	43.54

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	39483.66	38100.29	41984.56	39856.17	1607.47
Maximum Write IOPS	51462.59	51298.60	51625.59	51462.26	133.49

Table G.3 IET Virtual MBps With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	111.30	112.10	111.10	111.50	0.43
Maximum Write Throughput	109.43	109.72	109.65	109.60	0.12
Exchange 03	2.72	1.85	2.46	2.34	0.37
Exchange 07	3.48	4.47	4.19	4.05	0.42
SQL 16K	8.11	8.59	8.24	8.31	0.20
SQL 64K	24.98	24.87	24.00	24.62	0.44
Web Server	2.55	2.41	2.49	2.48	0.06
Workstation	2.49	2.55	2.49	2.51	0.03

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	106.54	106.39	106.84	106.59	0.19
Maximum Write Throughput	112.94	113.78	113.46	113.39	0.35
Exchange 03	2.00	2.48	2.01	2.16	0.22
Exchange 07	4.01	4.10	4.21	4.11	0.08
SQL 16K	8.62	8.65	8.70	8.65	0.03
SQL 64K	24.70	25.98	25.69	25.46	0.55
Web Server	2.59	2.55	2.72	2.62	0.07
Workstation	2.58	2.35	2.40	2.44	0.10

Table G.4 IET Physical MBps With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	112.95	112.25	112.53	112.58	0.29
Maximum Write Throughput	109.13	108.90	109.04	109.02	0.10
Exchange 03	1.94	1.94	2.04	1.97	0.05
Exchange 07	3.59	3.59	3.70	3.62	0.05
SQL 16K	7.52	7.52	7.58	7.54	0.03
SQL 64K	14.67	14.67	14.38	14.57	0.14
Web Server	3.74	3.74	3.74	3.74	0.00
Workstation	3.66	3.67	3.65	3.66	0.01

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	76.12	75.32	75.80	75.75	0.33
Maximum Write Throughput	109.34	109.80	109.99	109.71	0.27
Exchange 03	2.10	2.34	1.57	2.00	0.32
Exchange 07	3.84	3.26	3.55	3.55	0.24
SQL 16K	7.34	6.54	6.99	6.96	0.33
SQL 64K	14.87	14.22	14.62	14.57	0.27
Web Server	3.21	3.91	3.89	3.67	0.33
Workstation	3.79	3.10	2.90	3.26	0.38

Table G.5 SCST Virtual IOPS With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	52305.12	51398.67	47725.29	50476.36	1980.19
Maximum Write IOPS	39547.13	36859.23	38932.23	38446.20	1149.89

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	46737.27	46530.27	48675.23	47314.26	966.06
Maximum Write IOPS	38356.23	38756.28	38956.00	38689.51	249.36

Table G.6 SCST Physical IOPS With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	45629.19	46098.24	48802.29	46843.24	1398.43
Maximum Write IOPS	44116.23	42648.11	43941.50	43568.61	654.79

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	48434.53	48332.62	48535.85	48434.33	82.97
Maximum Write IOPS	45129.75	45523.66	46087.91	45580.44	393.22

Table G.7 SCST Virtual MBps With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	93.48	92.81	94.83	93.82	1.01
Maximum Write Throughput	110.97	110.23	110.09	110.43	0.39
Exchange 03	1.45	1.53	2.56	1.85	0.51
Exchange 07	2.76	2.26	2.98	2.67	0.30
SQL 16K	3.87	3.90	3.87	3.88	0.01
SQL 64K	8.53	9.86	9.12	9.17	0.54
Web Server	2.11	1.56	1.69	1.79	0.23
Workstation	2.54	1.55	1.23	1.77	0.56

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	87.98	88.76	88.95	88.56	0.42
Maximum Write Throughput	114.21	115.23	113.01	114.15	0.91
Exchange 03	1.45	1.95	1.91	1.77	0.23
Exchange 07	2.87	3.56	2.20	2.88	0.56
SQL 16K	4.32	4.20	3.99	4.17	0.14
SQL 64K	9.45	11.63	9.10	10.06	1.12
Web Server	1.42	1.76	2.54	1.91	0.47
Workstation	1.62	2.87	1.12	1.87	0.74

Table G.8 SCST Physical MBps With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	112.57	111.17	111.70	111.81	0.58
Maximum Write Throughput	111.87	112.98	112.98	112.61	0.52
Exchange 03	1.39	1.53	1.53	1.48	0.06
Exchange 07	2.05	2.19	2.20	2.15	0.07
SQL 16K	3.15	3.41	3.25	3.27	0.11
SQL 64K	7.97	8.10	7.58	7.88	0.22
Web Server	2.11	2.11	2.10	2.11	0.01
Workstation	2.10	2.10	2.29	2.17	0.09

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	84.10	85.12	86.14	85.12	0.83
Maximum Write Throughput	74.57	74.96	74.55	74.69	0.19
Exchange 03	1.41	1.52	1.51	1.48	0.05
Exchange 07	2.04	2.20	2.18	2.14	0.07
SQL 16K	3.15	3.34	3.26	3.25	0.08
SQL 64K	7.45	7.56	8.15	7.72	0.31
Web Server	2.10	2.05	2.01	2.05	0.04
Workstation	2.08	2.08	2.16	2.11	0.04

Table G.9 LIO Virtual IOPS With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	28778.93	35892.28	27675.99	30782.40	3641.18
Maximum Write IOPS	34956.78	32957.98	35334.10	34416.29	1042.62

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	28146.28	34148.96	28539.09	30278.11	2741.80
Maximum Write IOPS	33085.49	30156.56	34170.32	32470.79	1695.28

