



Theses and Dissertations

2014-07-01

Analysis of Data Center Network Convergence Technologies

Robert-Lee Daniel LeBlanc
Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Construction Engineering and Management Commons](#)

BYU ScholarsArchive Citation

LeBlanc, Robert-Lee Daniel, "Analysis of Data Center Network Convergence Technologies" (2014). *Theses and Dissertations*. 4150.

<https://scholarsarchive.byu.edu/etd/4150>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Analysis of Data Center Network Convergence Technologies

Robert-Lee Daniel LeBlanc

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Joseph J. Ekstrom, Chair
C. Richard G. Helps
Dale C. Rowe

School of Technology
Brigham Young University

July 2014

Copyright © 2014 Robert-Lee Daniel LeBlanc

All Rights Reserved

ABSTRACT

Analysis of Data Center Network Convergence Technologies

Robert-Lee Daniel LeBlanc
School of Technology, BYU
Master of Science

The networks in traditional data centers have remained unchanged for decades and have grown large, complex and costly. Many data centers have a general purpose Ethernet network and one or more additional specialized networks for storage or high performance low latency applications. Network convergence promises to lower the cost and complexity of the data center network by virtualizing the different networks onto a single wire. There is little evidence, aside from vendors' claims, that validate network convergence actually achieves these goals. This work defines a framework for creating a series of unbiased tests to validate converged technologies and compare them to traditional configurations. A case study involving two different network converged technologies was developed to validate the defined methodology and framework. The study also shows that these two technologies do indeed perform similarly to non-virtualized network, reduce costs, cabling, power consumption and are easy to operate.

Keywords: data center, networks, ethernet, fibre channel, infiniband, fibre channel over ethernet, FCoE, virtualization, converged fabric, Cisco UCS, Xsigo, Oracle OVN

ACKNOWLEDGMENTS

My wife has been my strongest supporter and always encourages me to be better than I am. It is because of her dedication to our family that I could even possibly have any chance of completing this work. It is the faith she has in me and the love that I have for her that keeps me going when things get tough. I appreciate my three little ones at home who are my cheerleaders and think so highly of me even though I do not deserve it. They are so happy to see me when I've worked long hours and it warms my heart and keeps me grounded and focused on the important things in life. I thank my in-laws for their acts of service so that I could work extra hours to get this study and thesis completed.

Thank you to Paul Nez Jr. who personally provided the hand drawings for my thesis and the 2011 IT Capstone group (Stefano Gessati, Aaron Kimbler, Chase Nebeker, Francisco Parra and Jordan Sheen) who built and executed the tests in the study. I also thank BYU and those who had faith in me and allowed me to implement this vision in a real world production environment. I also thank Dr. Ekstrom for guiding me and encouraging me to do this work and having faith in me. Without him, I would have been lost and this thesis would not have turned out nearly as good. I wish to thank those on my committee who provided useful feedback to make this thesis beneficial for me and others.

TABLE OF CONTENTS

LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
1 Introduction.....	1
1.1 Background.....	1
1.2 Problem Statement.....	2
1.3 Hypothesis.....	2
1.4 Justification.....	2
1.5 Assumptions.....	3
1.6 Delimitations.....	3
2 Review of the Literature.....	4
2.1 Introduction.....	4
2.2 Understanding Virtualization Concepts.....	5
2.2.1 Server or Machine Virtualization.....	6
2.2.2 Storage Virtualization.....	9
2.2.3 I/O Virtualization.....	14
2.3 History of Data Center Network Architectures.....	15
2.3.1 Separate Physical Networks.....	16
2.3.2 VLAN Tagging.....	17
2.3.3 Storage Area Networks (SAN).....	19
2.3.4 Server Communication Networks.....	20
2.4 I/O Virtualization Techniques.....	21

2.4.1 Fibre Channel Over Ethernet (FCoE).....	23
2.4.2 Infiniband.....	25
2.4.3 PCI-E Virtualization.....	27
2.5 Industry Implementation of Communication Virtualization.....	28
2.5.1 Cisco Unified Computing System (UCS).....	28
2.5.2 Other FCoE Manufacturers.....	31
2.5.3 Oracle Virtual Networking (OVN previously Xsigo).....	32
2.5.4 Mellanox Bridge-X.....	35
2.5.5 NextIO.....	35
2.6 Conclusions.....	37
3 Methodology.....	37
3.1 Impetus for the Study.....	37
3.2 Introduction to Analysis Framework.....	39
3.3 Performance Testing.....	39
3.4 Usability.....	40
3.5 Costs.....	40
4 Case Study Procedures.....	42
4.1 Overall Approach.....	42
4.2 Measuring Performance.....	43
4.2.1 Measuring Ethernet Performance.....	45
4.2.2 Measuring vMotion Performance.....	47
4.2.3 Measuring Storage Performance.....	47

4.3 Measuring Usability.....	48
4.4 Cost Comparison.....	49
4.4.1 CapEx Cost Analysis.....	52
4.4.2 OpEx Cost Analysis.....	53
4.4.3 Other Cost Considerations.....	53
5 Results and Data Analysis.....	54
5.1 Ethernet Performance Analysis.....	54
5.2 vMotion Performance Analysis.....	60
5.3 Storage Performance Analysis.....	63
5.4 Usability Analysis.....	66
5.5 Costs Analysis.....	69
6 Conclusions and Recommendations.....	77
6.1 Conclusions.....	77
6.2 BYU's Experience with Xsigo (OVN).....	79
6.3 Recommendations.....	80
6.4 Future Work.....	81
REFERENCES.....	84
APPENDICES.....	90
Appendix A: Glossary of Terms.....	91
Appendix B: Automation Scripts.....	95
Appendix C: Network Performance Data.....	105
Appendix D: vMotion Performance Data.....	111

Appendix E: Storage Performance Data.....	112
Appendix F: Usability Testing Tasks.....	113
Appendix G: Usability Testing Grade Sheet.....	114
Appendix H: Usability Testing Data.....	115
Appendix I: CapEx Data.....	116
Appendix J: OpEx Data.....	119

LIST OF TABLES

Table 2-1: Ethernet Priority Recommendations.....	18
Table 2-2: Comparison of OVN and Cisco Cabling.....	34
Table 4-1: Virtual Machine Design of Experiments.....	44
Table 4-1: Virtual Machine Design of Experiments (cont.).....	45
Table 4-2: Ethernet Testing Configurations.....	46
Table 4-3: Storage Testing Profiles.....	48
Table 5-1: Testing Label Definition.....	54

LIST OF FIGURES

Figure 2-1: Server and Machine Virtualization.....	7
Figure 2-2: Storage Virtualization with Wide Striping.....	10
Figure 2-3: Storage Virtualization with Thin Provisioning.....	12
Figure 2-4: Storage Virtualization with Auto-Tiering.....	13
Figure 2-5: Storage Virtualization of LUN Between Storage Arrays.....	14
Figure 2-6: Traditional Server Network Configuration.....	17
Figure 2-7: Communication Virtualization Logical View.....	22
Figure 2-8: Cisco UCS Blade Architecture (Cisco 2012).....	30
Figure 2-9: A Comparison of Single and Multi Hop FCoE.....	32
Figure 2-10: OVN Configuration.....	33
Figure 2-11: NextIO Communications Virtualization.....	36
Figure 4-1: Traditional Blade Configuration.....	50
Figure 4-2: 10 Gb Blade Configuration.....	50
Figure 4-3: Cisco UCS Blade Configuration.....	51
Figure 4-4: Xsigo Blade Configuration.....	52
Figure 5-1: TCP Bandwidth Results.....	55
Figure 5-2: TCP Bandwidth Distribution.....	56
Figure 5-3: UDP Bandwidth Results.....	57
Figure 5-4: UDP Bandwidth Distribution.....	57
Figure 5-5: UDP Jitter Results.....	58
Figure 5-6: UDP Jitter Distribution.....	59

Figure 5-7: UDP Datagram Loss Results.....	59
Figure 5-8: VM vMotion Time Results.....	61
Figure 5-9: Total vMotion Time.....	61
Figure 5-10: vMotion Times Distribution.....	62
Figure 5-11: Storage I/O Performance.....	64
Figure 5-12: Storage Transfer Performance.....	64
Figure 5-13: Storage Latency Performance.....	65
Figure 5-14: Task Completion Results.....	66
Figure 5-15: Usability Errors for Given Tasks.....	67
Figure 5-16: User Perception of Administrative Console.....	68
Figure 5-17: Chassis Bandwidth.....	70
Figure 5-18: Blade Bandwidth.....	71
Figure 5-19: Cable Count.....	72
Figure 5-20: Guaranteed Bandwidth to Blade.....	72
Figure 5-21: Percent Bandwidth per Cable.....	73
Figure 5-22: I/O Costs per 16 Blades.....	74
Figure 5-23: I/O Power Consumption per 16 Blades.....	75

1 INTRODUCTION

1.1 Background

Data centers based on the x86 server technology have traditionally been built using many discrete components consisting of servers, switches and storage. It was easy to maintain these systems until the demand for redundancy grew. With the introduction of server virtualization the workloads increased rapidly and so did the challenge of keeping all the components connected.

In large organizations, it takes the coordination of several teams to make even the simplest of changes to the communication path of servers, storage and networking. The procurement, physical placement of equipment, and cabling can incur a great deal of expense in capital equipment. It can also take many man-hours to do the physical work which also places a strain on operational expenses. E-commerce businesses have the potential to lose large amounts of revenue if they experience service disruptions in their data centers for a few minutes. In a world of instant everything, if a business can't adapt quickly, they may lose a large portion of their clientele and the revenue that goes with it.

Server virtualization has helped companies become flexible, but it is only one piece of the puzzle. Even though a VM may be created in seconds, it can still take days or weeks for networks to be created and storage targets to be configured. In order for a company to be completely agile, the entire data center must be transformed into a virtual environment where all

hardware is a commodity and resources can be reallocated quickly, easily and with minimal physical changes.

This is very similar to the configuration of the Brigham Young University (BYU) data center. BYU has incorporated server virtualization into their data center environment but the network and storage components are still difficult to manage compared to virtual servers. Converged Input/Output (I/O) solutions should be able to reduce cabling, equipment, management and power while at the same time make the communications connectivity more flexible, but claims of equivalent or improved performance need to be verified.

1.2 Problem Statement

Vendors publish performance data regarding their converged I/O systems, but the details of how those conclusions are reached are not disclosed. What methodology should be used to develop tests which help compare different aspects of converged I/O systems to each other and traditional configurations?

1.3 Hypothesis

An analysis framework which provides guidelines for tests of several aspects of converged I/O can help an organization develop procedures to determine if converged I/O will increase performance, reduce costs and increase flexibility.

1.4 Justification

There is much information around virtual servers and their benefits in the server area, but even though there are indications that I/O convergence can provide similar benefits to I/O in the

data center, there is limited independent validation of this potential. We provide needed guidance to validate the concepts and benefits of I/O convergence through the development of an analysis framework and a case study of two specific I/O convergence technologies.

1.5 Assumptions

Though the analysis framework is generic, the case study environment is modeled after the BYU data center where server virtualization is deployed. Storage virtualization, although a contributing factor to complete data center virtualization, is not analyzed in this work.

1.6 Delimitations

Data center virtualization has three main components:

- Server virtualization
- I/O virtualization
- Storage virtualization

Although all three areas are of interest, this work will focus on I/O virtualization. The other areas of server and storage virtualization are discussed as needed to provide context for how I/O virtualization fits into the overall picture.

The case study in this work will validate the analysis framework by using it to construct a series of unbiased tests for converged I/O within the BYU data center environment.

2 REVIEW OF THE LITERATURE

2.1 Introduction

Many virtualization technologies have roots in industry which leads to a lag between technology deployment and academic analysis. This seems somewhat counter-intuitive as generally the data center is central to any company and much effort is expended to mitigate the risk of service interruptions as much as possible. Hence deploying new technologies unless they are fully validated through independent experimentation into the data center seems like a risky approach to running a business.

However it is very common for new technologies to be deployed into production data centers with varying amounts of testing in a lab environment and mostly based on the research done by the manufacturer of the technology. Because intellectual property is the manufacturer's edge in the marketplace, very little detail about the operation of the technology is disclosed to the public in order to keep it from their competitors. Much of the information and released white papers of these technologies focus on the cost benefits to the business for deploying the technology and is usually only as a comparison with a competing product.

Server virtualization, because of its long history and number of Open Source projects, has a much more complete academic analysis than I/O and storage virtualization and this has helped

VM technology become more mature in the process. Similar analysis needs to be performed in the area of I/O virtualization and convergence to help it mature as well.

2.2 Understanding Virtualization Concepts

“Virtualization, in computing, refers to the act of creating a virtual (rather than actual) version of something, including but not limited to a virtual computer hardware platform, operating system (OS), storage device, or computer network resources.”(Wikipedia, 2014e)

Implementing virtualization can have many benefits including the following:

- Server or machine virtualization abstracts the OS from the physical hardware. This has the benefit of running one or more OSES on one physical server which increases the utilization of the hardware. This reduces the number of physical servers needed which also reduces the power and cooling requirements of the data center. This compartmentalization provides portability of the OS between different hardware configurations.
- Storage virtualization abstracts the data that the OS sees from the physical disk. This provides the ability to manage the data in different ways without the involvement of the OS. Storage virtualization, for example, can relocate data to another physical disk without participation of the OS.
- I/O virtualization abstracts the data flows paths from the physical network connections. By providing large physical links to each device in the data center and then defining the number, types, and capacity of the data paths in software, all aspects of I/O can be reconfigured without physically moving cables.

2.2.1 Server or Machine Virtualization

Server and machine virtualization is the most mature of the three categories because it has been around in one form or another since the 1960s starting with the IBM CP-40. (Kohlbrenner, Morris, and Morris, 2012; Wikipedia, 2012a) Although the x86 architecture was not designed with machine virtualization in mind, the challenges of virtualizing the x86 architecture were overcome and virtual machine use expanded rapidly in the 2000s. Given the great success of machine virtualization, processor and I/O manufacturers have worked to include hypervisor-specific instructions into the architecture to increase the performance of virtual machines (VMs). (Wikipedia, 2012d)

There are two primary types of hypervisors in existence today that are considered a form of hardware virtualization because they virtualize the complete hardware of a computing system. These are known as hosted and bare-metal hypervisors as depicted in Figure 2-1: Server and Machine Virtualization.

A bare-metal hypervisor is usually a very small operating system in which most of the hardware components are passed to the VMs with very little modification. The hypervisor is responsible for ensuring that the resources of the physical machine are appropriately shared and protected between running VMs. Usually bare-metal hypervisors have a strict hardware compatibility list in which hardware has been tested to meet the requirements of sharing and isolation. VMs running on a bare-metal hypervisor are restricted to running OSes that can natively run on the hardware as the hardware architecture is passed mostly unmodified to the VM. (Barham et al., 2003; Bugnion et al., 1997; Popek and Goldberg, 1974; Rosenblum, 2004)

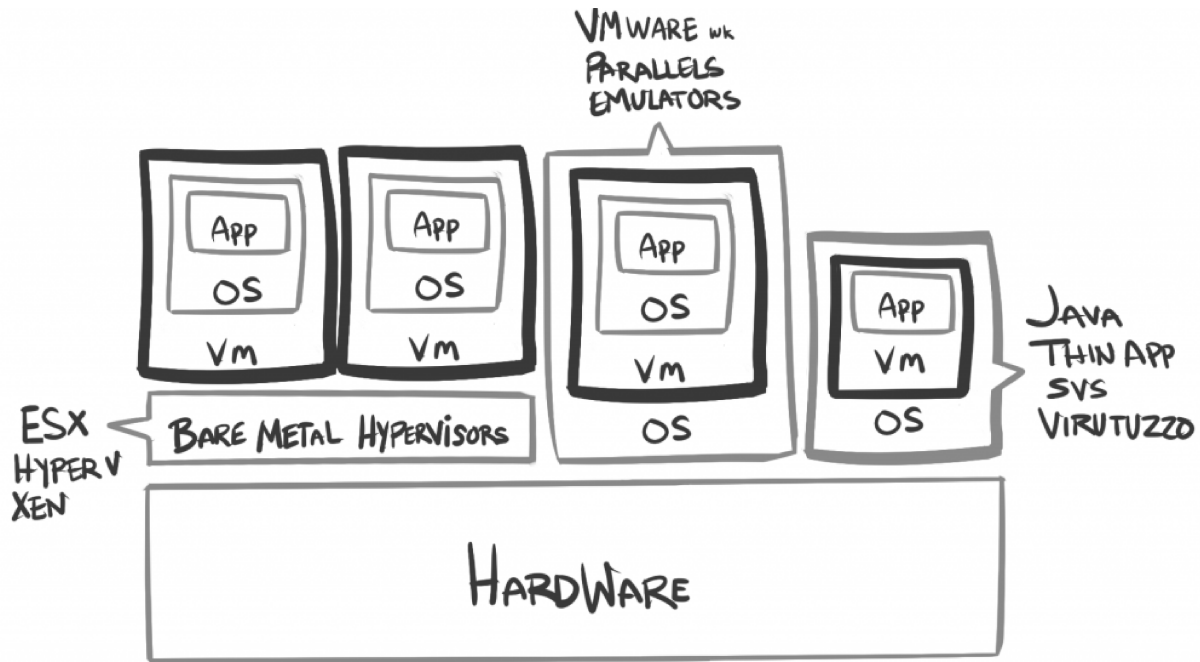


Figure 2-1: Server and Machine Virtualization

A hosted hypervisor runs on top of an existing OS and as such usually has access to any physical hardware that the host OS has access to. In many cases the hypervisor can perform some direct architecture calls like a bare-metal hypervisor, but the extent of what is available is restricted by the host OS. In some hypervisors a full binary translation can occur where a completely different architecture can run inside a VM than that which the host physical machine can support. An example would be running an x86 VM on a PowerPC architecture which was popular in the early 2000s before Apple moved to the x86 processor. (Popek and Goldberg, 1974; Rosenblum, 2004)

There are other virtualization techniques where application runtime environments can be virtualized inside of an OS. Instead of providing a complete VM with virtual hardware, it only creates a virtual application space that the application can run in. This allows for the application

to have greater isolation from other applications and even from the OS on which the application is running. Examples of the application virtualization can be found in Java, Altiris Software Virtualization Solution (SVS), VMware ThinApp and Parallels Virtuozzo. (Killalea, 2008; Parallels, 2012; Rosenblum, 2004) Usually bare-metal and hosted hypervisors are not run on the same hardware at the same time, but it is common to leverage application virtualization with either bare-metal or hosted hypervisors.

Because of the immaturity in the storage and I/O virtualization spaces, hypervisors are trying to do more to provide virtualization benefits in those spaces as well. VMware has been aggressive in this area by providing Storage vMotion and Storage Distributed Resource Scheduling (SDRS) to help move data transparently to the best disk. Storage vMotion is the process which VMware can move the virtual hard disks of a VM to a different storage Logical Unit Number (LUN) while the VM is running. The VM is unaware that its disks are being moved to a new LUN on the same or completely different storage system. This allows administrators to minimize the downtime to migrate to a new storage system or relocate data to a more appropriate tier of disk. SDRS automates the migration of virtual disks based on specific metrics given by the administrator such as LUN free space and LUN latency. VMware has also introduced integration with VXLAN and introduced NSX as a way to virtualize the network to provide more flexibility even across disjointed layer 2 segments. It will be interesting to see how far hypervisors will be able to push into these areas without the help of the underlying hardware. Without the hardware understanding the virtualization layer, the hypervisor will only be able perform to a certain degree, much as was the case with CPU processors and RAM. Storage vendors are already providing Application Programming Interfaces (APIs) for hypervisors to

leverage, which offloads much of the work from Storage vMotion on the storage fabric, hypervisors and storage systems. Offloading the disk operations to the storage unit, which can perform the same operations more efficiently than the server, provides improved performance during these disk intensive operations.

2.2.2 Storage Virtualization

Storage virtualization has been around for some time as well. The most familiar type of storage virtualization is the Redundant Array of Independent Disks (RAID) which abstracts the underlying disk configuration and provides a logical (virtual) disk that the system then utilizes. Because of this virtualization technique if a disk fails in a RAID protected group, the data is still available and can be rebuilt onto a new disk without the system or OS even knowing that a failure and replacement occurred. (Wikipedia, 2014c)

A fairly recent addition to storage virtualization is the logical volume management (LVM) which is found in Linux and Windows Operating Systems. LVM allows one to overcome limitations in disk sizes or partition tables by creating an abstraction that does not have to adhere to such limitations. The logical volume driver creates the logical volume and maps it physically to one or more partitions or disks. This allows the administrator to have greater flexibility in changing the size of the volumes or moving them between physical disks even while they are used. Other features such as snapshots can be provided by logical volumes even if the file system does not natively support such traits. (Wikipedia, 2014c)

A newer idea in storage virtualization is to take this abstraction one step further by uncoupling the logical volume from a RAID group. Modern storage virtualization systems can abstract the logical disk or LUN to exist over multiple RAID groups known as Wide Striping

depicted in Figure 2-2: Storage Virtualization with Wide Striping. (Waldspurger and Rosenblum, 2012) This has an advantage of increasing performance and reducing the possible fault domain of the disk system. (Pinder, 2010)

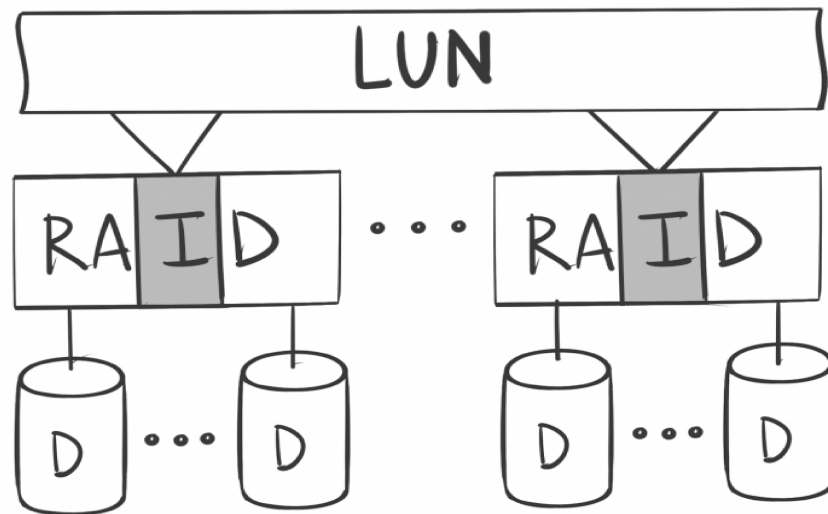


Figure 2-2: Storage Virtualization with Wide Striping

When a disk in a RAID group fails and is replaced with a new drive, the system must read all the data from the remaining disk in the RAID group to rebuild the missing data. It either computes what the real data was or computes the new parity information and writes that to the new drive. During this process none of the data is offline so the storage system is continuing to serve normal storage requests while at the same time reading all the data off the disks to generate the data that should be on the new drive. This process places an intense load on the drives in the RAID group until the new disk has been fully rebuilt. This increased load can cause a drive that is on the brink of failure to finally fail during the rebuild process.

If system has a certain number of drives of the same size, then by grouping a smaller number of disks into a RAID group, if one disk fails then the system has to less data to read in order to rebuild the lost data; this results in a faster rebuild time and minimizes the chances of another disk failing. The longer it takes to rebuild a RAID group, the longer the system is under higher load and is susceptible to an additional disk failure which could mean a total loss of data.

Wide striping can also reduce the occurrence of hot spots in a storage system by spreading the I/O load over as many physical disks as possible. Hot spots in traditional storage systems are caused by heavy workloads which are confined to a RAID group while other RAID groups are left idle. The idle RAID groups are not able to contribute their currently unused performance to help with the workload and so the process suffers with poorer performance even though there is still plenty of capacity in the system. Wide striping helps to utilize all the performance in the storage system by spreading the load to all the RAID groups in the system. (Pinder, 2010)

Another feature that modern storage virtualization systems provide is on-demand allocation of storage or thin provisioning. Thin provisioning only writes the actual data portion of the LUN to disks as shown in Figure 2-3: Storage Virtualization with Thin Provisioning. The storage system presents what appears to be the entire requested space to the server. The storage system does not use any storage on the physical disks until something other than zeros is written to it. The storage system keeps a mapping of blocks presented to the server with their corresponding blocks on physical disks. The blocks from many thin provisioned LUNs can be interleaved as each of the LUNs write to new blocks in storage. Thin provisioning allows a business to buy as much disk is needed at the time of purchase and defer additional purchases

until the disk is actually used. (Pinder, 2010) At BYU with hundreds of terabytes of provisioned storage, almost 60% of it was unused. (LeBlanc, 2011b)

Storage virtualization can also help solve the problem of selecting which tier of disk a workload should reside on for the best cost to performance ratio. A tier of disk can be categorized

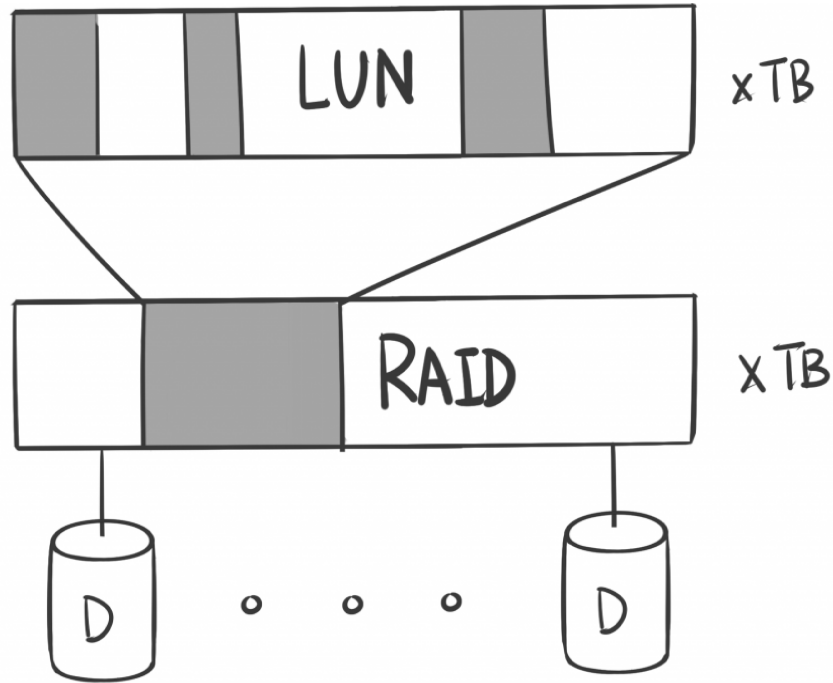


Figure 2-3: Storage Virtualization with Thin Provisioning

on several different characteristics, but the most common is I/O performance. High performance disks such as Solid State Disks (SSDs) or high performance Serial Attached SCSI (SAS) disks, usually have high I/O performance with low data capacity and are usually very expensive. Low performance disks such as Serial ATA (SATA) or Near-Line SAS (NL-SAS) sacrifice I/O performance for very high data capacity and are usually very cost effective. A virtualized storage solution can provide a LUN that has storage comprised from a number of high tiers of disks and

a lower tiers of disks which look to the OS as a single disk as depicted in Figure 2-4: Storage Virtualization with Auto-Tiering. As blocks within the LUN are heavily used, the system will automatically and transparently move them to a higher tier. Conversely, as blocks of data become idle, they are moved to lower tiers. This auto-teiring feature can help businesses reduce their costs while at the same time providing high performance. (Boles, 2011)

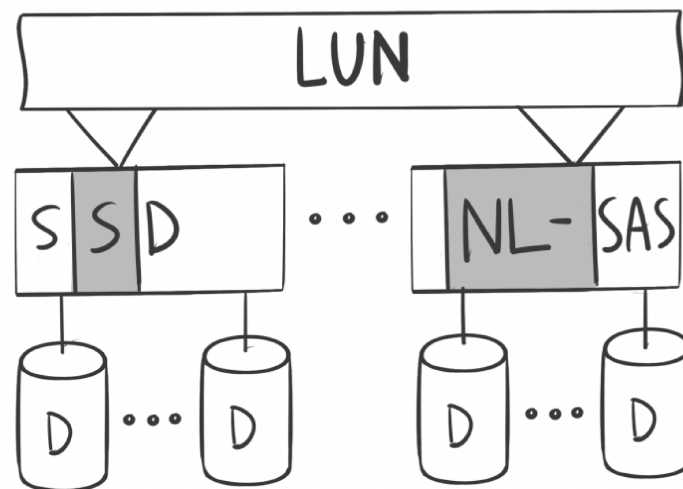


Figure 2-4: Storage Virtualization with Auto-Tiering

Some storage virtualization solutions can even entirely virtualize a LUN away from the storage system or array as depicted in Figure 2-5: Storage Virtualization of LUN Between Storage Arrays. This virtualization technique allows the LUN to be transparently moved from one storage array to another without downtime or having to reconfigure the server or OS. Such virtualization can be very useful in disaster recovery (DR) efforts as the same storage configuration can be used in both the primary and secondary sites. (Wikipedia, 2014c)

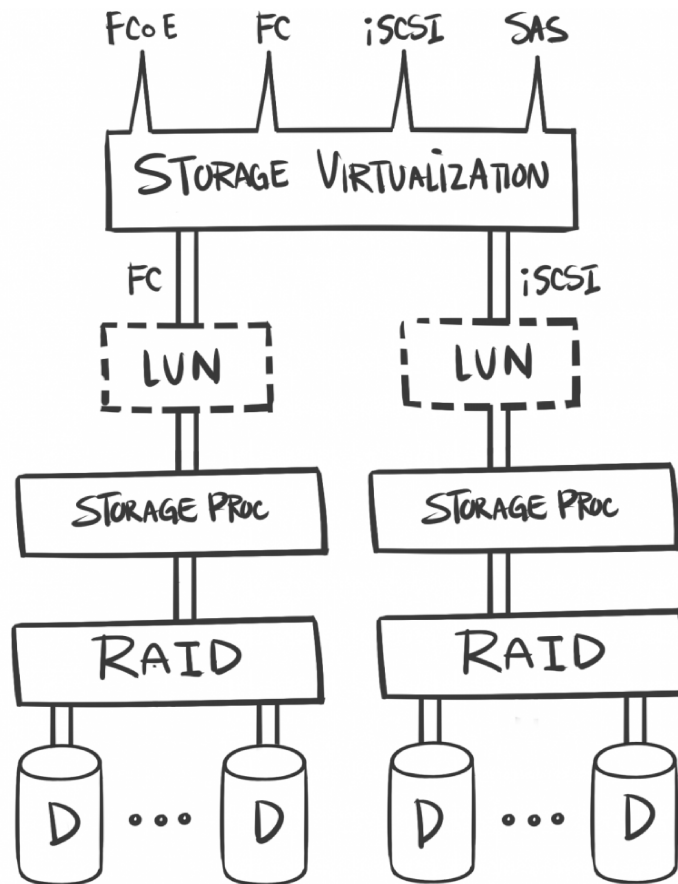


Figure 2-5: Storage Virtualization of LUN Between Storage Arrays

Although there are a number of documents and papers that describe these virtualization techniques and the vendors' claims for cost savings, there seems to be few academic papers proving the performance, reliability and cost savings of these virtualization techniques.

2.2.3 I/O Virtualization

I/O virtualization is one of the most challenging of these three areas of virtualization. Just as a machine with an Ethernet adapter won't operate on an Asynchronous Transfer Mode (ATM) network, I/O virtualization has similar challenges when interacting with traditional networks. However the data center has an advantage in this regard because it has a fairly clear line of

demarcation between the servers and access devices. This demarcation is usually found at a firewall or router where all servers exist on one side of the device and access devices are on the other. Since the data center is a controlled environment, the communication mechanism between this demarcation and the servers can be anything as long as when it gets to the router or firewall it is something the access devices can understand.

There have been many attempts to try to virtualize the Ethernet network and some of them include automatic bridge routing protocol, virtual LANs (VLANs), spanning tree and Multiprotocol Label Switching (MPLS) to name a few. (Wen, Tiwary, and Le-Ngoc, 2013) Although Fibre Channel (FC) has the ability to carry IP traffic, it never took the place of Ethernet in the data center and it has been used primarily as a storage network. IP over FC is being removed from vendor supplied drivers and even the Linux stack due to lack of use. (ServerFault, 2011) Infiniband (IB) also has the ability to transport both IP and iSCSI and provides massive amounts of bandwidth with extremely low latencies. The high performance of Infiniband has made it a favorite network in the high performance computing industry, but the perceived high cost and complex configuration has largely kept it from becoming a fixture in the traditional data center. (Byrne, 2012; Oracle, 2013)

2.3 History of Data Center Network Architectures

An abbreviated history of the evolution of the data center networks will be given to orient the reader regarding the study. The history represented here is mostly concerned with Ethernet, and Fibre Channel networks in a typical x86 architecture data center. There are many other configurations that could be discussed but for purposes of this study we will focus on this common configuration.

2.3.1 Separate Physical Networks

In the early days of x86 data centers each server could have several Ethernet connections, each for different purposes. There was a connection for the application data, a remote access connection and a connection for system backups to traverse across, so that the production network would not be affected during a system backup. Each of these connections were in a separate layer 3 domain which also meant they were in a separate layer 2 and layer 1 domain and had separate physical Ethernet switches. Such physical isolation was the only way to help ensure that traffic from one network would not adversely affect the traffic of another network.

Figure 2-6: Traditional Server Network Configuration shows a typical configuration which required a large amount of physical equipment and wiring. If redundancy was desired, then the amount of physical infrastructure would double.

As long as the number of physical servers stayed low, the management of the Ethernet network was tolerable, but as the infrastructure grew it was difficult to keep track of all the changes especially when almost all ports required manual configuration of speed and duplex configuration.

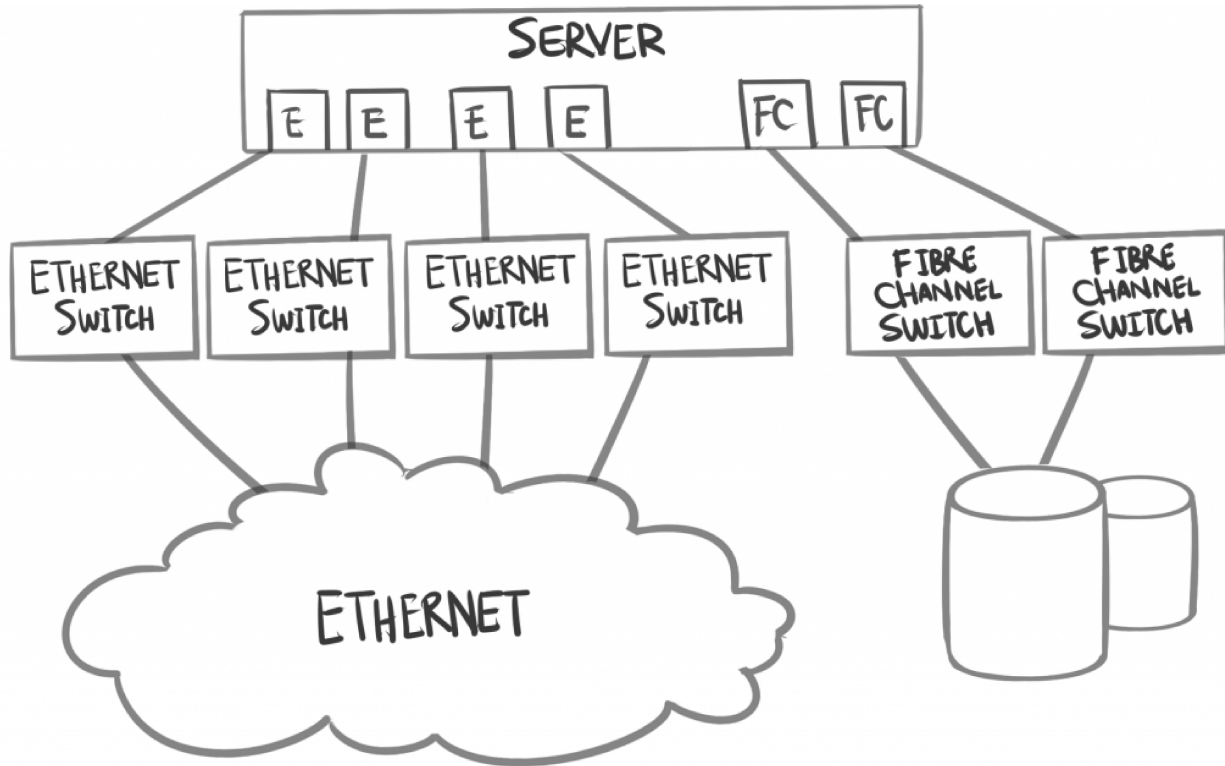


Figure 2-6: Traditional Server Network Configuration

2.3.2 VLAN Tagging

VLANs were introduced as a way to help reduce the number of physical switches required to provide layer 2 isolation. A VLAN creates a virtual layer 2 network that can only communicate with other switch ports that are designated for that VLAN. All broadcast and unicast traffic are confined to ports designated for that particular VLAN. VLANs also reduced the number of network interface cards (NICs) and cables needed by a server. If a server needed to communicate on more than one network then VLAN trunking is used which passes traffic for multiple VLANs on the same wire. In order to communicate on each of the VLANs trunked to a host, the host NIC driver creates a virtual NIC interface for each VLAN. The administrator then configures these virtual NICs with an IP addresses and routing as they would real physical NICs.

VLANs have been an important addition to Ethernet networks, but early VLAN implementations were susceptible to VLAN hopping where an attacker could persuade a switch to forward traffic of a VLAN to a port that was not configured for that VLAN. VLANs also do not address the concern about traffic on one network adversely impacting the performance of another network, but the VLAN standard does provide for a way to set priorities to packets. The handling of priorities is mainly up to the vendor and can differ between vendors, but the IEEE has made some broad recommendations as shown in Table 2-1: Ethernet Priority Recommendations. (Wikipedia, 2012b)

Table 2-1: Ethernet Priority Recommendations

PCP	Priority	Acronym	Traffic Types
1	0 (lowest)	BK	Background
0	1	BE	Best Effort
2	2	EE	Excellent Effort
3	3	CA	Critical Applications
4	4	VI	Video, < 100 ms latency and jitter
5	5	VO	Voice, < 10 ms latency and jitter
6	6	IC	Internetwork Control
7	7 (highest)	NC	Network Control

Setting the priority on network packets does not guarantee delivery of the packets or that a certain class of packets have a guaranteed bandwidth, it is a best effort attempt to try to get the higher priority packets through the queues first. For this reason sensitive networks or networks that have large amounts of traffic are still physically separated in order to guarantee isolation and reduce any possible impact.

2.3.3 Storage Area Networks (SAN)

Servers usually come with their own internal hard drives, but this creates some challenges in large data centers. In some cases a server is configured with too much disk space or not enough disk performance for the workload. Monitoring disk failures in each of the servers can also be difficult to manage because some RAID controllers will not communicate with the Operating System without complex configurations. Due to the small scale of storage in individual servers it is difficult to balance redundancy, performance, power and cost per unit of storage.

The Fibre Channel SAN was developed to help centralize the storage away from the individual servers and provide the capacity and performance to the servers as needed. This pooling of resources helps reduce the cost of storage while reducing the errors in over or under provisioning a host in terms of space and performance. Both space and performance can be adjusted from the central pool of storage. The Fibre Channel protocol was designed in the 1990's to be very thin to keep latency to a minimum. Fibre Channel has a protocol overhead of between 1.6% and 4.6% compared to TCP/IPv4 over Ethernet with 1500 byte frames of 5.1% to 6.1%. (Dykstra, 2001; Dedlk, Stephens, and Dedek, 1997; Complete Data Recovery, 2003) Due to the flexibility of the SCSI protocol it was implemented on top of Fibre Channel with little to no modification. This allows for SCSI devices to be connected to computers through a network instead of a bus inside of the computer. In SANs the storage can be centrally managed and monitored for many servers helping to reduce the overhead of storage management.

Although Fibre Channel had the ability to carry IP traffic at the time of its introduction, Ethernet was so pervasive and the defacto standard for server communications that Fibre

Channel was used primarily for storage traffic. (HP, 2011) Even though Ethernet storage existed many years before Fibre Channel was introduced, it was not an ideal storage transport for many applications due to its low throughput (10 Mbps compared to directly attached storage at 20x faster) and increased latencies. (Kerekes, 2004; Komiega, 2010)

With 10 Gbps Ethernet becoming more widely adopted and Fibre Channel commonly at 8 Gbps, there is little performance benefit today for using Fibre Channel. Still Fibre Channel is near and dear to the heart of enterprise data centers because of the reputation it has developed over the years. (Mearian, 2012) Additional benefits of Fibre Channel compared to Ethernet include:

- Separate network from server communication provides protection from congestion and failures of the Ethernet network
- Link failures in Fibre Channel are failed over faster than rapid spanning tree
- Fibre Channel provides multiple active paths through different switches and load balancing over available paths
- Zoning allows the same adapter to easily participate in multiple virtual networks

These benefits have made Fibre Channel a difficult protocol to move away from.

2.3.4 Server Communication Networks

Server communications between other servers and clients generally take place on an Ethernet network. The performance of Ethernet is usually sufficient for most applications, but there are some cases such as financial transactions, high performance computing and large database clusters where more performance is required. For these high bandwidth and/or low

latency situations a high performance network is deployed in addition to Ethernet and Fibre Channel. (Musich, 2008)

This high performance network, such as Infiniband, provides not only high bandwidth (56 Gbps per 4x link as of this writing) and low latencies (ping times measured in microseconds) but also provides additional features such as Remote DMA (RDMA) for data transfers and userland optimizations to reduce CPU and I/O loads on the servers. Infiniband drivers configure special memory locations in each host to reduce or eliminate expensive privileged kernel page copy operations. By allowing an userland application to write directly to the Infiniband buffers in a safe way, the CPU load and latency is reduced by eliminating the kernel's need to copy buffers to different memory locations. (Mellanox, 2014; Oracle, 2012; Haviv, 2009)

Quadrics and Myrinet are other high performance communication networks but because of the open and collaborative nature of the Infiniband consortium between vendors these solutions eventually gave way to Infiniband. The perceived high cost and complex configuration of Infiniband has kept it as a niche product even though it is cost effective and similar to operating Ethernet or Fibre Channel networks. (Kim, 2004; Morgan, 2009) Even though Infiniband is cost competitive to Ethernet, Ethernet is built into the motherboards of most server systems and adding the additional I/O cards for Infiniband when 10 Gbps Ethernet is sufficient is not cost effective. This also makes it difficult for Infiniband to enter into the traditional data center.

2.4 I/O Virtualization Techniques

What follows is a discussion of some of the modern attempts to virtualize and converge Ethernet, Fibre Channel and server communication networks onto a single fabric. These

solutions enable a single cable to carry two or more traditional protocols and still inter operate with traditional Ethernet and Fibre Channel networks as shown in Figure 2-7: Communication Virtualization Logical View. The basic premise of these solutions is to encapsulate the foreign protocol and route it through the network to a gateway device. The gateway then takes the encapsulated data and forwards it on the traditional network. When data is destined for a machine on the fabric, the gateway reads the data off the traditional network, encapsulates it and forwards it to the host on the fabric.

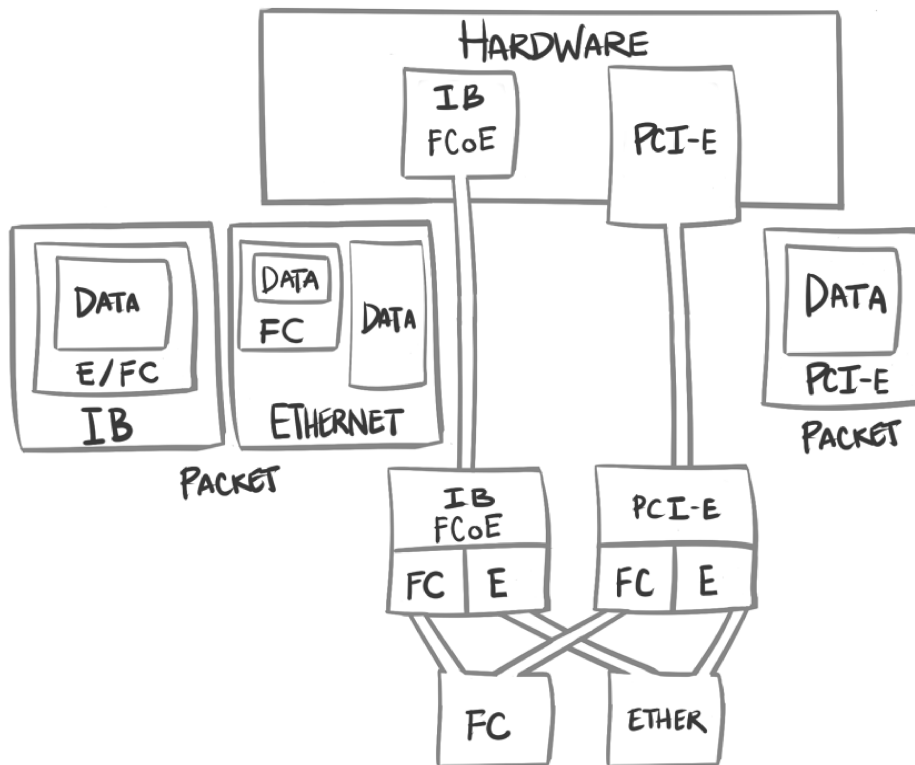


Figure 2-7: Communication Virtualization Logical View

Each of these technologies are still in their infancy and are trying to claim a majority of data center implementations. (Haff, 2009) There has been little independent research published about these new technologies and this is where this work adds value.

2.4.1 Fibre Channel Over Ethernet (FCoE)

Ethernet is the most ubiquitous networking solution in data centers around the world and it makes sense to try to virtualize or converge onto it. Fibre Channel Over Ethernet attempts to be a standards based way to carry Fibre Channel traffic over fairly common Ethernet networks. With 10 Gbps Ethernet becoming more common in the data center and many servers not requiring 20 or more Gbps to perform their functions. The 'extra' bandwidth could logically be used for storage traffic.

Although iSCSI and Network Attached Storage (NAS) protocols already provide storage over Ethernet, there are a few drawbacks that make it less than ideal for enterprise storage. (Cisco, 2009)

- Encapsulating storage data into Layer 3 Ethernet frames adds significant overhead especially if standard 1500 byte frames are used for compatibility reasons.
- Ethernet is a best-effort transport medium with no guarantee of packet delivery. Either a guaranteed protocol like TCP has to be used which suffers terribly when there is packet loss, or the application has to ensure that all packets are received and ordered correctly in UDP communications.
- Most Ethernet switches use a store and forward procedure for switching packets which adds latency to the storage system. Most 10 Gbps Ethernet switches are implementing

cut through switching which help with latency, especially when jumbo frames are enabled. (Cisco, 2008; Duffy, 2011a)

In order to attempt to overcome these drawbacks, the Data Center Bridging (DCB) standard has been introduced which extends the Ethernet standard to include the following: (HP, 2011)

- Priority-based Flow Control (PFC), 802.1Qbb allows the network to pause different traffic classes.
- Enhanced Transmission Selection (ETS), 802.1Qaz defines the scheduling behavior of multiple traffic classes, including strict priority and minimum guaranteed bandwidth capabilities. This should enable fair sharing of the link, better performance, and metering.
- Quantized Congestion Notification (QCN), 802.1Qau supports end-to-end flow control in a switched LAN infrastructure and helps eliminate sustained, heavy congestion in an Ethernet fabric. Before the network can use QCN, you must implement QCN in all components in the Converged Enhanced Ethernet (CEE) data path (Converged Network Adapters (CNAs), switches, and so on). QCN networks must also use PFC to avoid dropping packets and ensure a lossless environment.
- Data Center Bridging Exchange Protocol (DCBX), 802.1Qaz supports discovery and configuration of network devices that support PFC, ETS, and QCN.