Table G.10 LIO Physical IOPS With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	26932.55	24225.29	25924.69	25694.18	1117.19
Maximum Write IOPS	31695.28	34291.82	32452.28	32813.13	1090.31

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	25483.24	23956.29	24430.11	24623.21	638.15
Maximum Write IOPS	34111.79	34176.56	34180.92	34156.42	31.61

Table G.11 LIO Virtual MBps With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	111.46	112.45	112.30	112.07	0.44
Maximum Write Throughput	112.87	111.09	112.08	112.01	0.73
Exchange 03	1.54	1.29	1.76	1.53	0.19
Exchange 07	2.45	3.54	2.99	2.99	0.44
SQL 16K	5.44	6.34	5.64	5.81	0.39
SQL 64K	16.23	15.21	15.01	15.48	0.53
Web Server	2.08	1.98	2.09	2.05	0.05
Workstation	1.98	1.61	1.98	1.86	0.17

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	118.42	117.10	117.02	117.51	0.64
Maximum Write Throughput	115.18	114.26	114.10	114.51	0.47
Exchange 03	1.03	1.16	1.05	1.08	0.06
Exchange 07	2.91	2.98	2.92	2.94	0.03
SQL 16K	4.56	3.57	2.98	3.70	0.65
SQL 64K	9.87	9.79	9.97	9.88	0.07
Web Server	1.99	1.91	1.95	1.95	0.03
Workstation	1.76	1.80	1.98	1.85	0.10

Table G.12 LIO Physical MBps With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	106.65	109.01	112.01	109.22	2.19
Maximum Write Throughput	112.23	111.90	112.01	112.05	0.14
Exchange 03	1.55	1.97	1.13	1.55	0.34
Exchange 07	2.99	3.04	3.02	3.02	0.02
SQL 16K	5.01	5.69	5.10	5.27	0.30
SQL 64K	13.60	14.04	13.50	13.71	0.23
Web Server	3.29	3.08	2.89	3.08	0.16
Workstation	2.92	2.89	2.81	2.87	0.04

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	94.70	95.80	96.02	95.51	0.58
Maximum Write Throughput	97.65	108.66	99.34	101.88	4.84
Exchange 03	1.53	1.56	1.55	1.55	0.01
Exchange 07	2.84	3.21	3.03	3.03	0.15
SQL 16K	4.12	4.53	4.01	4.22	0.22
SQL 64K	13.64	13.37	13.91	13.64	0.22
Web Server	2.65	3.21	2.67	2.84	0.26
Workstation	2.67	3.42	2.21	2.77	0.50

Table G. 13 ISTGT Virtual IOPS With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	34529.12	34926.24	36759.00	35404.79	971.20
Maximum Write IOPS	25748.97	25359.67	27629.00	26245.88	990.84

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	19122.18	21387.58	20010.23	20173.33	932.01
Maximum Write IOPS	10748.28	9362.38	8472.23	9527.63	936.51

Table G.14 ISTGT Physical IOPS With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	23498.53	26456.27	23271.86	24408.89	1450.68
Maximum Write IOPS	20192.23	23139.23	20180.12	21170.53	1392.09

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum IOPS	24708.46	24623.08	22495.00	23942.18	1023.91
Maximum Write IOPS	21133.86	21404.05	21305.81	21281.24	111.66

Table G.15 ISTGT Virtual MBps With Standardard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	106.75	106.30	105.19	106.08	0.66
Maximum Write Throughput	95.68	96.53	95.57	95.93	0.43
Exchange 03	0.56	0.98	0.49	0.68	0.22
Exchange 07	1.01	1.23	1.62	1.29	0.25
SQL 16K	1.65	1.89	1.41	1.65	0.20
SQL 64K	9.85	10.10	10.98	10.31	0.48
Web Server	10.01	1.56	0.97	4.18	4.13
Workstation	1.12	0.86	1.49	1.16	0.26

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	64.65	71.06	79.87	71.86	6.24
Maximum Write Throughput	53.98	63.48	61.32	59.59	4.07
Exchange 03	0.32	0.76	0.23	0.44	0.23
Exchange 07	1.15	0.87	0.74	0.92	0.17
SQL 16K	1.34	1.87	1.54	1.58	0.22
SQL 64K	3.45	5.12	4.10	4.22	0.69
Web Server	1.04	1.12	1.01	1.06	0.05
Workstation	1.07	0.98	1.08	1.04	0.05

Table G.16 ISTGT Physical MBps With Standard and Jumbo Frames

Standard Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	112.75	112.12	111.70	112.19	0.43
Maximum Write Throughput	93.98	93.66	96.98	94.87	1.49
Exchange 03	3.31	3.26	3.10	3.22	0.09
Exchange 07	6.09	5.43	5.68	5.73	0.27
SQL 16K	7.90	11.76	10.53	10.06	1.61
SQL 64K	28.56	29.87	27.12	28.52	1.12
Web Server	75.47	81.26	65.02	73.92	6.72
Workstation	5.96	5.59	7.91	6.49	1.02

Jumbo Frames	Test 1	Test 2	Test 3	Average	STD
Maximum Throughput	107.56	105.20	106.19	106.32	0.97
Maximum Write Throughput	94.90	95.18	92.09	94.06	1.39
Exchange 03	0.91	0.80	1.86	1.19	0.48
Exchange 07	1.12	2.87	2.79	2.26	0.81
SQL 16K	2.34	3.48	4.70	3.50	0.97
SQL 64K	9.65	10.29	9.10	9.68	0.49
Web Server	53.79	67.89	79.21	66.96	10.40
Workstation	2.30	1.96	3.25	2.50	0.55