Due to the number and types of enhancements in the Ethernet standard to enable FCoE, many Ethernet switches will not be able to be upgraded by software to include these new features. In order to support FCoE throughout the data center all-new Ethernet infrastructure will have to be purchased and deployed. Additionally, vendors' interpretations of the FCoE standard allowed for some interoperability issues to occur between vendors. Due to these challenges many people are waiting to see how FCoE matures and those who are implementing it now are doing so only at the edge where ROI is quick and easy and there isn't as much uncertainty with compatibility. (Foskett, 2011; Munjal, 2011; The Data Center Overlords (blog), 2011)

Although cable reduction and management aren't strong arguments in favor of FCoE, it does have some benefits that can help in the data center. Single hop FCoE can do much to reduce the number of adapters needed in servers and the number of switches needed in blade chassis or at the top of rack for rack servers. In many cases, a vendor's CNA is already built into the server motherboard and is included with the cost of the server. Even with buying converged switches, which are more expensive than standard switches, you still save money by not having to buy Fibre Channel switches or adapter cards. Vendors claim that although cable costs are not reduced, significant savings can be realized through the reduction in hardware footprint with corresponding reductions in power consumption, heat generation and management expenses. (Munjaj, 2010; The Data Center Overlords (blog), 2011; Duffy, 2011b)

2.4.2 Infiniband

Infiniband, as mentioned previously, is a very high performance network that seems very suitable for virtualizing or converging network traffic onto. (Mellanox, 2014; Haviv, 2009) Some of these characteristics are: (Oracle, 2012)

- A very efficient data encoding algorithm in which every 10 bits on the wire carries 8 bits of data and is improved with Fourteen Data Rate (FDR) to 64 bits for every 66 bits sent.
- Low protocol overhead.
- Guaranteed delivery.
- Automatic link aggregation
- Fast fail over of failed links or components.
- Congestion avoidance and load balancing.

- RDMA for transfers between systems which can greatly reduce the CPU overhead in systems that transfer a large amount of data.

As part of the Infiniband standard, IP over IB (IpoIB) was developed for providing traditional layer 3 Ethernet type communication on the Infiniband fabric. However a number of challenges prevent it from operating with standard Ethernet networks very well. Since there is no layer 2 protocol information in IPoIB, (Network Working Group, 2006) an Infiniband to Ethernet gateway has to construct such data from very little information and it makes things such as VLANs nearly impossible. (Ayoub, 2012) Two storage protocols were also defined as part of the standard, SCSI Remote Direct Memory Access Protocol (SRP) and iSCSI Extensions for RDMA (iSER) which also proved to be difficult to use when leveraging an Infiniband to Fibre Channel gateway for the same layer 2 reasons as Ethernet. (Cisco, 2007; HP, 2011) Most IPoIB, SRP and iSER traffic is usually kept within an Infiniband fabric where it is much easier to manage.

Although Infiniband has had difficulty interacting with traditional networks in the past, in the last several years there have been advancements in the technology to completely encapsulate the layer 2 frames of Ethernet and Fibre Channel. This greatly increases the ability of these solutions to interact with more traditional networks in a way that administrators of those networks expect. (Oracle, 2013)

The largest difficulty for Infiniband to become a mainstream data center network is overcoming some of the misconceptions about the technology. There is a general idea that Infiniband is a dying technology, but Infiniband has been alive and growing in the high performance computing (HPC) arena. It is also perceived that Infiniband is more expensive, but costs are comparable to high speed Ethernet. Many people believe that Infiniband is too

complicated to configure and maintain. Infiniband is a different technology and does require a different tool set than what is traditionally used for Ethernet management. (Chu, 2012) It probably doesn't help perceptions that Infiniband has been a niche technology in HPC and other complicated configurations.

2.4.3 PCI-E Virtualization

PCI-E virtualization attempts to relocate the PCI devices from within the server and consolidate them into fewer devices that are shared between several servers. The servers are configured with a passive PCI-E bus extension card and are connected to gateway devices which house the PCI-E devices using high speed PCI-E cables. The PCI-E bus is a very different animal from previous generations of the PCI bus. PCI-E uses serial communication rather than parallel communication as in previous generations so it is able to overcome some of the challenges with high clock rates, cross talk and noise. The PCI-E protocol is also a packet based protocol which makes it much more flexible with how devices can be configured. The protocol natively supports I/O virtualization (IOV) of PCI-E devices in a Single Root Complex IOV or a Multi Root Complex IOV configuration. (Emerick, 2012)

For PCI-E virtualization to work, the physical PCI-E devices need to be compatible with either a Multi-Resource I/O Device configuration or a Sharable I/O Device configuration. A Multi-Resource I/O Device configuration is where a physical port on the PCI-E device can be mapped to one and only one Operating System. A Sharable I/O Device configuration is where a single port on the PCI-E device can be mapped to many Operating Systems. In the Sharable I/O device configuration the Operating System believes that it has full access to the physical port, but the hardware manages multiple queues internally. (Emerick, 2012)

Single root IOV is pretty straight forward and has benefits which can help certain workloads like hypervisors. Because of the ability to map resources directly to virtual machines, the hypervisor no longer needs to create the virtual I/O adapters in software and the hardware can be leveraged to perform the same function. This offload to the hardware can translate into large performance gains for the guest operating systems especially for network based traffic and modest gains in storage performance. When multiple physical hosts want to share the same physical ports then multi root IOV is required. This makes PCI-E virtualization a bit more complex as it requires all the hardware to support multi root IOV as well as additional software management of the PCI-E devices are needed to properly present the PCI-E devices to hosts even when they are offline compared to single root IOV. (Emerick, 2012)

PCI-E virtualization helps to reduce the cost of I/O like other technologies by reducing the number of adapters required in each server as well as increasing the utilization of the adapters. Based on the information gathered, PCI-E virtualization still seems to be quite young and not deployed in as many environments as the previously discussed technologies.

2.5 Industry Implementation of Communication Virtualization

For each of the virtualization technologies there are one or more vendors which implement the technology. The following sections represent the classification of vendors' technologies into the aforementioned categories along with any noteworthy features or functions.

2.5.1 Cisco Unified Computing System (UCS)

Cisco UCS is probably the name most people think of when talking about communication virtualization or network convergence. Cisco has put a lot of money into building and promoting

UCS and to some degree has leveraged its network prominence to promote its UCS line. Cisco has prevented server manufacturers like HP and Dell from producing modular blade chassis versions of the Cisco Nexus 10 Gbps switches. (Modine, 2010) This has caused many businesses which only use Cisco networking technologies either to remain at 1 Gbps for Ethernet, move to UCS or to do something entirely different. For the most part many are choosing to go with UCS and for good reasons. UCS is fairly easy to configure and maintain and the single cable wiring for both Ethernet and Fibre Channel is convenient. (Oltsik, 2009)

A UCS blade enclosure can house four full width blades with up to four processors each or eight half width blades with up to two processors each. There are few options for mezzanine cards for UCS since the specific Cisco FCoE protocol must be used. UCS uses a flat layer 2 network for its fabric and since it only supports single hop, it is not possible to create loops which require spanning tree. Cisco mentions that even though there is a hop from the CNA to the Fabric Extension (FEX) module on the back of the blade chassis then a hop to the UCS director it is still technically one hop as the switch like device is really a pass through device. This is somewhat evident in the fact that it requires a multiple of two cables (1,2,4) for it to multiplex the server traffic properly. (Cisco, 2012; Sultan, 2009)

Cisco UCS is proprietary and like many of these technologies will only work with Cisco servers and certified CNAs. Considerations must be made when deploying UCS as it will not virtualize the entire data center fabric and you will be tied to a single vendor for servers and switches. Because of the strict hardware requirement of UCS however, they are able to do some very useful things such as ensuring that all hardware is operating at a certain firmware version and the ability to move a server's identity including all BIOS setting, MAC addresses and

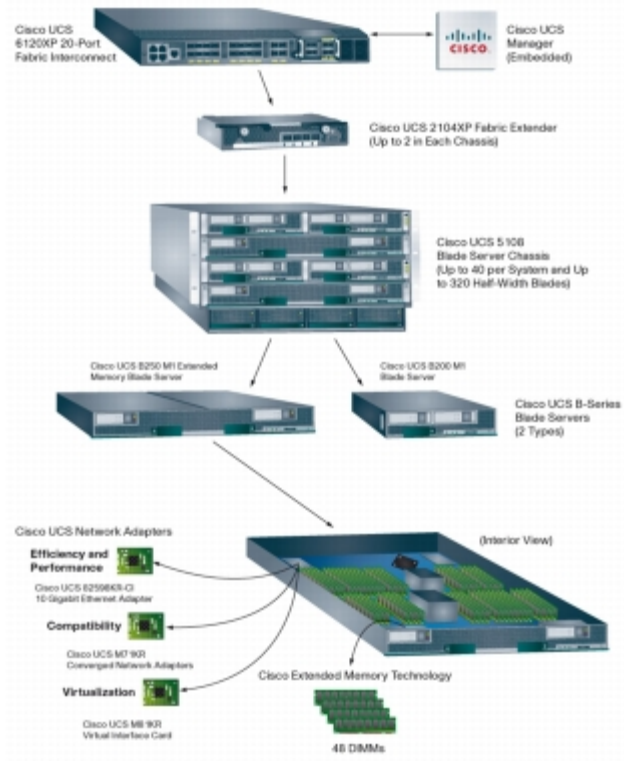


Figure 2-8: Cisco UCS Blade Architecture (Cisco 2012)

worldwide names (WWNs) to a new physical server with very little effort. This can be a critical feature when implementing many discrete physical servers in an environment. UCS directors are deployed in pairs for redundancy and are designed for top of rack or end of row installation and can not provide redundancy for chassis between rows. If your workloads are network intensive then an important consideration is that the maximum guaranteed traffic to a server blade is 20 Gbps for both Ethernet and Fibre Channel using newer fabric extender modules. (Cisco, 2010; Cisco, 2012)

2.5.2 Other FCoE Manufacturers

Brocade, Dell, HP, IBM, Juniper and other manufacturers provide more of the standard FCoE type offering however they are still somewhat divided into two camps that provide FCoE a little differently. Converged Enhanced Ethernet (CEE) and Data Center Bridging (DCB) handle the FC packets a little differently, but each of them provides some additional value-add when used in the vendor's preferred mode. These technologies can support server profiles, but they are not as feature rich as Cisco UCS; and they have a much broader ability to be installed into any server manufacturer and form factor desired. (HP, 2011; Dell, 2010)

In cases of HP and Dell servers, the CNA is already built into the motherboard and no add on card is needed. These vendors also provide the ability to move the MAC addresses and WWNs of the adapter (or server profiles) to other servers sometimes even between discreet blade chassis, but only within their brand. With the virtualization technology at the edge of the network, it can potentially create many management silos which can become increasingly difficult as the environment grows. (HP, 2011; Dell, 2010)

Substantial savings can be realized by adopting one of these vendors and they still have the ability to expand to meet challenging bandwidth requirements by adding additional CNA adapters to the server. The expected progression of FCoE from the access device to the core allows time for the multi-hop FCoE and differences between CEE and DCB to be resolved which helps give confidence in more generic FCoE implementations such as these. For the moment FCoE does little to help reduce the cabling costs and maintenance throughout the entire data center, but it makes a good start by tackling the part of the network with the most capital expense. (HP, 2011; Dell, 2010)

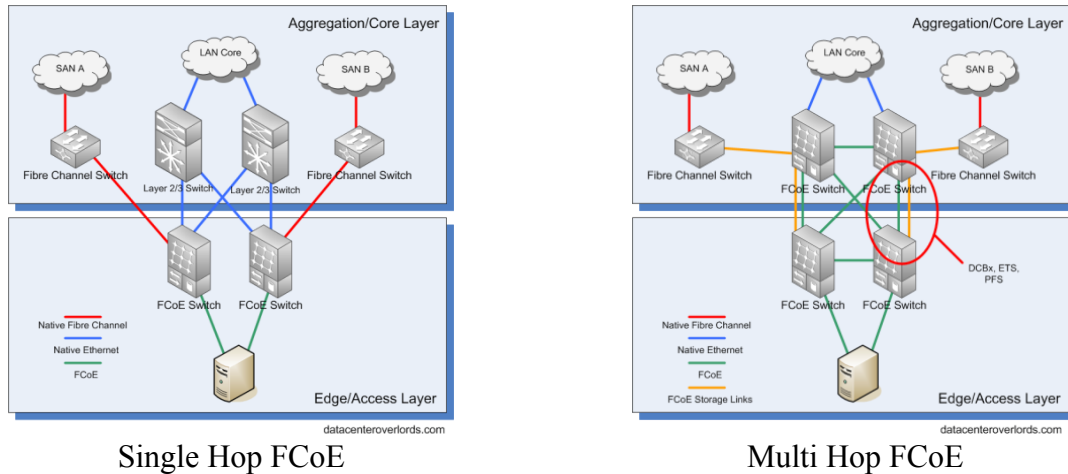


Figure 2-9: A Comparison of Single and Multi Hop FCoE

2.5.3 Oracle Virtual Networking (OVN previously Xsigo)

OVN is one of the two products that are primarily focusing on virtualizing communications on top of the Infiniband protocol. OVN uses standard Infiniband host channel adapters (HCA) and switches to provide a data center fabric. The OVN directors provide a subnet manager and backup subnet manager which configures the Infiniband fabric for the administrator. OVN then uses a proprietary encapsulation method to pass unaltered layer 2 frames of Ethernet and Fibre Channel to the HCA. The OVN driver presents to the host one or more virtual network and/or Fibre Channel adapters that act like traditional adapters. (VMware, 2012; M2 Presswire, 2012) Because the proprietary encapsulation traffic runs on top of a standard Open Fabrics Enterprise Distribution OFED stack, all other standard RDMA, SRP, iSER and functions beneficial to Message Passing Interface (MPI) can be leveraged at the same time. This along with OVN's Server Fabric allows hosts to communicate between each other at native Infiniband speeds without any need to reconfigure applications. (Poulton, 2009)

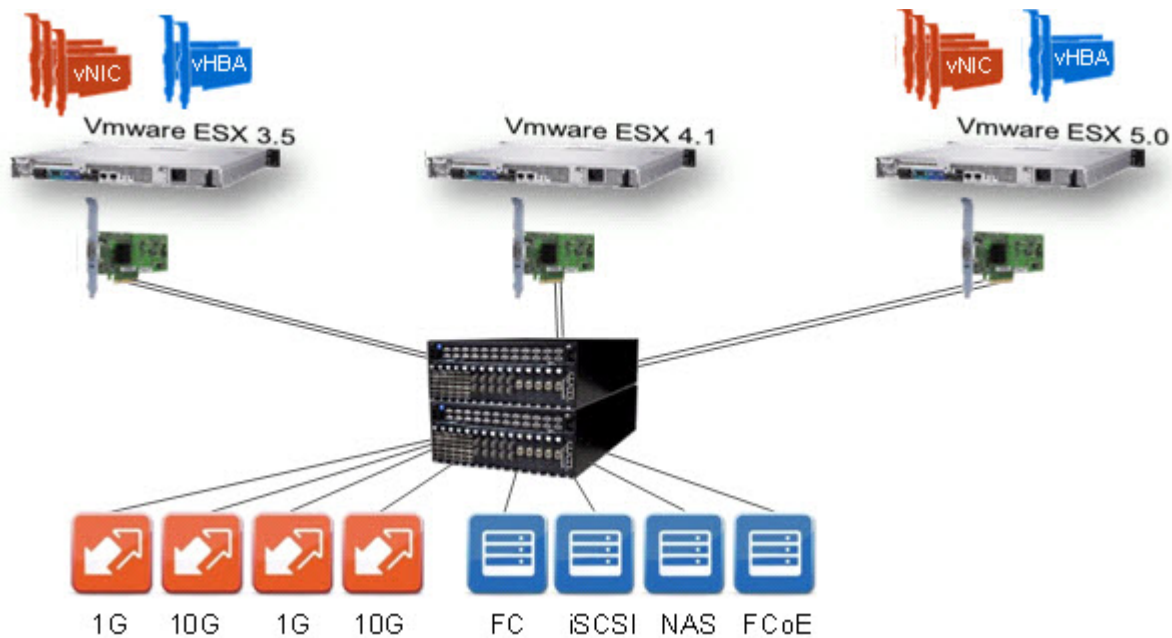


Figure 2-10: OVN Configuration

OVN has made running Infiniband in the data center very easy. Since the subnet manager configures the network and it is included in the OVN director all that is required by the administrator is to correctly physically plug the servers to the switches. Due to standard Infiniband components being used, any server that can install a HCA can have its traffic converged. Because there is no hop restriction to OVN's implementation, the fabric can cover the entire data center reducing costs and components throughout. The OVN directors can also be placed at any location in the fabric allowing them to be located near points of demarcation or separated in different fault domains within the data center. (Poulton, 2009)

Speed and costs of Infiniband are other factors that have to be weighed. With Quad Data Rate (QDR) speeds being the norm and FDR recently introduced each server has two 40 Gbps ports or two 56 Gbps ports available for Ethernet and Fibre Channel traffic. The HCAs for the servers are generally more expensive than a 10 Gbps Ethernet adapter and slightly more expensive

than a 10 Gbps CNA. Infiniband switches are generally much less expensive than Ethernet or Converged switches especially when the cable count can be reduced by four or five times. In order to provide complete I/O redundancy to a blade chassis of 16 blades (two switches in the blade chassis to two separate upstream switches) it will require 4 cables. Those four cables with QDR speed can guarantee each server 10 Gbps of Ethernet and Fibre Channel traffic, the same that takes 16 cables with Cisco UCS for instance. (LeBlanc, 2011a)

Table 2-2: Comparison of OVN and Cisco Cabling

# Blades	# Cables	Chassis Bandwidth	Blade Bandwidth
16 Dell/HP	4	160 Gb/s	10 Gb/s
8 Cisco	4	40 Gb/s	5 Gb/s
16 Cisco	16	160 Gb/s	10 Gb/s

OVN has the ability to share any I/O module with any server, but there is a limit to the number of servers that can use a module at the same time. Still a benefit of using Infiniband with its large bandwidth capability is the ability to deploy new technology without having to upgrade the infrastructure. When 16 Gbps Fibre Channel becomes widely available, the fabric can already leverage the full potential immediately, the same goes for 40 Gbps Ethernet. OVN also has the ability to move MAC addresses and WWNs of a server between any servers in the fabric, even if they are from different vendors or different form factors. The largest challenge with OVN is deciding to implement Infiniband in the data center and at least be aware of the technology and getting over the incorrect perceived notions about Infiniband, otherwise it is a very similar management feel to FCoE or Cisco UCS. (Poulton, 2009)

2.5.4 Mellanox Bridge-X

Mellanox's Bridge-X solution is also built on Infiniband like OVN, but the major difference is that Mellanox is trying to extend the Infiniband protocol to include Ethernet over Infiniband (EoIB), Ethernet Tunneling over IPoIB (eIPoIB) and Fibre Channel over Infiniband (FCoIB). The adoption of these protocols have been very slow and EoIB has been depreciated in favor of eIPoIB which provides the same benefits but with much better performance. The demand for traditional Fibre Channel over Infiniband has been so low that Mellanox has provided Application-Specific Integrated Circuit (ASIC) functions for FCoIB, but has not developed the software to take advantage of it. Most people deploying Infiniband and using storage over it require the high performance that SRP and iSER provide and are not as interested in compatibility with “older” protocols. (Mellanox, 2012)

Up to this point Bridge-X has been targeted primary at high performance computer installations due to their current investments in Infiniband. Mellanox has in the last couple of years began work to include hypervisors in their supported Operating Systems in an attempt to enter the traditional data center market. Theoretically, it could be easy for data centers to move between OVN fabrics and Bridge-X fabrics since they both use the same standard Infiniband components. It should also be possible to run both solutions on the same fabric providing an easy migration from one to the other. Bridge-X can be a viable converged infrastructure if there isn't a need for Fibre Channel compatibility at the moment.

2.5.5 NextIO

NextIO provided PCI-E virtualization in which a low cost pass through PCI-E module is installed in each server and is connected to an I/O Maestro device. The I/O Maestro holds

multiple PCI-E modules which can be mapped to servers and is typically installed at the top of rack. There are no configurations of IO Maestro that support large fabrics most likely due to the restricted cable length of 3 meters. (Morgan, 2011)

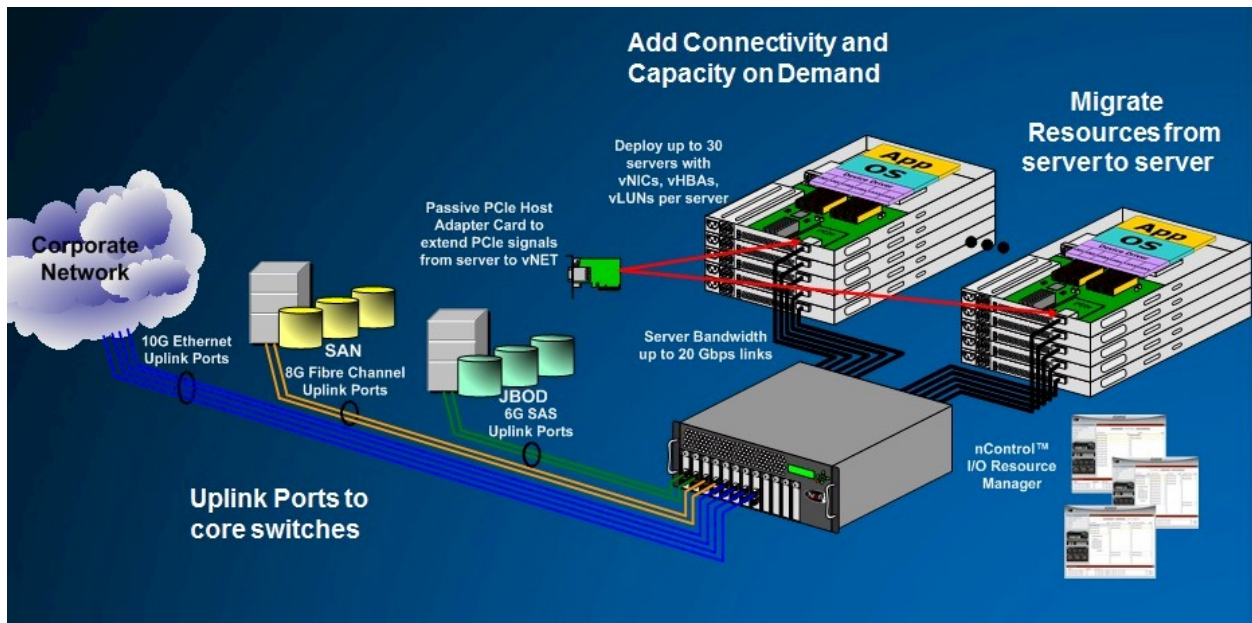


Figure 2-11: NextIO Communications Virtualization

One of the interesting thing about NextIO is their ability to house several GPUs in a small footprint that can be dynamically assigned to servers in their NetCore product. These GPU solutions are useful for high performance compute applications that tend well to the use of graphic processors for computation. (Morgan, 2011)

Although NextIO is standards based using the new PCI-E virtualization protocols, it was the only major player in this technology. Although some costs savings can be realized it may not be as much as other technologies that have much more competition in the market. There are also no blade manufacturers that support NextIO and so only rack mount servers would be able to

leverage this technology. (Morgan, 2011) NextIO reported on 19 Aug 2013 that it has closed its operation. (Maleval, 2013)

2.6 Conclusions

The implementation of machine and storage virtualization in the data center has driven the desire for network convergence. The majority of the information available for network convergence is vendor specific or so specialized that one can not draw general conclusions applicable to other data centers. There is also a lack of independent validation that compares traditional network technologies with converged network technologies or to other converged network technologies. We have found no independent results that prove the feasibility of the technology in the data center, only marketing literature which states that it possible, simpler, and less expensive.

3 METHODOLOGY

3.1 Impetus for the Study

There is very little independent data in the converged I/O space and a lack of experimentation to prove the validity of this technology outside of vendor claims. Many vendors specify that they make some percentage of improvement over some unspecified technology which provides no useful information. In order for data center engineers and architects to make

informed decisions about these technologies, the classification and comparison of the technologies needs to be performed in an unbiased manner that corresponds to a typical configuration that exists in their own data centers.

The literature seems to focus on several key factors that should be validated through experimentation, namely:

1. Cost of the converged communication technology from a capital expense (CapEx) standpoint.
2. Increased flexibility to reconfigure the fabric quickly and with as little disruption as possible.
3. Reduce manpower and other operational expenses (OpEx) by having less equipment and cabling to maintain.
4. Performance on par or better than a non converged counterpart.

An accurate comparison of the strengths and weaknesses of different solutions requires that they are compared as evenly and consistently as possible. The results of the experiment should be able to be validated by third parties by simply following the testing procedures. Vendor's results are not reliable because they do not fully disclose the testing environment, the equipment, and method used in their testing. It is impossible to know how fairly the competition was compared to their own products.

To overcome these shortcomings, an analysis framework is presented which outlines testing methodologies to be used in the generation of converged I/O tests. This analysis framework facilitates unbiased, statistically valid comparisons between multiple solutions.

3.2 Introduction to Analysis Framework

The analysis framework for this work will be divided into three focus areas: performance, usability and costs (encompassing both CapEx and OpEx). The focus areas of the analysis framework facilitate constructing tests that compare multiple vendor solutions as fairly and completely as possible; creating a complete picture of the technologies from multiple perspectives.

3.3 Performance Testing

Vendors have stated performance maximums, but these are generally special case maximums and can be much different from what is observed in real-world situations. The goal of this focus area is to test technologies as evenly and as close to real-world situations as possible. Vendor solutions may be implemented very differently from each other, but the tests between systems must be as similar as possible. Leveraging server virtualization technologies may help reduce the variability of the testing platform to some degree and help keep the tests between systems as similar as possible.

Tests should be designed to push the technology past the stated performance maximums to understand where the actual maximum occurs and what happens during situations of high stress and congestion. Pushing the system past the limit exposes areas where the technology completely fails or needs improvement.

Statistically designed experiments ensure complete coverage of the possible variables that could affect performance and allow strong comparisons to be made between solutions. The statistical test should be a reduced factorial design to reduce the amount of testing, but still

provide a strong statistical analysis. (Rice, 1995) A statistical analysis will expose information from the data gathered that may not be obvious at first glance.

Performance data that is relevant for comparing technologies include: bandwidth, latency, dropped packets, jitter and IOps. All interesting forms of traffic should be tested to ensure each performs as expected.

3.4 Usability

Important factors when considering new technologies are usability and how knowledge transfers from the current to the new technology. The usability testing should cover common tasks that administrations would be expected to perform on the technology. The time to complete the tasks as well as the number of errors performed should be recorded and analyzed to see how well existing knowledge transfers to the new technology and how quickly it can be learned.

A combination of scalar and free form questions should be asked of each participant to help understand subjective components of the system and compare that to the objective results from the tasks section. This will help determine if a technology is more usable and/or more efficient than another. Efficiency is not necessarily tied to usability and visa versa, so it is important to test for both characteristics.

3.5 Costs

CapEx costs generally the driving factor when purchasing equipment, but the long term OpEx costs of running and maintaining the equipment may not generally be considered in a purchasing decision. A technology that is inexpensive to acquire may cost more in the long run to

operate. This area of the framework should expose the immediate CapEx costs of the technology as well as the long term OpEx costs.

Comparisons should be made as evenly as possible, using equivalent configurations where possible. List pricing should not be compared to discounted prices, nor should peak wattage be compared to average wattage. A standard should be set and followed for all technologies normalizing values to the standard if necessary.

CapEx costs include hard expenses required to get the technology running such as: equipment, installation, cabling, supporting infrastructure, etc. OpEx costs include expenses that keep the equipment running after installation such as: power, cooling, maintenance, support, etc.

4 CASE STUDY PROCEDURES

4.1 Overall Approach

A case study was implemented to validate the framework presented in this work. In this study two solutions were examined, Cisco UCS represented FCoE technologies and Xsigo (now Oracle OVN) represented Infiniband technologies. The selection of the two technologies is based on a number of institutional requirements which complicate the adoption of converged I/O at BYU. First is the requirement that Ethernet switches in the data center must come from Cisco. This is due to some poor experiences in the past trying to integrate switches from other vendors into our current infrastructure. Second, electrical power has been historically the largest physical constraint in the BYU data center. The university has standardized on blade server architectures due to their power efficiencies. This reduces the power consumption and the heat generation of the servers but comes with a drawback of limited I/O connectivity in the blade systems. The challenge is to find a converged solution that lowers costs, increases bandwidth, is easy to use, and fits within the constraints of switch and server requirements that exists. (Houston, 2010)

With these constraints, any solution that is designed solely for rack mount servers such as Next I/O was eliminated. Solutions which require Ethernet switches from Brocade, Dell, Force10, HP, IBM, Juniper, and other non-Cisco systems were also eliminated. Mellanox and Xsigo technologies are not Ethernet switches so they are considered as candidates, but since

Mellanox does not have Fibre Channel capability to interface with the large amount of Fibre Channel storage already in use, it will not be considered as part of the study. This left Cisco UCS and Xsigo Systems as part of the study. These solutions are the most mature in their respective space and allow for the best chance of success.

Both of these technologies were implemented side by side in a controlled environment running VMware ESXi 4.1 workloads. In order to provide minimum redundancy, four Infiniband cables were used between the Xsigo directors and blade chassis. In order to match bandwidth capacity, Cisco was configured with 8 cables for an 8 blade enclosure to the fabric interconnects, its maximum configuration. The vendors were given adequate time to prepare the environment before testing began. Both environments were run through a series of statistical hypothesis tests to compare performance between them. A mock-up of normal day to day tasks was developed and a group of individuals who are familiar with server and network administration were asked to perform the tasks. Additionally, CapEx and OpEx costs were analyzed to validate cost savings.

A capstone team from the Information Technology program in the School of Technology at BYU in 2010-2011 designed, built, and executed the performance and usability tests using the analysis framework and under the direction of the data center staff. The capstone team consisted of Stefano Gessati, Aaron Kimbler, Chase Nebeker, Francisco Parra and Jordan Sheen.

4.2 Measuring Performance

JMP statistical software was used to generate the design of experiments and configuration of 20 VMs to ensure complete coverage of the several variables of the virtual machines including: Operating System (Windows 2008 Server or Ubuntu Linux), number of virtual CPUs (1 to 4), RAM (512 MB to 16 GB), CPU load (0 to 100%) and RAM load (0 to 100%) as shown

in Table 4-1: Virtual Machine Design of Experiments. Ten virtual machines were located on one physical host and the other ten were on a second physical host. The reduced factorial design of the experiment requires only the min and max of the variables for proper statistical analysis. Although less than 20 virtual machines were needed to perform the statistical analysis, the additional VMs allowed for additional data points without much additional overhead.

The 20 VMs resided on the same storage system and were migrated between Cisco and Xsigo systems to provide identical testing configurations. The general null hypothesis for the statistical hypothesis testing was that there was no difference between Cisco and Xsigo. The general alternative hypothesis was that Xsigo performs better than Cisco. This hypothesis was fabricated before any of the tests were performed based on the fact that Infiniband has higher bandwidth, lower latency and a smaller protocol stack than Ethernet.

Table 4-1: Virtual Machine Design of Experiments

Host	Name	OS	CPUs	RAM (MB)	CPU Load (%)	RAM Load (%)
A	cn1	Windows	1	512	100	100
A	cn2	Windows	4	16384	100	100
A	cn3	Linux	1	512	100	100
A	cn4	Linux	4	512	0	0
A	cn5	Linux	4	16384	0	100
A	cn6	Linux	1	16384	100	0
A	cn7	Windows	4	512	100	0
A	cn8	Linux	1	512	0	0
A	cn9	Windows	4	16384	0	100
A	cn10	Windows	1	16384	0	0
B	cn11	Linux	4	512	0	100
B	cn12	Windows	4	512	0	0

Table 4-1: Virtual Machine Design of Experiments (cont.)

Host	Name	OS	CPUs	RAM (MB)	CPU Load (%)	RAM Load (%)
B	cn13	Linux	4	512	100	100
B	cn14	Linux	1	16384	100	100
B	cn15	Windows	1	512	0	100
B	cn16	Linux	1	16384	0	0
B	cn17	Windows	1	512	100	0
B	cn18	Windows	4	16384	100	0
B	cn19	Windows	1	16384	0	100
B	cn20	Linux	4	16384	100	0

4.2.1 Measuring Ethernet Performance

Ethernet performance was measured between virtual machines using iperf. Iperf was chosen so that Ethernet testing would not be impacted by the performance of the disk system at the source or destination. Iperf generates the data used in the transmission testing from the system memory without the need for disk access. Each VM connected to its partner on the other physical blade (i.e. cn1 communicated with cn11) and transferred as much data as quickly as possible for the TCP tests.

Table 4-2: Ethernet Testing Configurations shows the multiple tests that were conducted with iperf using TCP for bandwidth measurements and UDP for packet loss and jitter in contention situations. Each VM in the UDP test attempted to consume 10 Gbps of bandwidth which over saturated the network and helped us determine what happens when the system is pushed to the extreme. These tests were run with standard 1500 byte frames and with 9000 byte jumbo frames to compare the performance of different frame sizes. Tests were also performed

between blades in the same chassis (B2B) and between blades in different chassis (C2C) to see if the number of switch hops impacted the performance.

Table 4-2: Ethernet Testing Configurations

ID	Type	Jumbo Frames	Blade 1	Blade 2	Interval	TCP/UDP
1	B2B*	On	Zinc-ib1	Zinc-ib2	300	TCP
2	B2B	Off	Zinc-ib1	Zinc-ib2	300	TCP
3	B2B	On	Zinc-ib1	Zinc-ib2	300	UDP
4	B2B	Off	Zinc-ib1	Zinc-ib2	300	UDP
5	B2B	On	Cisco1-sc	Cisco2-sc	300	TCP
6	B2B	Off	Cisco1-sc	Cisco2-sc	300	TCP
7	B2B	On	Cisco1-sc	Cisco2-sc	300	UDP
8	B2B	Off	Cisco1-sc	Cisco2-sc	300	UDP
9	C2C*	On	zinc-ib1	platonium1	300	TCP
10	C2C	Off	zinc-ib1	platonium1	300	TCP
11	C2C	On	zinc-ib2	platonium2	300	UDP
12	C2C	Off	zinc-ib2	platonium2	300	UDP
13	C2C	On	cisco1-sc	cisco3-sc	300	TCP
14	C2C	Off	cisco1-sc	cisco3-sc	300	TCP
15	C2C	On	cisco1-sc	cisco3-sc	300	UDP
16	C2C	Off	cisco1-sc	cisco3-sc	300	UDP

These tests provided information regarding how well the fabric passes Ethernet traffic without being constrained by other factors such as disks.

4.2.2 Measuring vMotion Performance

Measuring vMotion performance of the fabrics gives an indication of how quickly a workload can be moved from one server to another in a real-world situation. Since the process of migrating a VM impacts the performance of the virtual machine during the migration, it is important to reduce the time and impact to VMs as much as possible. This is especially true of critical tier 1 or high demand applications.

The 20 VMs in Table 4-1: Virtual Machine Design of Experiments were transferred from one host to another one at a time and the time to perform the vMotion was recorded. This test was performed twice and the times for the Xsigo VMs were compared to the Cisco UCS VMs by a summation of the times and an average per VM. By following VMware's best practice to isolate vMotion traffic and enable Jumbo frames on the network maximum performance was assured.

4.2.3 Measuring Storage Performance

Storage performance can be very difficult to measure because performance is generally limited by the storage system and not the fabric. Dynamic workloads on a target storage system and the time of day or week can cause results to vary widely when multiple tests are run at different times. It is also prohibitively expensive to acquire a storage system that will fully tax multiple 8 Gb connections. Because of the limitations of rotational media providing the necessary performance for the fabric tests, we used a blade server that has 48 GB of RAM and provided 45 GB of it to VMware as a RAM disk over the Fibre Channel fabric using the SCST software on Linux. VMs used this RAM disk as a standard VMFS volume and because of the limited RAM disk size, these tests were performed with 3 Windows 2008 R2 VMs which have 4

CPUs and 16 GB of RAM with a 10 GB VMDK from RAM disk each. The goal was to prevent resource starvation of the host systems and saturate the 4 Gbps Fibre Channel modules of the two systems.

The VMs ran IOMeter 1.1.0 RC1 and ran the three IO profiles listed in Table 4-3: Storage Testing Profiles three times in a random order on each VM. These tests showed the performance of the fabric for storage traffic under a contention scenario. The tests provided bandwidth, latency, and I/O per second (IOps) which was used to compare the technologies.

Table 4-3: Storage Testing Profiles

Maximum I/O Rate – 512 byte transfer request size, 67% read, 33% write, 100% sequential
Maximum Throughput – 64K transfer request size, 67% read, 33% write, 100% sequential
Database Simulation – 2K transfer request size, 67% read, 33% write, 100% random

4.3 Measuring Usability

In this part of the study a list of daily tasks (Appendix F: Usability Testing Tasks) was created to be performed by new users of the technology. The participants are people who identified themselves as system administrators and were chosen because of their varied experience in system administration; ranging from students to seasoned engineers.

The participants were given the vendor-provided system manual which they could reference at anytime during the exercise. The group was randomly divided into two categories: the first group performed the tasks on Cisco UCS first, then on Xsigo; and the second group performed the tasks on Xsigo first then on Cisco UCS. They were timed on the duration to complete each of the tasks and the number of errors made during the tasks were counted. The

participants were asked to submit free form responses and rate aspects of their experience using the technology.

The data was compiled to see if there was a significant difference in completion times between the two technologies and if the subjects felt that one was easier to use than the other.

4.4 Cost Comparison

Brigham Young University has standardized on server blade chassis for a number of years because of the density and power efficiencies that can be gained from sharing as much infrastructure as possible. The only drawback to blade servers is the limited I/O that a blade can support due to the limited mezzanine card slots. Converged I/O has the ability to remove this constraint and continue to make blade servers a good choice for BYU as server consolidation continues to push more and more virtual machines onto a single server. As a result, the pricing analysis will be based on blade servers in this study, but could be easily extended to rack mount servers.

Four configurations were compared in the cost analysis. The traditional configuration is a baseline of what BYU has been configuring before converged I/O. This consists of six Cisco Ethernet Switches in a server blade enclosure connected to a Cisco 6509 at the End of Row (EoR). There are also two Fibre Channel switches in the blade enclosure with each connected to one upstream SAN switch. These blades have an add in mezz card with four 1Gb Ethernet ports and a second mezz card with two 8Gb FC ports.

The 10G configuration uses four Ethernet pass through modules to allow 10 Gb Ethernet connectivity to the blade. Each blade has four twinax cables to two Cisco 2248PQ FEX top of rack (ToR) units. The Cisco 2248PQ FEX are then connected to a Cisco N6004 at the EoR which

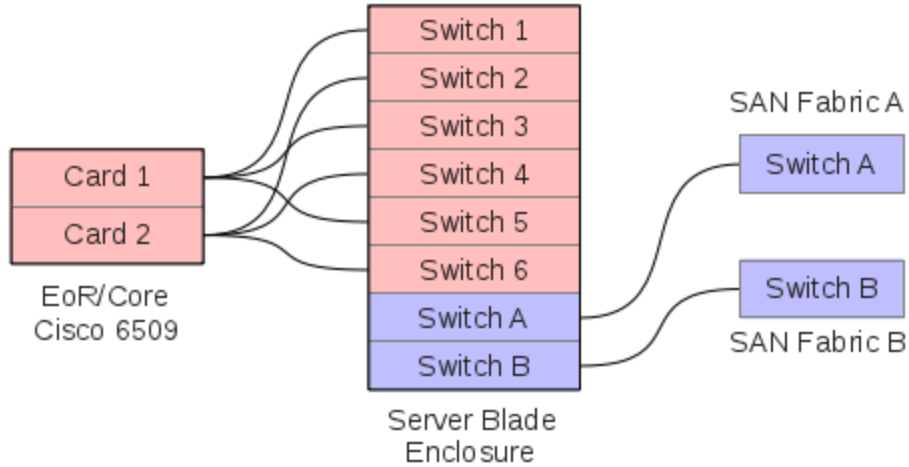


Figure 4-1: Traditional Blade Configuration

is then connected to a Cisco 6509 at the core. The Fibre Channel configuration is similar to the traditional configuration. Each blade has a mezzanine card with two 10Gb Ethernet ports and a second mezz card with two 8Gb FC ports.

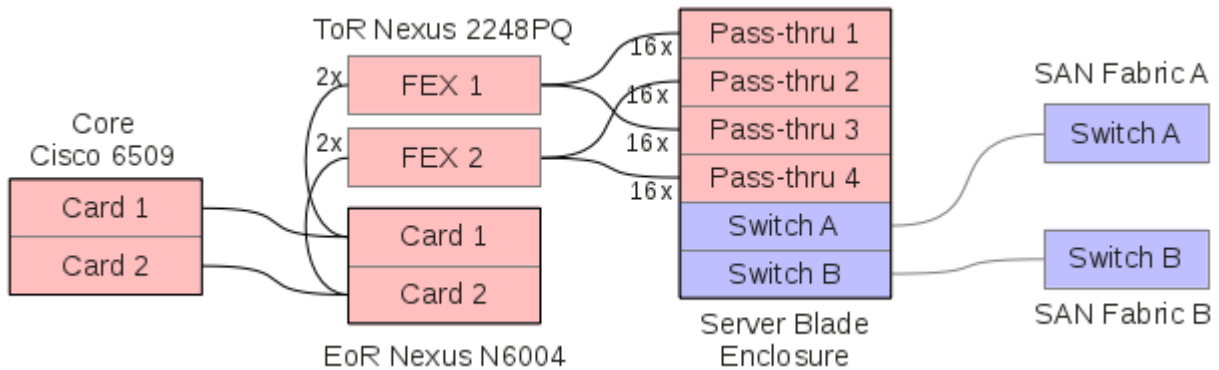


Figure 4-2: 10 Gb Blade Configuration

As of this writing there is now an option to replace the server blade pass through modules with Cisco B22HP modules which are also FEX units like the 2248PQ and can eliminate many cables and rack space for the 2248PQ. The Cisco N6509 can also do FCoE from the HP blade

CNAs which is another option to reduce the Fibre Channel footprint when used in conjunction with the Cisco B22HP FEX. The primary analysis will be around the configuration that was available at the time.

The Cisco UCS configuration uses two blade enclosures to standardize on a 16 blade configuration for easier comparison. Each blade enclosure has two FEX modules and each FEX has four twinax cables to a UCS Fabric Interconnect (FI) at the EoR or middle of row (MoR). The UCS FI then have a 10 Gb Ethernet to a core Cisco 6509 and two connections each to FC switches. Each blade is fitted with only a two port 10 Gb CNA.

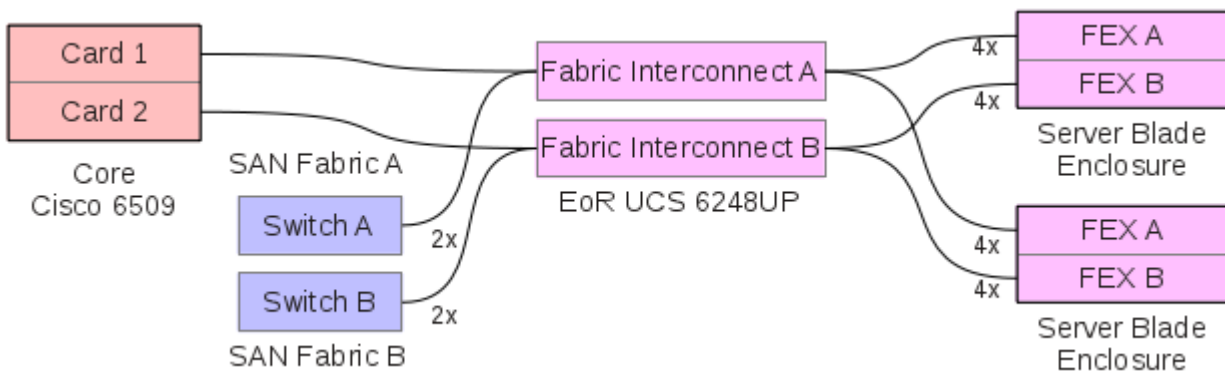


Figure 4-3: Cisco UCS Blade Configuration

The Xsigo configuration has two blade enclosure QDR Infiniband switches and each switch has two connections to an EoR Infiniband switch. The EoR Infiniband switches are cross connected to two directors. Each director has one 10 Gb Ethernet and two FC connections to core switches. Each blade only has one two port QDR HCA.

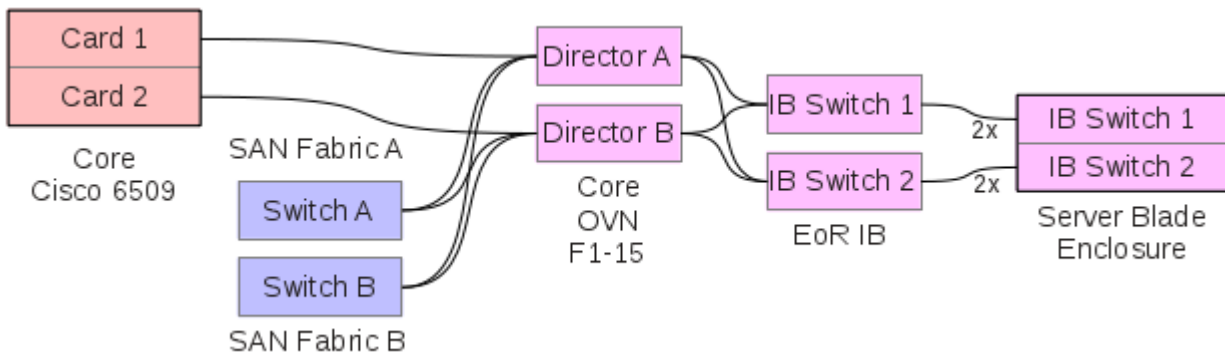


Figure 4-4: Xsigo Blade Configuration

These configurations were chosen as they allow the most growth by minimizing cabling to core switches and maximizing cost effective ToR and EoR equipment. The analysis uses a maximum configuration of the EoR switches, Cisco UCS FI and Xsigo Director then divides the total cost by the number of down stream ports to the blade enclosure. This normalizes the price over different pricing models between vendors.

4.4.1 CapEx Cost Analysis

Since vendors generally apply discounts of varying degrees to enterprise customers list prices were used which can be easily obtained from any vendor. By comparing the costs of different converged I/O technologies and traditional I/O technologies it can be determined if there is a cost savings and by how much. By applying a specific vendor discount percentage that an enterprise may receive they are able to compute a rough comparison of their own based on the data in this study. Since only the costs of the communication technologies is desired, the cost of the servers were removed from the analysis as much as possible. The capital expense was

calculated for the PCI-e or mezzanine cards, switches, cables, GBICs, cable installation and licensing for the equipment.

4.4.2 OpEx Cost Analysis

The operational expense analysis consists of power and cooling costs. Power utilization was computed based on vendor documentation, which depending on the vendor documents peak power usage or typical power usage or both, and amortized over all the ports of the device. This allowed a projection of the energy usage over a continuous scale. Power is expressed in watts which an enterprise can calculate their estimated costs by using their local energy rates. BYU has a highly efficient data center with a PUE of 1.37 (Zeeman, 2014), but because of the variability of efficiencies of different data centers, an average data center power usage effectiveness (PUE) rate of 2.0 was used for the calculations. (Stansberry, 2013) Individual PUEs for an enterprise can be substituted for a more accurate power consumption.

4.4.3 Other Cost Considerations

By projecting these costs over any number of servers, it is easy to see if one solution has a cost advantage. It should also be noted that Brigham Young University employs a two fault domain configuration in their data center. Because of these separate fault domains, equipment is usually purchased in pairs so that one can be located in Site A and the other in Site B. The cost analysis includes the appropriate costs to provide the equipment in such a fault domain configuration.

5 RESULTS AND DATA ANALYSIS

5.1 Ethernet Performance Analysis

In the attempt to saturate the capacity of the equipment, there were a number of tests which failed to start or did not provide useful data. Even though these few erroneous results were thrown out there was still enough data to allow for a reliable statistical analysis. The VMs CN1-CN10 sent data to CN11-C20 so the numbers for CN1-CN10 match the numbers for CN11-CN20 and we only need to use half the data for our analysis. Using the TCP tests, the overall average bandwidth per VM was generally about 40% higher for Xsigo compared to the Cisco tests.

Table 5-1: Testing Label Definition

WX-Y2Z		
W	T	TCP Test
	U	UDP Test
X	J	Jumbo frames
	S	Standard frames
Y2Z	B2B	Blade to Blade
	C2C	Chassis to Chassis

Figure 5-1: TCP Bandwidth Results shows the results from the TCP testing. The chart shows the difference between Xsigo and Cisco in data transfers between blades in the same

chassis and blades in separate chassis. It also shows a comparison between standard 1500 byte frames and 9000 byte jumbo frames.

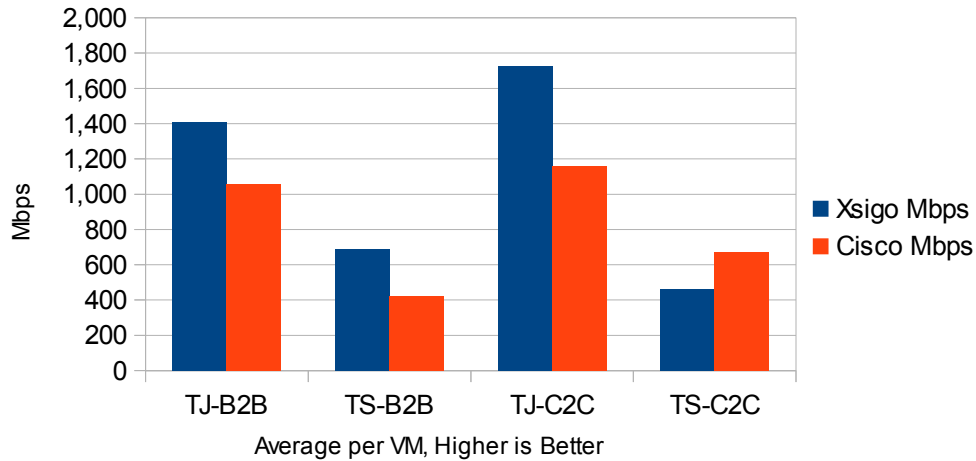


Figure 5-1: TCP Bandwidth Results

Even though the average bandwidth of the VMs was higher for Xsigo, when performing a statistical analysis on the data from each VM there was no statistical significance with a 95% confidence interval; the null hypothesis was not rejected. Even though there is no statistical significance in this test, a 40% increase is about 400 total Mb/s or 40 Mb/s per VM which may help some services not fail completely during times of saturation.

This gives Xsigo a slight advantage over Cisco in the TCP tests. An area of additional study could include the effects of Infiniband's guaranteed delivery mechanism and the effects it has on the TCP window. It is possible that since Infiniband ensures that every packet is delivered to its destination, the TCP recovery mechanism does not go into effect. In theory this could prevent the TCP window from being cut in half which would increase latency and impact performance until the TCP window increases in size again. The Fibre Channel over Ethernet

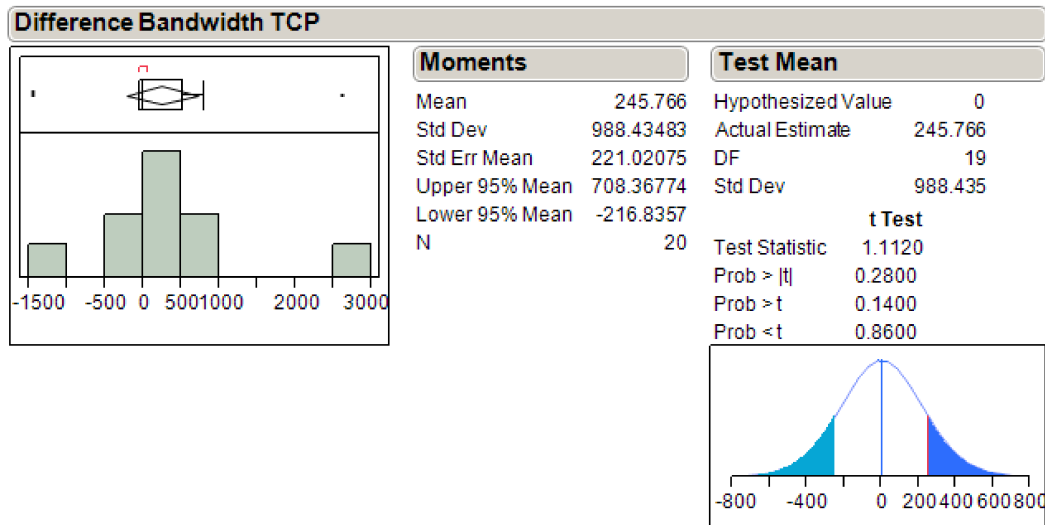


Figure 5-2: TCP Bandwidth Distribution

protocol should also provide guaranteed delivery so it is uncertain why the discrepancy exists between the implementations. It is possible that guaranteed delivery for FCoE only applies to storage traffic and Ethernet traffic is still best effort delivery.

Both solutions had 20 Gb/s of Ethernet gateway (North-South) capacity and both performed close to the theoretical limit using Jumbo frames. The lower performance of standard frames could be due to all the VMs running on two hosts and not enough CPU capacity to perform the build up and tear down of the TCP packets but this too could use some additional experimentation.

Cisco UCS performed better in almost all the UDP bandwidth tests by about 20% which at first was surprising given how much Xsigo outperformed Cisco in the TCP tests.

Again the null hypothesis was not rejected when a statistical analysis was performed on the test results meaning there was no statistical difference between them.

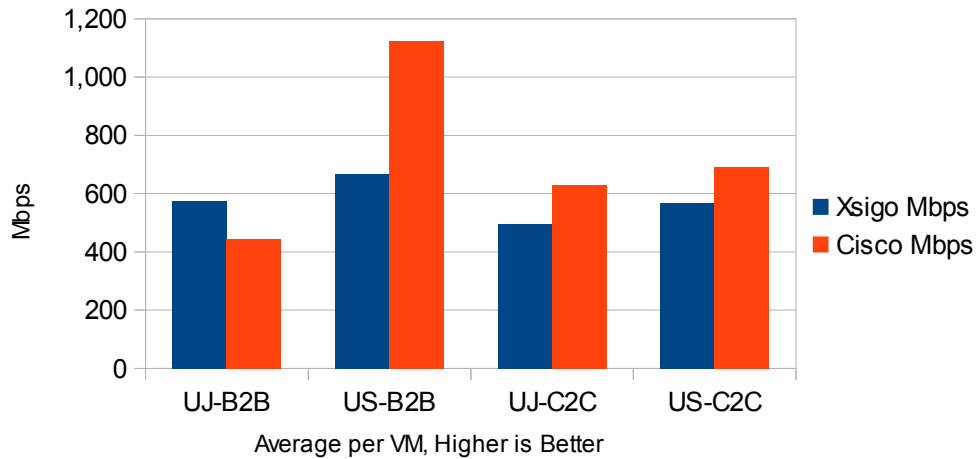


Figure 5-3: UDP Bandwidth Results

Cisco UCS also performed much better in the UDP jitter tests than did Xsigo which again was pretty surprising given that Infiniband configures routes for each connection from vNIC to gateway card. One would expect a pretty consistent packet delay for the traffic, but the guaranteed delivery of Infiniband possibly causes packets that would otherwise be dropped to be just delayed. Cisco UCS on the other hand probably understands that the packets are UDP and

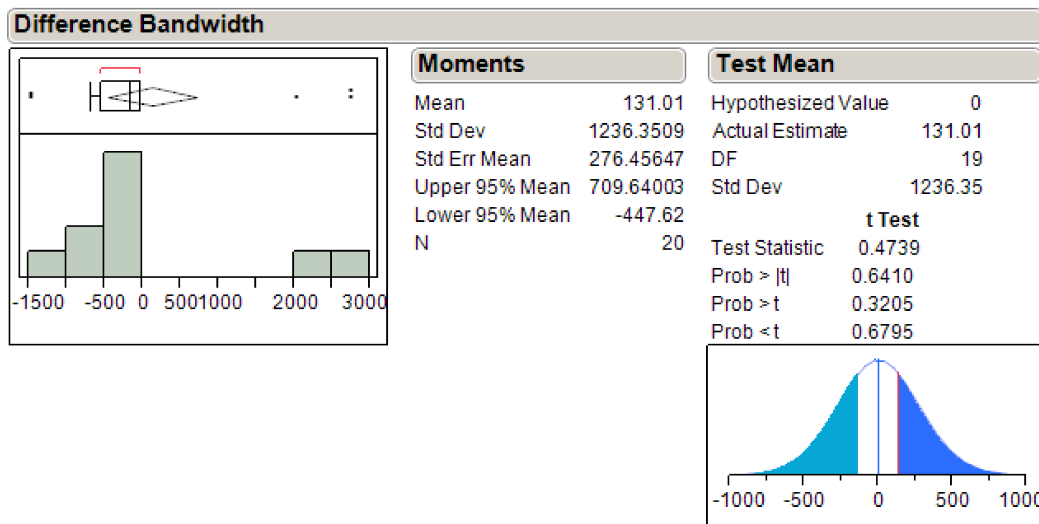


Figure 5-4: UDP Bandwidth Distribution

does not try to guarantee delivery of them if there is congestion on a network segment. This is a reasonable explanation but could use some additional experimentation to really understand what is going on in this test.

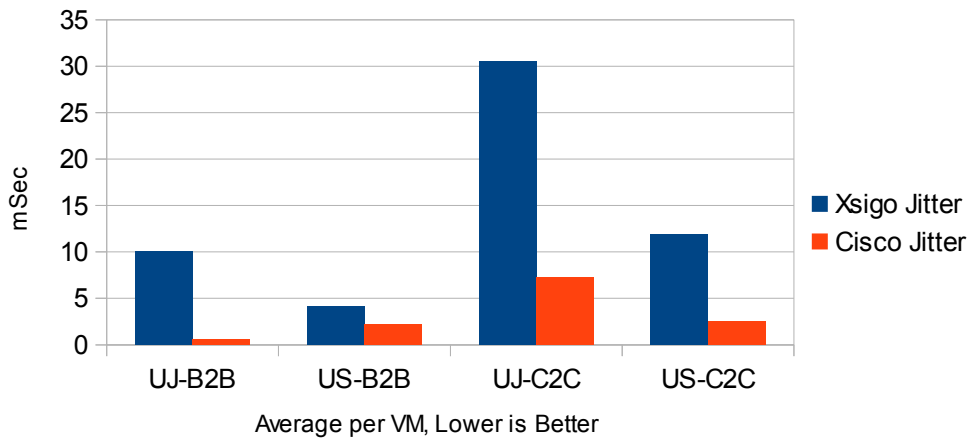


Figure 5-5: UDP Jitter Results

The null hypothesis was not rejected in the one-sided t-test analysis again meaning that there was no difference between them. Since it was decided to use a confirmatory test to build a stronger statistical test, it would be improper statistical form to build the hypothesis that Cisco UCS is better based on the results of the data. Proper statistical form would require developing the counter alternative hypothesis then building a new test, perform the test and analyze the results, which was not done for this study.

The UDP datagram loss tests again at first were surprising in that Xsigo performed so poorly compared to Cisco UCS. Xsigo lost almost 75% of all packets while Cisco only lost about 40%. After working through the data it is clear how the values from the Ethernet tests correspond and are accurate given what we know about the working of Infiniband and Ethernet.

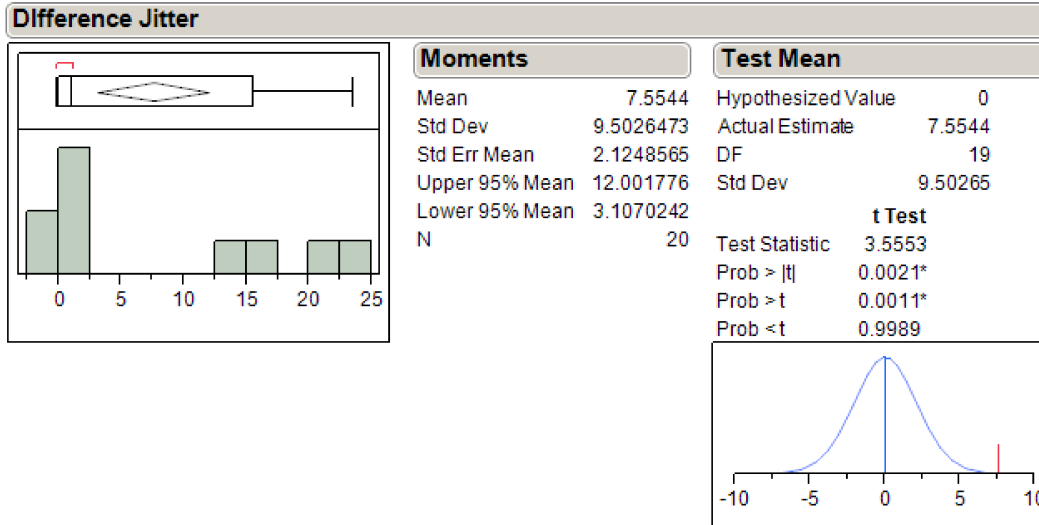


Figure 5-6: UDP Jitter Distribution

The likely cause of the poorer performance with Xsigo in this contention situation is that it is trying to guarantee delivery of the UDP packets. Once the Infiniband fabric and adapters become backlogged, the packets start getting delayed or dropped at the adapter because there are no more buffers left. This appears to explain the lower bandwidth, the higher jitter and higher packet loss.

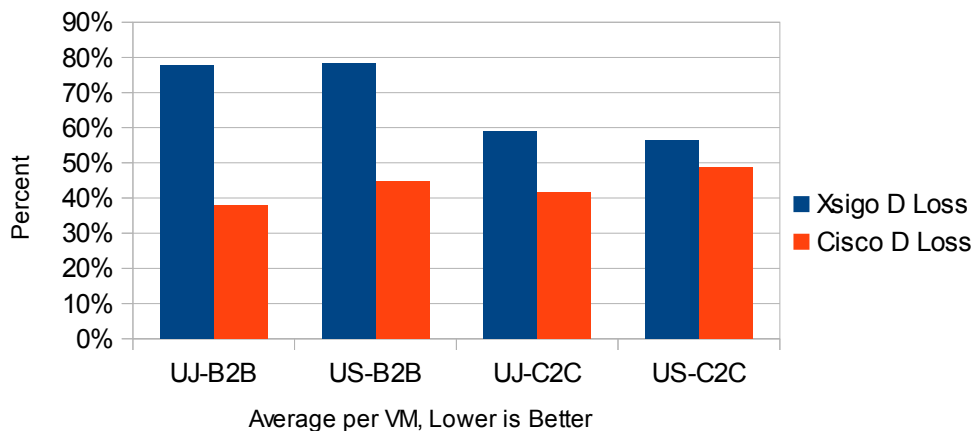


Figure 5-7: UDP Datagram Loss Results

Cisco UCS on the other hand seems to understand that the UDP packets do not require guaranteed delivery and can be dropped when and where the congestion is happening. This gives Cisco the lower jitter, higher bandwidth and the lower packet loss.

More experimentation should be conducted in this area and if this hypothesis is validated, then it could help the Infiniband solutions handle such contention situations better. Infiniband does have a non-guaranteed packet delivery mechanism called unconnected mode and if it is not being leveraged, it may be advantageous to get the host drivers to be able to distinguish between TCP and UDP traffic and use the appropriate transport mechanism in the Infiniband layer.

5.2 vMotion Performance Analysis

The ability to vMotion virtual machines quickly between hosts becomes increasingly important as the hosts that run the VMs become larger. For instance, when only running a handful of VMs on a host that has 20 GB of RAM allocated, the workloads can be moved relatively quickly even over 1Gb Ethernet connections. However when you have 100 or more VMs that add up to 256 GB of RAM or more, it could take hours to move the workloads to other hosts using 1 Gb connections. When a host has an imminent failure, the reduction in vMotion times can mean the difference between an outage and uninterrupted service.

Historically one of the main deterrents of blade servers is the lack of I/O bandwidth that can be presented to the blade servers due the limited number of card slots available. These converged architectures have the ability to remove that hurdle and present as much or more I/O bandwidth as a rack mount server. This high bandwidth ability can enable much lower vMotion times helping avoid outages and reduce maintenance times.

Xsigo performed vMotions faster on average taking 5 seconds longer in some cases and 14 seconds less in other cases.

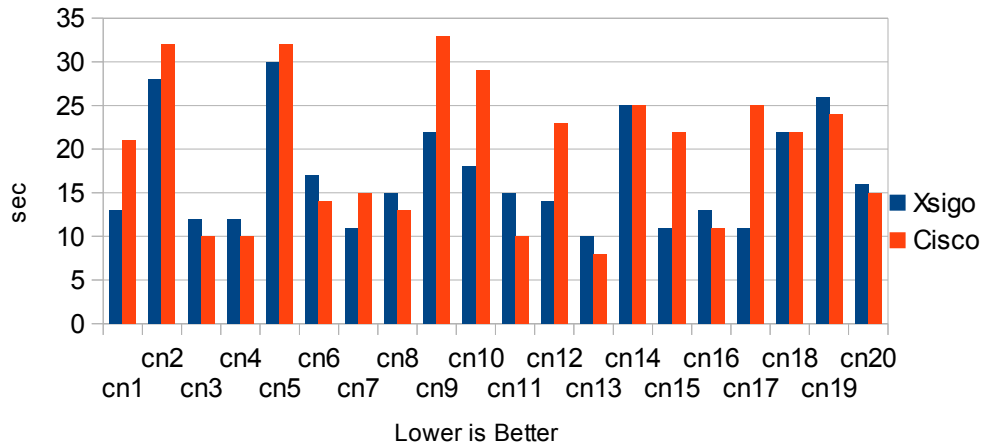


Figure 5-8: VM vMotion Time Results

Overall Xsigo took a total of 53 seconds less to complete all the vMotions than Cisco UCS. This represents a 13% reduction in vMotion time as compared to UCS which with such a small test group doesn't amount to much, but could add up to a significant amount of time with very large hosts given the linear scaling nature of vMotion.

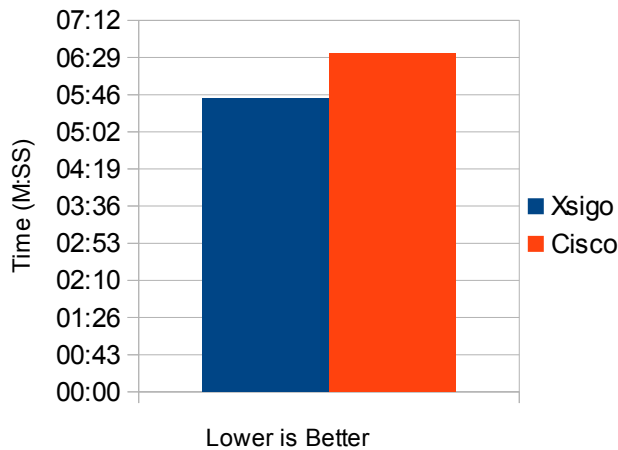


Figure 5-9: Total vMotion Time

The null hypothesis was rejected by a one-sided t-test with a 95% confidence interval. Given the results of the TCP test and that vMotion uses TCP, these results are in line with the previous test.

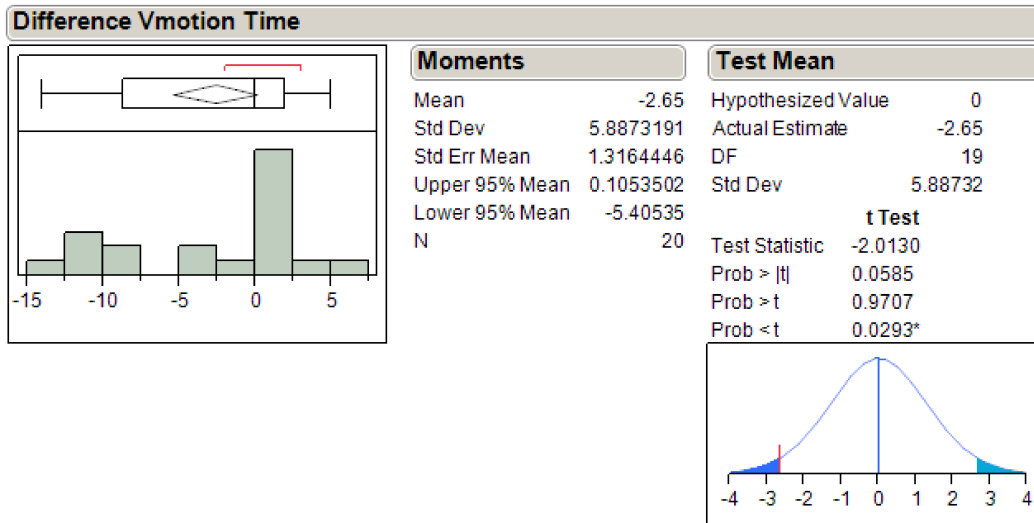


Figure 5-10: vMotion Times Distribution

It is expected that with recent developments of Xsigo's Server Fabric that there is a much larger advantage for vMotion on the Infiniband fabric. Xsigo's Server Fabric leverages native Infiniband communications between Infiniband hosts so that traffic does not have to pass through the Xsigo directors and 10 Gb Ethernet modules. This is termed East-West traffic as it never leaves the Infiniband fabric. With Xsigo Server Fabric, the traffic can be routed in the shortest possible path and at native Infiniband speeds currently 56 Gb/s per link. This along with the ability to support 64K MTUs can significantly reduce the vMotion times of very large virtual machines. Small VMs are unlikely to see much improvement as most of the time is currently spent setting up the VM for vMotion, syncing the final state and handing over the running of the

VM. The amount of time spent transferring data is already relatively small for these smaller VMs.

The new technologies introduced by Xsigo Server Fabric warrants experimentation to see just how much of an advantage Infiniband can provide to reducing the vMotion times for Large VMs and the effects it has on small VM vMotions.

5.3 Storage Performance Analysis

Storage performance was the most difficult test run on the equipment. Qlogic specifies a maximum of 200,000 IOps on the Fibre Channel adapter that was used in the Linux RAM disk host. Based on the results of the testing it appears that the adapter limit was reached and prevented the saturation of the UCS and Xsigo fabrics as intended.

In the IOps test both solutions did very well and they performed identically. The three virtual machines generated 130,000 IOps which is more than many storage systems can handle. Because both systems performed identically it indicates that limitations were encountered on the physical end with the Linux RAM disk host. It would have been better to see more separation between the results even if it was only between the database and max I/O profiles.

In the data transfer testing, there was much more separation between the classes of tests, but again both solutions performed nearly identically. The database test profile had a 2 KB request size and the max I/O profile had a 512 B request size which puts the storage transfer results right in line with the IOps test. Because the database profile request size was four times larger than the max I/O profile, the bandwidth is four times as large. Even though the IOps were the same between these two profiles, the bandwidth was not and suggests that there was an artificial cap of IOps on the Linux RAM disk adapter. The max throughput profile, with a 64K

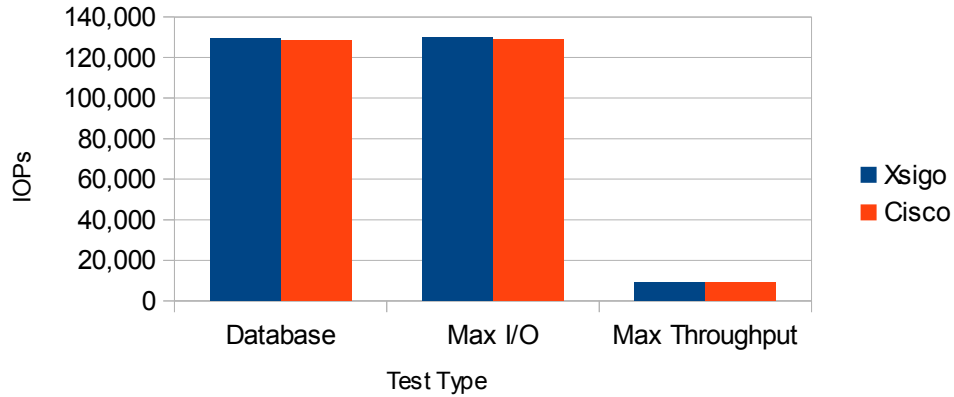


Figure 5-11: Storage I/O Performance

request size, tells a different story where it may have been limited by the bandwidth of the Linux Fibre Channel adapter and is verified by a drop in IOps.

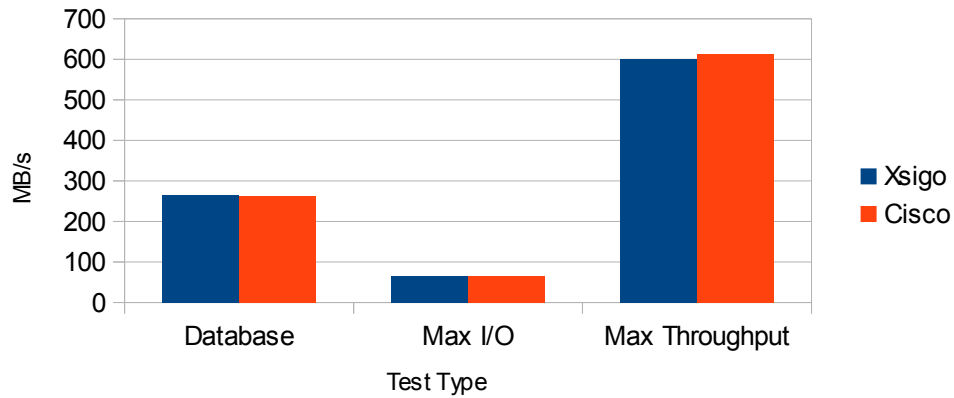


Figure 5-12: Storage Transfer Performance

The latency results indicate that there was an issue with the Linux RAM disk host and not with the fabrics. Both systems performed equally well in the response times of the database and max I/O profiles; the fabric did not add significant latency to the requests. It is difficult to say if

the extremely large latency in the max throughput is due to the larger packet size or if congestion in some part of the system is the driving factor.

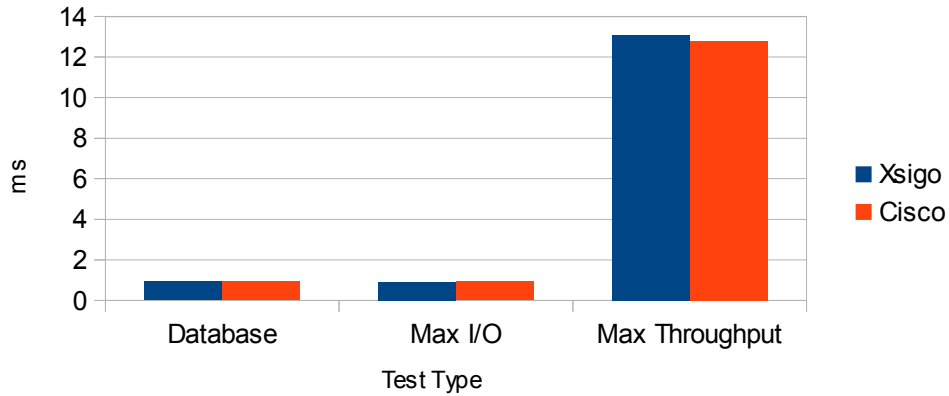


Figure 5-13: Storage Latency Performance

The storage tests were not conclusive other than both systems are capable of nearly reaching theoretical physical limits and should perform similarly as non-virtualized Fibre Channel systems. Additional areas of study would include multiple physical Linux hosts to remove the limitations seen in this testing and to see what happens when the fabric Fibre Channel gateways are saturated. This response would be helpful in determining the safe overcommitment rate of the Fibre Channel gateways and if one solution handles such situations more gracefully. As high performance solid state disk arrays are starting to become more commonplace in the data center, high IOPs can be an important factor to consider especially when scaling out these fabric systems. When many hosts on the fabric are performing I/O they could overwhelm the gateways which may have been artificially limited in the past by the performance of rotational disks in traditional storage arrays.

5.4 Usability Analysis

It is difficult for a system to be very useful if it is cumbersome to use. The usability test determined how easy each system was to learn and use. Eleven subjects who have not used converged communication systems like UCS or Xsigo participated in our testing which asked them to perform four tasks (Appendix F: Usability Testing Tasks) on each system. The operating manual as well as contextual help and the Internet was made available to the participants. Their times were recorded and summed together along with the number of errors in their tasks. They were asked to perform the tasks a second time and the results were recorded and summed. The participants then performed the outlined tasks on the second system.

The participants on average completed the first set of tasks on the UCS system 20% faster than the Xsigo system. In the second attempt of the tasks on the same system the time was dramatically reduced in both systems to less than 50% of the time of the first attempt. UCS was still about 15% faster than Xsigo after the second attempt.

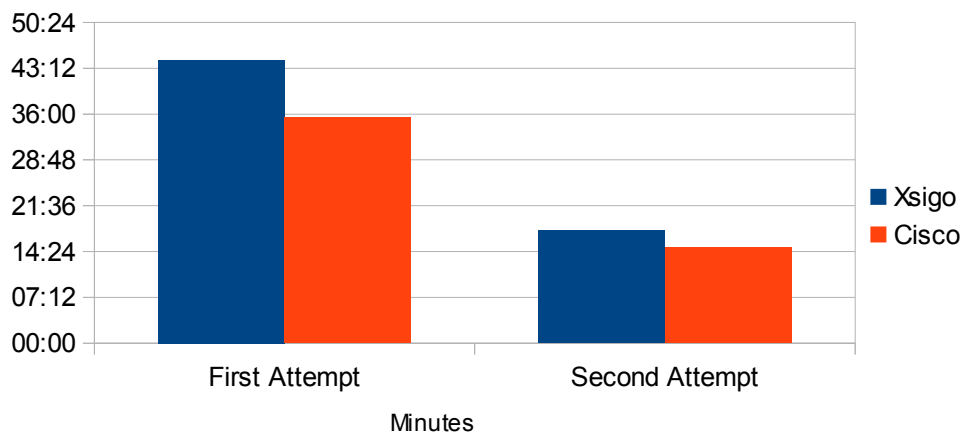


Figure 5-14: Task Completion Results

The number of errors made during the testing was quite high on the first attempt at the tasks, but almost completely disappeared on the second attempt. This shows that the system has a fairly low learning curve and once the concepts are understood, the tasks can be completed faster and with less errors.

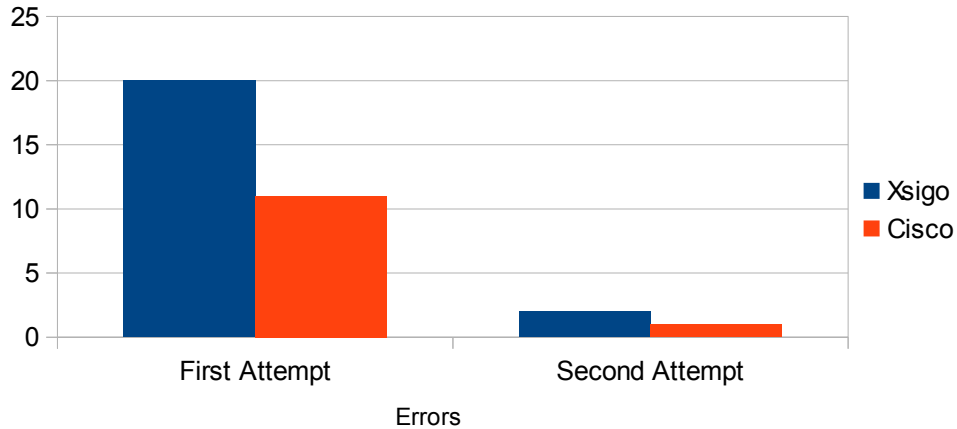


Figure 5-15: Usability Errors for Given Tasks

It was interesting to note that whether the participant completed the UCS tasks first or the Xsigo tasks first, it had no effect on the first attempt of the other system. Xsigo consistently took longer and the times between participants were fairly consistent. This tells us that there is enough differences in terminology and implementation that knowledge could not be easily transferred between systems.

One of the most counter intuitive results from this test is the participants' correlation between likability and ease of use. Even though Cisco had twice the number of positive responses from a likability stance, Xsigo lead by ten points on the Likert scale response for ease of use. It is difficult to say what exactly caused this discrepancy but some of the free form responses suggests that some people were familiar with the Cisco interface and some felt more

comfortable in a more traditional application with right-click abilities than a web application. Some of the notes for Xsigo tended towards the awkwardness that can be found in many web pages and that it needed more polishing. In my experience the UCS interface required more clicks to perform similar tasks than the Xsigo interface and that could have contributed to a better ease of use score.

Even though there is a learning curve required for implementing communication convergence, it does not appear to be so difficult as to be a huge barrier. Additional experiments could be focused in this area on some of the more complicated tasks that were not addressed in this study. The ability to have a significant amount of time lapse between repeating the tasks could provide more data with regards to understanding the technology rather than the participant's ability to regurgitate a procedure. Additional study with regards to impact of man hours for certain tasks could provide a cost savings projection model that could help determine the impact of convergence from a human resource perspective.

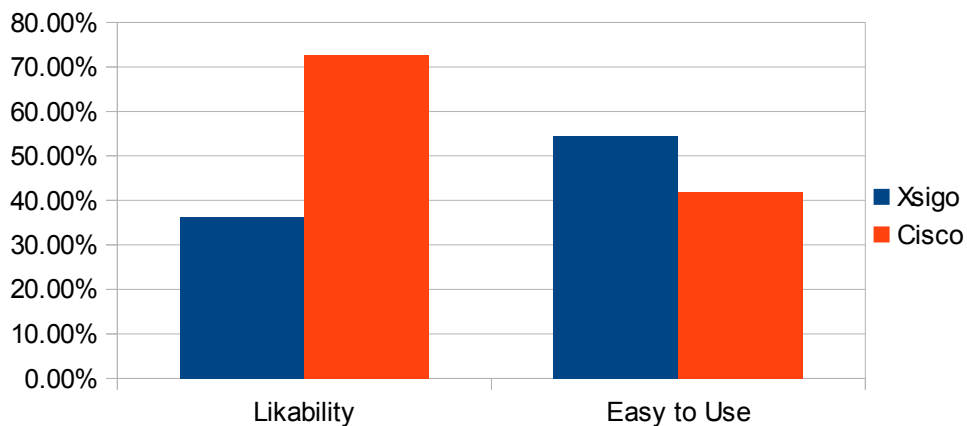


Figure 5-16: User Perception of Administrative Console

5.5 Costs Analysis

The cost analysis is probably one of the trickiest components of the study due to the variance of pricing and bundling of components, etc. Although the pricing used for the analysis is list pricing and no one usually pays list price, it should still provide useful comparisons between technologies that are still relevant. This will give values that should be pretty relative over time as pricing fluctuates.

The first part of the analysis will look at available bandwidth to the chassis and blades and compare them to each other. An analysis of cable counts between the solutions is provided given the configuration described earlier. These figures then allow a computation of guaranteed bandwidth per blade under extreme contention situations. Cable redundancy is analyzed by determining the amount of bandwidth each cable contributes to the whole. The costs associated with the equipment, cabling and necessary components to connect everything is compared between each solution. Finally, power consumption is computed to give a comparison of long term expense or each solution. Each environment had different priorities and this will allow one to determine whether bandwidth, cable management and/or total cost is most important in their environment.

Figure 5-17: Chassis Bandwidth Shows how much bandwidth is available to the chassis as a whole based on the configurations mentioned in 4.4 Cost Comparison. The 10Gb configuration has the highest bandwidth, but at a cost of many more cables compared to any of the other configurations. Both the Cisco and Xsigo configurations provide more bandwidth to the blade enclosure than the traditional configuration.

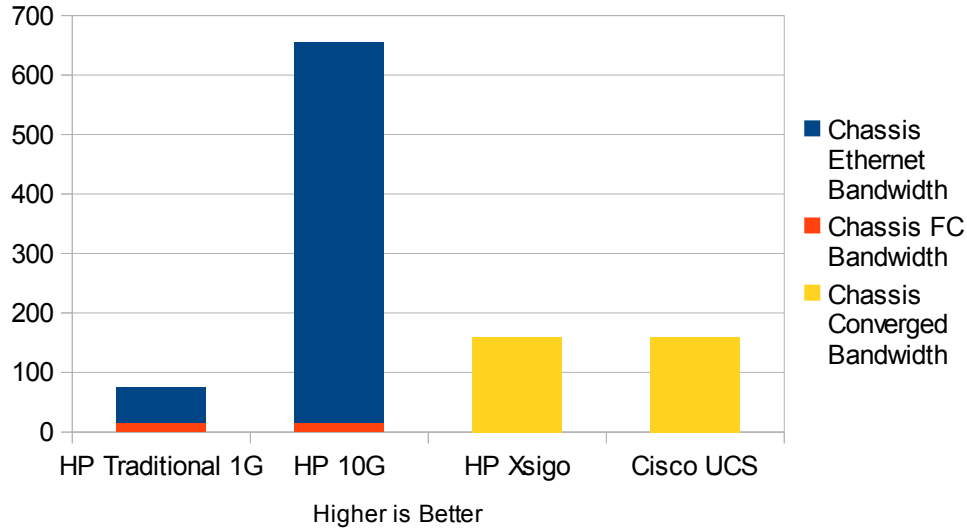


Figure 5-17: Chassis Bandwidth

Figure 5-18: Blade Bandwidth Shows the maximum amount of bandwidth that is available to each blade if there is no contention for the resources from other blades in the chassis. This is determined by adding up the bandwidth of all the ports within a single blade. It is interesting to note that even though the traditional configuration only has 1 Gb Ethernet ports, it still has more bandwidth than Cisco UCS. However, UCS does provide much less management overhead within its configuration by reducing the number of adapters and types of cables to connect the system. The Xsigo configuration provided the greatest amount of bandwidth to a single blade with two 40 Gb ports.

Figure 5-19: Cable Count Show the number of cables that each chassis is configured with. Since HP blade enclosures have 16 blades per chassis and Cisco UCS has 8, the Cisco numbers have been doubled to normalize them to the HP figures. Due to the pass through modules of the 10 Gb configuration, there is a one to one cable to blade Ethernet port ratio. This should translate into higher costs of cables, ports, management and power. In order to attempt to

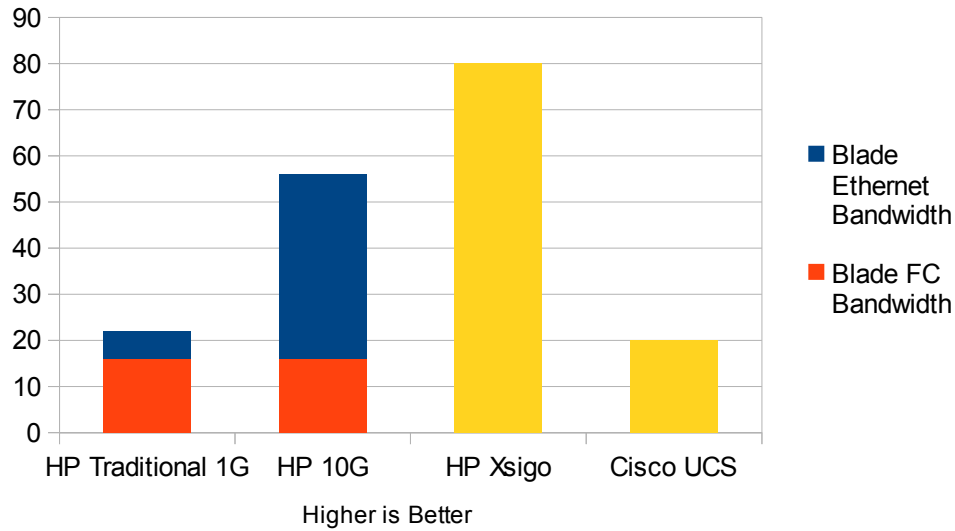


Figure 5-18: Blade Bandwidth

provide similar bandwidth to the blade chassis between Xsigo and Cisco UCS, many more cables had to be used in the UCS configuration pushing it past the number of cables in the traditional configuration. The Xsigo configuration could have used only one cable per chassis Infiniband switch allowing us to reduce the corresponding number of UCS cables, but Xsigo would not have had cable redundancy to the EoR switch. The cable redundancy was important enough to warrant configuring the extra cables in the UCS configuration.

Figure 5-20: Guaranteed Bandwidth to Blade shows the amount of guaranteed bandwidth to each blade in a complete I/O saturation scenario where every blade in the enclosure is trying to use all the bandwidth it can. The 10 Gb Ethernet configuration has the highest bandwidth guaranteed per blade due to the one to one subscription ratio of the Ethernet links, but comes at a price of additional cable management. The traditional configuration had the lowest level of guaranteed bandwidth, but given that the Ethernet links are 1 Gb instead of 10 Gb, it is only about four to one over subscribed which is not as bad as Xsigo which is eight to one. Cisco UCS

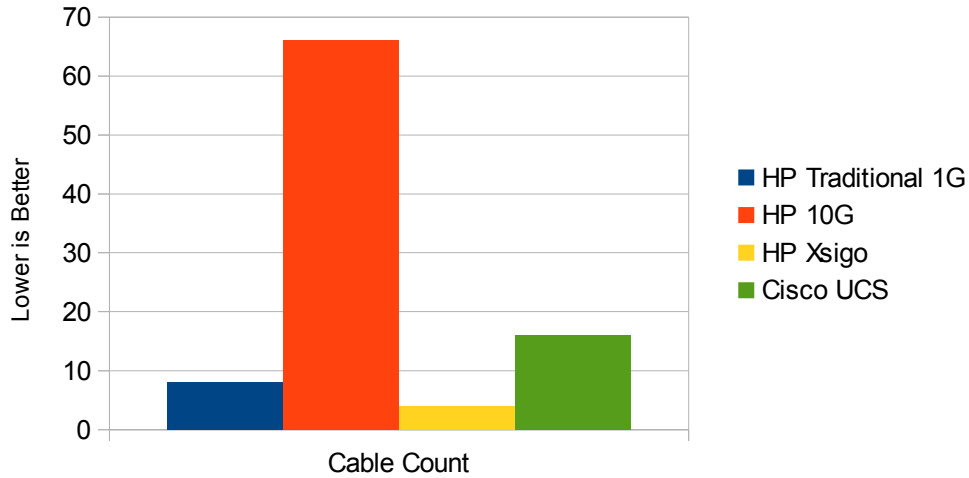


Figure 5-19: Cable Count

only had a two to one over subscription and provides the same guaranteed bandwidth as Xsigo, but with four times the cables.

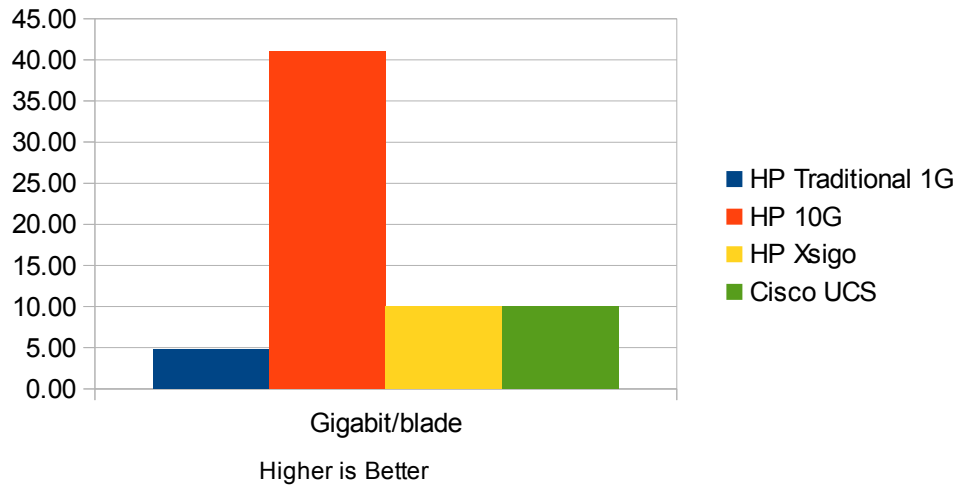


Figure 5-20: Guaranteed Bandwidth to Blade

Figure 5-21: Percent Bandwidth per Cable shows the amount of contribution that each cable provides to the chassis bandwidth. Although having many cables improves redundancy and

bandwidth, it also has a higher management overhead. The Xsigo configuration had the highest efficiency with each cable providing 25% of the chassis bandwidth. UCS and the traditional configurations were in the middle with UCS needing more cables, but also providing more overall bandwidth and redundancy.

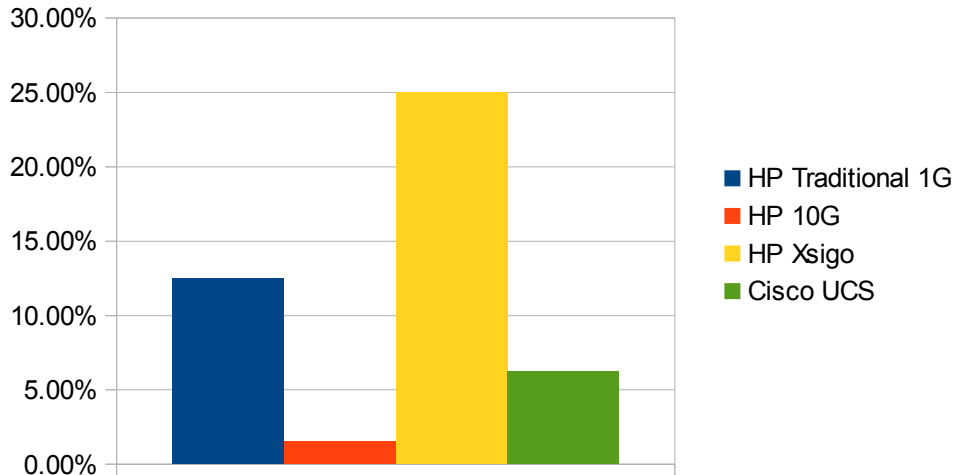


Figure 5-21: Percent Bandwidth per Cable

The results of this exploration show that there are several trade-offs between the varying technologies. Solutions that have high bandwidth and resiliency also have higher management costs; high bandwidth and low cable management solutions have a trade-off of lower redundancy between components. These trade-offs need to be considered when making a converged infrastructure decision.

For the cost analysis portion, the switch costs have been amortized over the complete number of ports available. Because of this amortization, smaller installations will have a higher actual costs compared to these figures since a number of ports will have to be purchased initially and left unused.

In Figure 5-22: I/O Costs per 16 Blades the cost of the I/O for each configuration is given in list prices. Both converged network solutions offered a substantial reduction in I/O costs per 16 blades. A 50% or more cost savings from a CapEx prospective can be realized by implementing Xsigo or Cisco UCS.

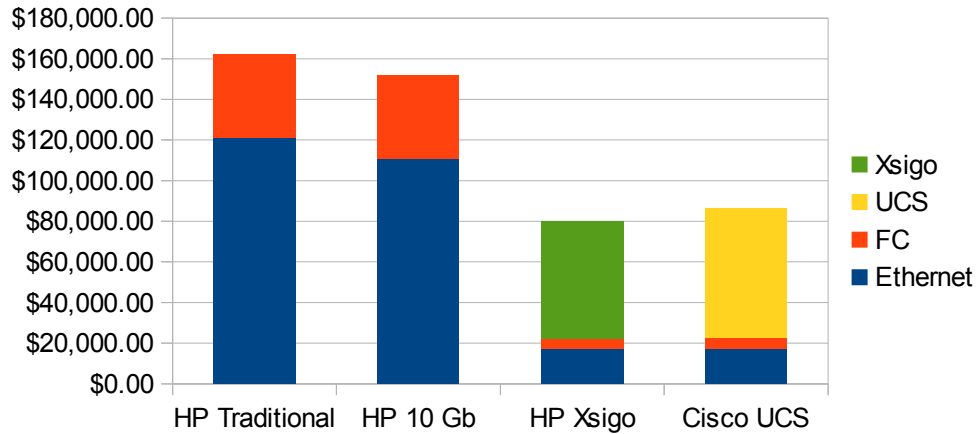


Figure 5-22: I/O Costs per 16 Blades

Cisco UCS provides a 12 port base license with the UCS FI and additional ports have to be licensed for an additional cost. This allows a UCS configuration to be started with a small upfront cost and deferring some of the cost of growing the system to later years. However, the cost of the port licensing is so expensive that it make sense to buy a new set of FIs once the first 12 ports are occupied and leave the rest vacant. This somewhat detracts from the vision of a single converged fabric. Xsigo does not license any of the ports and it promotes building a large single fabric, but the full cost of some of the components such as Infiniband switches have to be paid upfront even if only a fraction of the ports are to be used initially.

Converged I/O provides power savings due to the reduced number of components in the system. Xsigo provided the lowest power consumption of the solutions and Cisco UCS configuration was surprisingly similar to the traditional configuration. The converged adapters in the Cisco UCS configuration have a high power requirement as well as the fabric interconnects which keeps it on par with the traditional configuration for 16 blades. The 10 Gb configuration uses only slightly more power compared to the traditional configuration by leveraging the more power efficient Nexus switches and FEXs instead of the Catalyst switches.

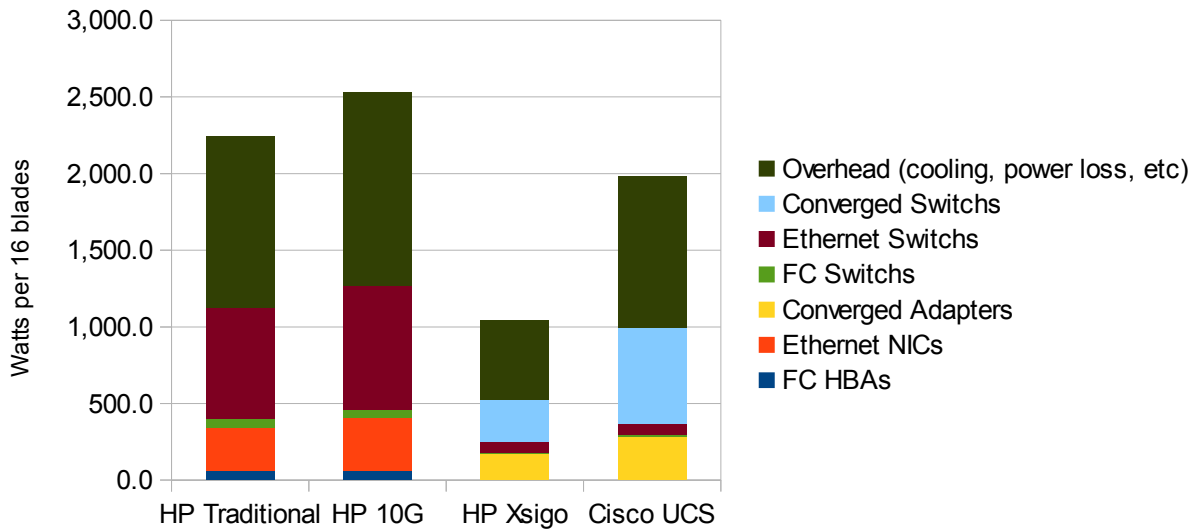


Figure 5-23: I/O Power Consumption per 16 Blades

The overhead energy is calculated based on an average data center power usage effectiveness (PUE) of 2.0. The overhead power figure includes the power needed to cool the equipment, power loss in wiring, distribution units, uninterruptible power supplies (UPS), lighting and other power consumption needed to run the data center as a whole. Depending on the specific PUE of the data center in which the equipment will be placed the total power

consumption may be more or less. These values are intended to give an idea of the possible power consumption and more experimentation is needed to look at other converged solutions and configurations.

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

An analysis framework was developed to facilitate the creation of unbiased testing techniques of converged I/O systems. By performing a case study of Xsigo and Cisco UCS in a BYU data center environment the methodology was validated to be effective in providing a complete analysis of the two systems which showed that converged I/O would increase performance, reduce costs and increase flexibility.

Communication tests were constructed using statistical design of experiment techniques which allowed for an unbiased comparison of technical abilities of the two systems. A usability test measured the accuracy and speed which test subjects completed a list of tasks. Feedback was solicited from the subjects regarding their experience using the technologies which helped explain the contradictory results from the tests. A cost analysis was performed on the converged infrastructures to compare them to a traditional configuration taking into account hardware, cabling, and power aspects to show potential CapEx and OpEx savings.

Both Xsigo and Cisco USC performed well through the tests achieving performance that is similar to that expected from traditional communication configurations. There were some interesting results when the systems were fully congested and solutions handled them differently. Xsigo performed well under the TCP and vMotion tests while Cisco did very well in the UDP

test set. A test with a combination of TCP and UDP traffic should have been conducted to see how the interaction of traffic impacts each protocol as the system becomes congested. Overall, neither solution has any major deficiencies which would be considered detrimental preventing installation in a production data center.

In the storage performance tests, limitations were encountered in the target storage test system. Although it was correctly determined that a physical media target would prove too variable and not allow full congestion of the converged infrastructure, it was unexpected that the host bus adapter of the RAM disk Linux machine would be the bottleneck. From the data that was acquired, both systems once again performed very well with little variation between them. Despite the RAM disk machine limitation, the experiment showed that neither system has greater overhead or latency than a traditional configuration.

Results showed that the usability of the systems was not so difficult as to render the tasks impossible. Speed and accuracy improved in the tasks as they were performed multiple times. It was interesting to note that even though both systems are converged infrastructure, the nomenclature and procedures of one system did not directly carry over to the other system. The first time a task on a new converged product was completed, it took the same amount of time regardless if the administrator had never used a converged product before or if they had previously used the other technology. It is reasonable to expect that there would be some reduction in time on the second product. The most unexpected result from the usability study was the orthogonal result of likability and ease of use. Even though most people liked the Cisco interface, they found the Xsigo interface easier to use.

New technology is not very useful if it does not provide a significant added value to the organization. Through the cost analysis, converged infrastructure will provide a cost saving over traditional infrastructure. By reducing the number of devices and cables, the capital costs for equipment is reduced. This reduction also translates into less power usage and fewer cables to maintain which reduces operational expenses as well. In addition to the monetary savings, converged infrastructure can provide additional bandwidth to servers. These bandwidth gains can remove one of the largest obstacles of deploying blade servers in an environment. This becomes more important as servers increase in density and processing power.

The analysis framework of converged I/O now gives system administrators a tool to use for comparing several important aspects of converged I/O systems.

6.2 BYU's Experience with Xsigo (OVN)

Based on the results of this study, BYU chose to implement Xsigo as part of its initiative to innovate the data center. Since Xsigo was deployed, the performance has been as expected. There was significant improvements in network traffic and a large reduction in vMotion times. Because of the additional bandwidth to the blades and progresses in CPU and RAM, a higher consolidation of virtual machines was achievable. BYU moved from 10 VMs per host to almost 50 VMs per host while at the same time reducing physical servers by almost 80%; cable and switch reduction has been over 95%. This has significantly reduced the amount of management time and resources that has gone into the communications infrastructure. These savings have allowed the server and storage team to focus on other business strategies.

There has been some learning with the new system as well. Xsigo has worked hard to make Infiniband turn-key so that system administrators don't need to know anything about it.

However, it is important in large environments that the system administrators do understand how Infiniband works. This knowledge aids in the troubleshooting process and helps to make better design decisions. Although BYU ran into several problems along the way, Xsigo worked hard to get them resolved quickly. In October 2012, Xsigo was acquired by Oracle and there was hope that a larger company would provide more resources for the product. However, the last year has been very challenging to get the right support for the product. Oracle has shifted the focus of the once vendor neutral product to be more focused on Oracle servers and Oracle Virtual Machine. Support for new VMware drivers or fixes has all but stalled since the acquisition which impedes BYU's ability to apply the latest security patches, bug fixes, and features from VMware.

Given that compute capacity will increase and the performance of storage will ramp up to even higher I/Os, with SSDs becoming a commodity, Infiniband technically is the right choice for high performance, highly virtualized data centers. However, given that Mellanox has not gone mainstream in the data center market, there are no good alternatives to Oracle OVN on Infiniband. BYU is currently evaluating many options which may include retrofitting the blades with FCoE for the production traffic and leveraging Infiniband to provide high performance vMotion.

6.3 Recommendations

The primary recommendation is that system administrators who are contemplating converged I/O, leverage the framework to examine the focus areas of the technology and how it will impact their environment. Each environment is different and so are the objectives of the organization. The framework allows for tests to be constructed to accommodate these differences and leverage them as advantages to choosing the right technology.

The secondary recommendations from this study is for the vendors. Both OVN and Cisco UCS should focus performance efforts around the contention scenarios. OVN can leverage Infiniband's non-reliable communication mechanism in order to improve UDP performance during times of congestion. This will take some rework of the drivers in order to intercept the packets, interpret them, and send them down the appropriate transport method. Cisco conversely can look to improve their TCP performance during congestion which will help them perform vMotions faster, especially when many hosts may enter maintenance mode at the same time.

6.4 Future Work

There are many aspects of this study that can be expanded upon. The goal of this work was to develop an analysis framework and test it using a case study laying the groundwork for future experiments.

The following aspects of the framework could be expanded:

1. Performance testing should include the same tests on the current traditional networks to establish a baseline to compare the converged solutions to.
2. Interactions of different network and storage protocols in a contention situation are important and should be tested as well as each protocol separately.
3. The usability testing should include a time between tests to help determine if the new procedures are learned or memorized.

The following areas within the study could use further exploration:

1. What types of TCP and/or UDP traffic causes the most congestion in these converged architectures?

2. What are the effects of congestion with simultaneous TCP and UDP traffic and how does it compare to traditional networks under similar congestion scenarios?
3. The storage component of the study could be expanded to see the effects of congestion on the converged architecture and how it compares to a traditional Fibre Channel architecture.
4. VMware is experimenting with RDMA for vMotion and other services, is there a difference between RDMA over Ethernet and Infiniband?
5. Usability testing could be greatly expanded to include much more complex configuration, troubleshooting, and monitoring tasks. Also extending the time between tasks will help determine how much of the tasks are learned versus memorized.
6. A more encompassing evaluation of other FCoE and converged solutions can be performed to give a more accurate cost model for converged architectures.
7. An analysis of the time system administrators spend on traditional systems compared to converged architectures would provide some definitive evidence of what BYU's experience has shown.
8. A power analysis of actual power usage versus the values published in the vendor documentation would give a better view of what the real world power savings would be in a converged infrastructure.

This study has shown that the analysis framework developed is useful for determining if converged communication networks can provide organizations with many benefits in terms of cost, innovation, speed, and better management. As converged infrastructures becomes more

mainstream, the technology will continue to improve and innovate further enhancing the benefits identified in this study.

REFERENCES

- Ayoub, Ali. 2012. "Ethernet Services over IPoIB." https://www.openfabrics.org/ofa-documents/doc_download/510-ethernet-tunneling-over-ipoib.html.
- Barham, Paul, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. 2003. "Xen and the Art of Virtualization." *ACM SIGOPS Operating Systems Review* 37 (5): 164. Accessed January 31, 2014. doi:10.1145/1165389.945462.
- Boles, Jeff. 2011. "Auto Tiering Crucial to Storage Efficiency." *SearchStorage*. Accessed January 02, 2014. <http://searchstorage.techtarget.com/podcast/Auto-tiering-an-important-technology-in-todays-storage-industry>.
- Bugnion, Edouard, Scott Devine, Kinshuk Govil, and Mendel Rosenblum. 1997. "Disco: Running Commodity Operating Systems on Scalable Multiprocessors." *ACM Transactions on Computer Systems* 15 (4): 412–47. doi:10.1145/265924.265930.
- Byrne, Jeff. 2012. "InfiniBand Networking for Storage and Converged Data Centers." *SearchStorage*. Accessed January 03, 2014. <http://searchstorage.techtarget.com/opinion/InfiniBand-networking-for-storage-and-converged-data-centers>.
- Chu, Albert. 2012. "Infiniband - You Gotta Have Pretty Good Game." *There Are Many Fish in the Sea ... but There's Only One Bacon (blog)*. Accessed January 17, 2014. <http://albertchu.blogspot.com/2012/02/infiniband-you-gotta-have-pretty-good.html>.
- Cisco. 2007. "Cisco SFS InfiniBand Software Configuration Guide." http://www.cisco.com/en/US/docs/server_nw_virtual/2.10.0_release/configuration/guide/swcg.pdf.
- . 2012. "Cisco UCS B-Series M2 Blade Servers." http://www.cisco.com/en/US/prod/collateral/ps10265/ps10280/ps10915/data_sheet_c78-588109.pdf.

- . 2008. “Cut-Through and Store-and-Forward Ethernet Switching for Low-Latency Environments.”
http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/white_paper_c11-465436.pdf.
- . 2009. “Small Computer System Interface over IP and Fibre Channel over Ethernet : A Comparison.”
http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/white_paper_c11-495142.pdf.
- . 2010. “Understanding Cisco Unified Computing System Service Profiles.”
http://www.cisco.com/en/US/prod/collateral/ps10265/ps10281/white_paper_c11-590518.pdf.
- Complete Data Recovery. 2003. “Fibre Channel Tutorial.” Accessed February 19, 2014.
http://www.recoverdata.com/fc_tutorial.htm.
- Dedlk, Jan, Gary Stephens, and Jan Dedek. 1997. *What Is Fibre Channel? (4th Ed.)*. Ancot Corp.
- Dell. 2010. “Dell™ FCoE Solutions for Server I/O Convergence.”
<http://www.dell.com/downloads/global/products/pwcnt/en/dell-fcoe-solutions-for-server-io-convergence.pdf>.
- Duffy, Jim. 2011a. “2011 Tech Priorities: Are You Ready to Flatten Your Data Center Network.” *NetworkWorld*. Accessed January 10, 2012.
<http://www.networkworld.com/news/2011/010311-outlook-tech-priorities-data-center.html>.
- . 2011b. “FCoE: From Fee to Free.” *NetworkWorld*. Accessed October 01, 2012.
<http://www.networkworld.com/news/2011/020811-fcoe-free.html>.
- Dykstra, Phil. 2001. “Protocol Overhead.” Accessed January 03, 2014.
<http://sd.wareonearth.com/~phil/net/overhead/>.
- Emerick, Ron. 2012. “PCI Express IO Virtualization Overview.”
http://www.snia.org/sites/default/education/tutorials/2012/fall/networking/RonEmerick_PCI_Express_IO_Virtualization_Overview-r2.pdf.
- Foskett, Stephen. 2011. “Multi-Hop FCoE Is Not Ready For Prime Time (Yet).” Accessed January 18, 2014. <http://blog.fosketts.net/2011/10/21/fcoe-ready-prime-time/>.
- Haff, Gordon. 2009. “I/O Virtualization’s Competing Forms.” *CNET News*. Accessed March 29, 2012. http://news.cnet.com/8301-13556_3-10381070-61.html.

- Haviv, Yaron. 2009. "Unified Fabrics Reduce Power Use." *Communications News* 46 (2). Nelson Publishing: 20–22. <https://www.lib.byu.edu/cgi-bin/remotearch.pl?url=http://search.ebscohost.com/login.aspx?direct=true&db=iuh&AN=37332548&site=ehost-live&scope=site>.
- Houston, Kevin. 2010. "Shared I/O – The Future of Blade Servers?" *Blades Made Simple*. Accessed June 08, 2014. <http://bladesmadesimple.com/2010/06/shared-io-the-future-of-blade-servers/>.
- HP. 2011. "Converged Networks with Fibre Channel over Ethernet and Data Center Bridging." <http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01681871/c01681871.pdf>.
- Kerekes, Zsolt. 2004. "Fibre-Channel SAN History - the First Decade." *StorageSearch*. Accessed December 14, 2012. <http://www.storagesearch.com/view61.html>.
- Killalea, Tom. 2008. "Meet the Virts." *Queue* 6 (1): 14–18. Accessed February 19, 2014. doi:10.1145/1348583.1348589.
- Kim, Ted H. 2004. "Brief History of InfiniBand: Hype to Pragmatism." *Musing of a Random Dude (blog)*. Accessed January 17, 2014. https://blogs.oracle.com/RandomDude/entry/history_hype_to_pragmatism.
- Kohlbrenner, Eric, Dana Morris, and Brett Morris. 2012. "The History of Virtual Machines." *The Core of Information Technology*. Accessed April 02, 2012. <http://www.cs.gmu.edu/cne/itcore/virtualmachine/history.htm>.
- Komiega, Kevin. 2010. "The Impact of Emerging Standards on LAN and SAN Infrastructures: Creating and Deploying a Unified Fabric in the Data Center Requires Consideration of Today's and Tomorrow's Cabling Systems." *Cabling Installation & Maintenance* 18 (3). PennWell Corporation: 7–11. <https://www.lib.byu.edu/cgi-bin/remotearch.pl?url=http://search.ebscohost.com/login.aspx?direct=true&db=mth&AN=48441418&site=ehost-live&scope=site>.
- LeBlanc, Robert. 2011a. "Chassis Breakdown."
- . 2011b. "Disk Usage."
- M2 Presswire. 2012. "New Xsigo Data Centre Fabric Product Line Upgrades Leading Open Fabric Architecture Vendor-Neutral Fabric Improves Software Defined Networking (SDN) Management, Supports Five Major Hypervisors with 56 Gbps Server-to-Server Connections", June 6. <http://search.proquest.com/docview/1018643898?accountid=4488>.

- Maleval, Jean-Jaques. 2013. "Ten-Year Old NextIO Closed Doors." *StorageNewsletter*. Accessed January 20, 2014. <http://www.storagenewsletter.com/rubriques/start-ups/nextio-closed-doors/>.
- Mearian, Lucas. 2012. "Protocol Wars: Can Fibre Channel Survive Ethernet's Assault?" *ComputerWorld*. Accessed January 03, 2014. http://www.computerworld.com/s/article/9226756/Protocol_wars_Can_Fibre_Channel_survive_Ethernet_s_assault_?taxonomyId=19&pageNumber=5.
- Mellanox. 2014. "An Introduction to InfiniBand." Accessed February 19, 2014. http://www.mellanox.com/pdf/whitepapers/IB_Intro_WP_190.pdf.
- . 2012. "Building a Scalable Storage with InfiniBand." http://www.mellanox.com/related-docs/whitepapers/WP_Scalable_Storage_InfiniBand_Final.pdf.
- Modine, Austin. 2010. "Cisco Cuts and Runs on Dell Blade Partnership." *The Register*. Accessed January 18, 2014. http://www.theregister.co.uk/2010/02/16/report_cisco_cuts_4001d_blade_switch/.
- Morgan, Timothy Prickett. 2011. "NextIO Punts I/O Virtualizing Maestro." *The Register*. Accessed January 20, 2014. http://www.theregister.co.uk/2011/10/19/nextio_vnet_maestro_switch/.
- . 2009. "Super Interconnect Maker Quadrics Going Titsup." *The Channel*. Accessed January 17, 2014. http://www.theregister.co.uk/2011/10/19/nextio_vnet_maestro_switch/.
- Munjal, Deepak. 2010. "FCoE: Out of the Frying Pan and into the Fire." *The HP Blog Hub (blog)*. Accessed January 18, 2014. <http://h30507.www3.hp.com/t5/HP-Networking/FCoE-Out-of-the-frying-pan-and-into-the-fire/ba-p/85879#.UwP-w5CJAa1>.
- . 2011. "This Is Not the Convergence You Were Looking for." *The HP Blog Hub (blog)*. Accessed January 18, 2014. http://h30507.www3.hp.com/t5/HP-Networking/This-is-not-the-convergence-you-were-looking-for/ba-p/91225#.UwP_HZCJAa1.
- Musich, Paula. 2008. "Cisco Fills Out Convergent Data Center Switch Line." *eWeek*. Accessed October 01, 2012. <http://www.eweek.com/c/a/IT-Infrastructure/Cisco-Fills-Out-Convergent-Data-Center-Switch-Line>.
- Network Working Group. 2006. "RFC 4755 - IP over InfiniBand: Connected Mode (RFC4755)." <http://www.faqs.org/rfcs/rfc4755.html>.
- Oltsik, Jon. 2009. "The Good and Bad of Cisco's UCS Servers." *CNET News*. Accessed January 18, 2014. http://news.cnet.com/8301-1001_3-10197139-92.html.

- Oracle. 2012. “Delivering Application Performance with Oracle’s InfiniBand Technology.” <http://www.oracle.com/technetwork/server-storage/networking/documentation/o12-020-1653901.pdf>.
- . 2013. “Oracle Virtual Networking Overview and Frequently Asked Questions.” <http://www.oracle.com/us/products/networking/virtual-networking/oracle-virtual-networking-faq-1875177.pdf>.
- Parallels. 2012. “Parallels® Virtuozzo Containers An Introduction to OS Virtualization and Parallels Virtuozzo Containers.” http://www.parallels.com/fileadmin/parallels/documents/hosting-cloud-enablement/pvc/whitepapers/Parallels_Virtuozzo_Containers_WP_an_introduction_to_os_EN.pdf.
- Pinder, Steve. 2010. “Thin Provisioning: Over-Allocation, Wide Striping, Space Reclamation.” *ComputerWeekly*. Accessed January 02, 2014. <http://www.computerweekly.com/answer/Thin-provisioning-Over-allocation-wide-striping-space-reclamation>.
- Popek, Gerald J., and Robert P. Goldberg. 1974. “Formal Requirements for Virtualizable Third Generation Architectures.” *Communications of the ACM* 17 (7): 412–21. doi:10.1145/361011.361073.
- Poulton, Nigel. 2009. “Xsigo – Try It Out, I Dare You!” *Nigelpoulton.com (blog)*. Accessed January 20, 2014. <http://blog.nigelpoulton.com/xsigo-try-it-out-i-dare-you/>.
- Rice, John A. 1995. *Mathematical Statistics and Data Analysis*. Edited by Jennifer Burger. Second Edi. Belmont, CA: Duxbury Press.
- Rosenblum, Mendel. 2004. “The Reincarnation of Virtual Machines.” *Queue* 2 (5): 34. Accessed February 19, 2014. doi:10.1145/1016998.1017000.
- ServerFault. 2011. “How Do I Set up IP over FC.” Accessed January 02, 2014. <http://serverfault.com/questions/300699/how-do-i-set-up-ip-over-fc>.
- Stansberry, Matt. 2013. “Uptime Institute 2013 Data Center Industry Survey.” Accessed June 08, 2014. <http://www.pducables.com/documents/uptime-institute-2013-data-center-survey.pdf>.
- Sultan, Omar. 2009. “What Is a FEX?” *Cisco Blogs (blog)*. Accessed January 18, 2014. http://blogs.cisco.com/datacenter/what_is_a_fex/.
- The Data Center Overlords (blog). 2011. “A Tale of Two FCoEs.” Accessed January 20, 2014. <http://datacenteroverlords.com/2011/11/21/a-tale-of-two-fcoes/>.

- VMware. 2012. "VMware KB: Configuring Xsigo Systems I/O Directors on ESX 4.x, ESXi 4.x and ESXi 5.x (Partner Verified and Support)." http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2005239.
- Waldspurger, Carl, and Mendel Rosenblum. 2012. "I/O Virtualization." *Communications of the ACM* 55 (1): 66. Accessed February 19, 2014. doi:10.1145/2063176.2063194.
- Wen, Heming, Prabhat Kumar Tiwary, and Tho Le-Ngoc. 2013. *Wireless Virtualization. Wireless Virtualization SE - 1*. SpringerBriefs in Computer Science. Cham: Springer International Publishing. Accessed February 19, 2014. doi:10.1007/978-3-319-01291-9.
- Wikipedia. 2012a. "IBM CP-40." Accessed March 17, 2012. http://en.wikipedia.org/wiki/IBM_CP-40.
- . 2012b. "IEEE P802.1p." Accessed December 14, 2012. http://en.wikipedia.org/wiki/IEEE_802.1p.
- . 2014c. "Storage Virtualization." Accessed January 02, 2014. http://en.wikipedia.org/wiki/Storage_virtualization.
- . 2012d. "Timeline of Virtualization Development." Accessed March 17, 2012. http://en.wikipedia.org/wiki/Timeline_of_virtualization_development.
- . 2014e. "Virtualization." Accessed January 02, 2014. <http://en.wikipedia.org/wiki/Virtualization>.
- Zeeman, Ron. 2014. "BYU Data Center PUE."

APPENDICES

APPENDIX A: GLOSSARY OF TERMS

ATM – Asynchronous Transfer Mode

ASIC – Application-Specific Integrated Circuit

ATM – Asynchronous Transfer Mode

BIOS – Basic Input/Output System

CapEx – Capital Expense

CEE – Converged Enhanced Ethernet

Cisco UCS – Cisco Unified Computing System

CNA – Converged Network Adapter

CPU – Central Processing Unit

DCBX – Data Center Bridging Exchange Protocol

DMA – Direct Memory Access

DR – Disaster Recovery

eIPoIB – Ethernet tunneling over IpoIB, Ethernet tunneling over IP over Infiniband

EoIB – Ethernet over IB, Ethernet over Infiniband

ETS – Enhanced Transmission Selection

FCoE – Fibre Channel over Ethernet

FCoIB – FC over IB, Fibre Channel over Infiniband

FDR – Fourteen Data Rate

EoR – End of Row

Fex module – Fabric Extension module

GBIC – Gigabit Interface Converter

GPU – Graphical Processing Unit

HCA – Host Channel Adapter

IEEE – Institute of Electrical and Electronics Engineers

IB - Infiniband

I/O – Input/Output

I/Ops – I/O per second

IOV – I/O Virtualization

IP – Internet Protocol

IpoIB – IP over IB, Internet Protocol over Infiniband

iSCSI – Internet Small Computer System Interface

iSER – iSCSI Extensions for RDMA

LAN – Local Area Network

LUN – Logical Unit Number

MAC – Medium Access Control

MoR – Middle of Row

MPI – Message Passing Interface

MPLS – Multiprotocol Label Switching

MTU – Maximum Transmission Unit

NAS – Network Attached Storage

NL-SAS – Near-Line SAS

OFED – Open Fabrics Enterprise Distribution

OpEx – Operational Expense

OS – Operating System

PCI – Peripheral Component Interconnect

PCI-e – PCI Express

PFC – Priority-based Flow Control

PUE – Power Usage Effectiveness

QCN – Quantized Congestion Notification

QDR – Quad Data Rate

RAID – Redundant Array of Independent Disks

RAM – Random Access Memory

RDMA – Remote Direct Memory Access

SAN – Storage Area Network

SAS – Serial Attached SCSI

SATA – Serial ATA

SCSI – Small Computer System Interface

SDRS – Storage Distributed Resource Scheduler

SRP – SCSI RDMA Protocol

SSD – Solid State Disk

Storage DRS – Storage Distributed Resource Scheduler

SVS – Altiris Software Virtualization Solution

ToR – Top of Row

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

VLAN – Virtual Local Area Network

VM – Virtual Machine

VMDK – Virtual Machine Disk

VMFS – Virtual Machine File System

VXLAN – Virtual eXtensible Local Area Network

WWN – World Wide Name

APPENDIX B: AUTOMATION SCRIPTS

```
##### copycpuburn.bat #####

@echo off
net use x: /delete
net use x: \\testcontroller.byu.edu\share
xcopy x:\cpuburn\* c:\windows\system32\*
net use x: /delete

##### getDateofLinuxBoxes.bat #####

@echo off
::get date
plink cn3 -l capstone date
plink cn4 -l capstone date
plink cn5 -l capstone date
plink cn6 -l capstone date
plink cn8 -l capstone date
plink cn11 -l capstone date
plink cn13 -l capstone date
plink cn14 -l capstone date
plink cn16 -l capstone date
plink cn20 -l capstone date

##### hosts.txt #####

iperf0.byu.edu
iperf1.byu.edu

##### initiateTCPTest.bat #####

::@echo off
::Sample run udp client .ps1
echo Mapping drives...
set str=%computername%
net use v: /delete
net use v: \\testController\share /user:capstone c@pst0n3
net use x: /delete
net use x: \\testController\logs\%str% /user:capstone c@pst0n3
echo Done mapping.
echo Starting UDP Client.
powershell v:\runTcpClient.ps1

##### initiateUDPTTest.bat #####

::@echo off
::Sample run udp client .ps1
echo Mapping drives...
set str=%computername%
net use v: /delete
net use v: \\testController\share /user:capstone c@pst0n3
net use x: /delete
net use x: \\testController\logs\%str% /user:capstone c@pst0n3
echo Done mapping.
```

```

echo Starting UDP Client.
powershell v:\runUdpClient.ps1

##### iperf0Host.bat #####

@echo off
psexec \\iperf0.byu.edu -u capstone -p c@pst0n3 C:\script\iperfHost.bat

##### iPerf-lset.ps1 #####

#
# Powershell Run iPerf script
# FJS, 2/17/2011
#

###Variables

#nameArray // dns name for each VM
$nameArray = @(("testController.byu.edu","capwintemplate.byu.edu"),
              ("iperf0.byu.edu","iperf1.byu.edu"));
#VMoSArray // OS for each VM in the nameArray
$VMoSArray = @(("win","win"),
              ("win","win"));

$lin = "lin";
$win = "win";

$vmA = "";
$vmB = "";
$vmAOS = "";
$vmBOS = "";

#1. Randomly pick which set, and assign the VMs for that set. Also assign OS
$rand = New-Object System.Random
$set = $rand.next(1,3);
switch ($set){
    1 {$vmA = $nameArray[0][0]; $vmAOS = $VMoSArray[0][0]; $vmB =
$nameArray[1][0]; $vmBOS = $VMoSArray[1][0];}
    2 {$vmA = $nameArray[0][1]; $vmAOS = $VMoSArray[0][1]; $vmB =
$nameArray[1][1]; $vmBOS = $VMoSArray[1][1];}
    default {"Set could not be determined."}
}
#2. Now that have the set, determine which command to run for each machine in
the set.
#foreach VM check for Os, and assign Script.
Write-Host $vmA, $vmB, $vmAOS, $vmBOS;

if ($vmAOS.compareTo("lin").equals(0)){
Write-Host "It's Linux"
#Run vmA script
}else {
Write-Host "It's Windows"
#run vmA script
}

##### killAllHosts.bat #####

@echo off
::Kill All Hosts!!!!
START plink cn11 -l capstone pkill iperf
START psexec \\10.11.31.112.byu.edu -u administrator -p c@pst0n3 TASKKILL /F /FI
"IMAGENAME eq iperf*"
START plink cn13 -l capstone pkill iperf
START plink cn14 -l capstone pkill iperf
START psexec \\10.11.31.115.byu.edu -u administrator -p c@pst0n3 TASKKILL /F /FI

```

```

"IMAGENAME eq iperf*"
START plink cn16 -l capstone pkill iperf
START psExec \\10.11.31.117 -u administrator -p c@pst0n3 TASKKILL /F /FI
"IMAGENAME eq iperf*"
START psExec \\10.11.31.118 -u administrator -p c@pst0n3 TASKKILL /F /FI
"IMAGENAME eq iperf*"
START psExec \\10.11.31.119 -u administrator -p c@pst0n3 TASKKILL /F /FI
"IMAGENAME eq iperf*"
START plink cn20 -l capstone pkill iperf

#### KillLoad.bat ####

@echo off
::Kill load
START TASKKILL /F /FI "IMAGENAME eq burn*"
START TASKKILL /F /FI "IMAGENAME eq python*"

#### readingfile.ps1 ####

#
# iPerf Testing
#

#VARIABLES
$vms = @();
$jobs = @();
$winScript;
$linScript;

#Read in the file
foreach ($line in Get-Content "C:\Scripts\jordan\servers.txt"){
    $line = $line.split(",");
    #Create vmObj
    $vmObj = New-Object Object;
    #add myDnsName
    $vmObj | Add-Member NoteProperty myDnsName $line[0];
    #add OS
    $vmObj | Add-Member NoteProperty OS $line[1];
    #add vmobj to vms array
    $vms += $vmObj;
}

#Check for valid number of servers
if (($vms.length % 2) -ne 0){
    Write-Host "You have an odd number of servers in the source file. Try
again."
    exit;
}

#Get an incrementing value for the VMS. For example, if there are 4 vms, then
the complement
#of vms[0] will be vms[2], vms[1] will be vms[3]; We loop until we get to 5,
then we stop. Everything after
#is just an iperf server that listens, so it doesn't need a complimentary vm
$inc = $vms.length / 2;
$i = 0;
foreach ($vm in $vms){
    if($i -le $inc-1){
        $vm | Add-Member NoteProperty hostVM $vms[$i + $inc].myDnsName;
        $vm | Add-Member NoteProperty type client;
    }else{
        $vm | Add-Member NoteProperty hostVM x;
        $vm | Add-Member NoteProperty type host;
    }
    #Print vm properties
    $vm;
    #increment i
}

```

```

    $i++;
    #if we passed our halfway point, break
    if ($i > $inc){
    break;
    }
}
Write-Host;
#Reset $i for later use
$i = 0;

#Loop through objects and add script property, basically what we want to pass
through psexec or plink
##For linux, each script needs to be in the home directory and have execute
permissions.

#Windows iperf HostScript
    #psexec \\myDnsName -u user -p PathTowinHost.bat
#Linux iperf HostScript
    #plink myDnsName -l user perl linHost.pl
#Windows iperf ClientScript
    #? how to specify host for the batch script? parameter
#Lunx iperf ClientScript
    #? how to specify host for perl script?

#loop through the array, and run the script associated with each VM
## START iPerf HOSTS

#cmd /c psexec \\iperf0.byu.edu -u capstone -p c@pst0n3 C:\script\iperfHost.ba
foreach ($vm in $vms){
    #Start the hosts...
    if ($vm.type.equals("host")){
        if ($vm.OS.equals("win")){
            #Create the Windows version
            $a = "psexec \\";
            $b = " -u capstone -p c@pst0n3 C:\script\iperfHost.bat";
            $vm.myDnsName;
            $z = $a + $vm.myDnsName + $b;
            #run the script
            Start-Job{
                cmd /c $z;
            }
            #jobs += $job;
        } else {
            Write-Host "Starting job for" $vm.myDnsName;
            $job = Start-Job -ScriptBlock{
                #If this fails it's because the testController
                doesn't have all key access to the box
                plink $vm.myDnsName -l capstone perl iperfHost.pl;
            }
            $jobs += $job;
        }
    }
}

## START iperf CLIENTS

#Save it back?

```

```

##### runcopy.bat #####

@echo off
psexec @winboxes.txt -u capstone -p c@pst0n3 -c
C:\Scripts\jordan\dev\copycpuburn.bat

##### runIperfHostsDefault.bat #####

@echo off
START iperf -s >> C:\iperf-%TIME:~0,2%-~3,2%-~6,2%.log

##### runTCPClient.bat #####

::Windows iperf Client script
set str=%computername%
echo.%str%
set str=%str:cn=%
echo.%str%
set /a host=%str%+10
echo %host%
set host=cn%host%
echo %host%
net use x: /delete
net use x: \\testController\logs\%computername% /user:capstone c@pst0n3
echo %computername%-~4,2%-~7,2%-~0,2%-~3,2%-~6,2%
iperf -c %host% -f m -t 120 -i 1 >> x:\%computername%-~4,2%-~7,2%-~0,2%-~3,2%-~6,2%.txt

##### runTest10.bat #####

@echo off
::RUN 10 TESTS
START z:\jordan\dev\killAllBurn.bat
START z:\jordan\dev\killAllHosts.bat
START z:\jordan\dev\jumboFramesON.bat
START z:\jordan\dev\startAllHosts.bat
START z:\jordan\dev\startAllBurn.bat
z:\jordan\dev\startAllClients.bat

##### runUDPClient.bat #####

::Windows iperf Client script
set str=%computername%
echo.%str%
set str=%str:cn=%
echo.%str%
set /a host=%str%+10
echo %host%
set host=cn%host%
echo %host%
net use x: /delete
net use x: \\testController\logs\%computername% /user:capstone c@pst0n3
::xcopy C:\CN* x:\CN*
time /t
iperf -c %host% -u -i 1 -t 300 -f m -b 10g >> x:\%computername%-~4,2%-~7,2%-~0,2%-~3,2%-~6,2%.txt
time /t

##### saveBackData.bat #####

net use x: /delete
net use x: \\testcontroller.byu.edu\share c@pst0n3 /USER:testcontroller\capstone
mkdir x:\%computername%
for /f "tokens=1-5 delims=%" %%d in ("%time%") do echo YES >> x:\%computername%\%d-%e.txt

```

```

##### servers.txt #####

testcontroller.byu.edu,win
capwintemplate.byu.edu,win
iperf0.byu.edu,win
iperf1.byu.edu,win

##### startAllClients.bat #####

@echo off
::Starts iperf for all Clients
START psexec \\10.11.31.101 -u administrator -p c@pst0n3 -c -f runClient.bat
START psexec \\10.11.31.102 -u administrator -p c@pst0n3 -c -f runClient.bat
START plink cn3 -l capstone perl /mnt/testShare/runClient.pl
START plink cn4 -l capstone perl /mnt/testShare/runClient.pl
START plink cn5 -l capstone perl /mnt/testShare/runClient.pl
START plink cn6 -l capstone perl /mnt/testShare/runClient.pl
START psexec \\10.11.31.107 -u administrator -p c@pst0n3 -c -f runClient.bat
START plink cn8 -l capstone perl /mnt/testShare/runClient.pl
START psexec \\10.11.31.109 -u administrator -p c@pst0n3 -c -f runClient.bat
START psexec \\10.11.31.110 -u administrator -p c@pst0n3 -c -f runClient.bat

##### startAllTCPClients.bat #####

@echo off
::Starts iperf for all Clients
START psexec \\10.11.31.101 -u administrator -p c@pst0n3 -c -f
initiateTCPTest.bat
START psexec \\10.11.31.102 -u administrator -p c@pst0n3 -c -f
initiateTCPTest.bat
START plink cn3 -l capstone perl /mnt/testShare/runTCPClient.pl
START plink cn4 -l capstone perl /mnt/testShare/runTCPClient.pl
START plink cn5 -l capstone perl /mnt/testShare/runTCPClient.pl
START plink cn6 -l capstone perl /mnt/testShare/runTCPClient.pl
START psexec \\10.11.31.107 -u administrator -p c@pst0n3 -c -f
initiateTCPTest.bat
START plink cn8 -l capstone perl /mnt/testShare/runTCPClient.pl
START psexec \\10.11.31.109 -u administrator -p c@pst0n3 -c -f
initiateTCPTest.bat
START psexec \\10.11.31.110 -u administrator -p c@pst0n3 -c -f
initiateTCPTest.bat

##### startAllTCPHosts.bat #####

@echo off
::Start iPerf on all Hosts
START plink cn11 -l capstone iperf -s -i 1
START psexec \\10.11.31.112 -u administrator -p c@pst0n3 iperf -s -i 1
START plink cn13 -l capstone iperf -s -i 1
START plink cn14 -l capstone iperf -s -i 1
START psexec \\10.11.31.115 -u administrator -p c@pst0n3 iperf -s -i 1
START plink cn16 -l capstone iperf -s -i 1
START psexec \\10.11.31.117 -u administrator -p c@pst0n3 iperf -s -i 1
START psexec \\10.11.31.118 -u administrator -p c@pst0n3 iperf -s -i 1
START psexec \\10.11.31.119 -u administrator -p c@pst0n3 iperf -s -i 1
START plink cn20 -l capstone iperf -s -i 1

##### startAllUDPClients.bat #####

@echo off
::Starts iperf for all Clients
date /t
time /t

```

```

START psexec \\10.11.31.101 -u administrator -p c@pst0n3 -c -f
initiateUDPTTest.bat
START psexec \\10.11.31.102 -u administrator -p c@pst0n3 -c -f
initiateUDPTTest.bat
START plink cn3 -l capstone perl /mnt/testShare/runUDPCClient.pl
START plink cn4 -l capstone perl /mnt/testShare/runUDPCClient.pl
START plink cn5 -l capstone perl /mnt/testShare/runUDPCClient.pl
START plink cn6 -l capstone perl /mnt/testShare/runUDPCClient.pl
START psexec \\10.11.31.107 -u administrator -p c@pst0n3 -c -f
initiateUDPTTest.bat
START plink cn8 -l capstone perl /mnt/testShare/runUDPCClient.pl
START psexec \\10.11.31.109 -u administrator -p c@pst0n3 -c -f
initiateUDPTTest.bat
START psexec \\10.11.31.110 -u administrator -p c@pst0n3 -c -f
initiateUDPTTest.bat

##### startAllUDPHosts.bat #####

@echo off
::Start iPerf on all Hosts
START plink cn11 -l capstone iperf -s -u
START psexec \\10.11.31.112 -u administrator -p c@pst0n3 iperf -s -u
START plink cn13 -l capstone iperf -s -u
START plink cn14 -l capstone iperf -s -u
START psexec \\10.11.31.115 -u administrator -p c@pst0n3 iperf -s -u
START plink cn16 -l capstone iperf -s -u
START psexec \\10.11.31.117 -u administrator -p c@pst0n3 iperf -s -u
START psexec \\10.11.31.118 -u administrator -p c@pst0n3 iperf -s -u
START psexec \\10.11.31.119 -u administrator -p c@pst0n3 iperf -s -u
START plink cn20 -l capstone iperf -s -u

##### startIperfHost.bat #####

@echo off
START iperf -s

##### test.ps1 #####

#Run iPerf on Machine 1 and two
#Variables
$thread1 = Start-Job -ScriptBlock{
plink caplinuxtemplate.byu.edu -l capstone perl test.pl
}
$thread2 = Start-Job -ScriptBlock{
psexec \\iperf0.byu.edu -u capstone -p c@pst0n3 -c C:\Scripts\test.bat
}
Write-Host "YEASSSSS";

## Start iperf in server mode on cn11-cn20
#cn11
$thread11 = Start-Job -ScriptBlock{
plink cn11.byu.edu -l capstone perl test.pl
}
#cn12
$thread12 = Start-Job -ScriptingBlock{
psexec \\cn12.byu.edu -u capstone -p c@pst0n3 -c C:\Scripts\test.bat
}
#cn13
$thread13 = Start-Job -ScriptingBlock{
plink cn13.byu.edu -l capstone perl test.pl
}
#cn14
$thread14 = Start-Job -ScriptingBlock{
plink cn14.byu.edu -l capstone perl test.pl
}

```



```

#cn15
$thread15 = Start-Job -ScriptingBlock{
psexec \\cn15.byu.edu -u capstone -p c@pst0n3 -c C:\Scripts\test.bat
}
#cn16
$thread16 = Start-Job -ScriptingBlock{
plink cn16.byu.edu -l capstone perl test.pl
}
#cn17
$thread17 = Start-Job -ScriptingBlock{
psexec \\cn17.byu.edu -u capstone -p c@pst0n3 -c C:\Scripts\test.bat
}
#cn18
$thread18 = Start-Job -ScriptingBlock{
psexec \\cn18.byu.edu -u capstone -p c@pst0n3 -c C:\Scripts\test.bat
}
#cn19
$thread19 = Start-Job -ScriptingBlock{
psexec \\cn19.byu.edu -u capstone -p c@pst0n3 -c C:\Scripts\test.bat
}
#cn20
$thread20 = Start-Job -ScriptingBlock{
plink cn20.byu.edu -l capstone perl test.pl
}

##### turnOffALLJumboFrames.bat #####

@echo off
::Turn on Jumbo Frames for All
psexec \\10.11.31.101 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
psexec \\10.11.31.102 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
plink cn3 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
plink cn4 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
plink cn5 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
plink cn6 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
psexec \\10.11.31.107 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
plink cn8 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
psexec \\10.11.31.109 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
psexec \\10.11.31.110 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
plink cn11 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
psexec \\10.11.31.112 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
plink cn13 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
plink cn14 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
psexec \\10.11.31.115 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
plink cn16 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl
psexec \\10.11.31.117 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
psexec \\10.11.31.118 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
psexec \\10.11.31.119 -u Administrator -p c@pst0n3 -c -f turnOffJumboFrames.bat
plink cn20 -l capstone perl /mnt/testShare/turnOffJumboFrames.pl

##### turnOffAllLoads.bat #####

@echo off
::Turn off all Loads
psexec @winboxes.txt -u administrator -p c@pst0n3 -c -f killLoad.bat
START plink -l capstone cn3 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn4 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn5 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn6 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn8 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn11 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn13 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn14 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn16 perl /mnt/testShare/killBurn.pl
START plink -l capstone cn20 perl /mnt/testShare/killBurn.pl

##### turnOffJumboFrames.bat #####

```

```

@echo off
::Turn on Jumbo Frames for Windows Boxes
netsh interface ipv4 set subinterface "Primary" mtu=1500 store=persistent

##### turnOnALLJumboFrames.bat #####

@echo off
::Turn on Jumbo Frames for All
psexec \\10.11.31.101 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
psexec \\10.11.31.102 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
plink cn3 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
plink cn4 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
plink cn5 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
plink cn6 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
psexec \\10.11.31.107 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
plink cn8 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
psexec \\10.11.31.109 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
psexec \\10.11.31.110 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
plink cn11 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
psexec \\10.11.31.112 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
plink cn13 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
plink cn14 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
psexec \\10.11.31.115 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
plink cn16 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl
psexec \\10.11.31.117 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
psexec \\10.11.31.118 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
psexec \\10.11.31.119 -u Administrator -p c@pst0n3 -c -f turnOnJumboFrames.bat
plink cn20 -l capstone perl /mnt/testShare/turnOnJumboFrames.pl

##### turnOnAllLoads.bat #####

@echo off
::Map the Drives
psexec @winboxes.txt -u administrator -p c@pst0n3 -c -f turnOnLoad.bat
START plink cn3 -l capstone python /mnt/testShare/loads/cn3.py
START plink cn4 -l capstone python /mnt/testShare/loads/cn4.py
START plink cn5 -l capstone python /mnt/testShare/loads/cn5.py
START plink cn6 -l capstone python /mnt/testShare/loads/cn6.py
START plink cn8 -l capstone python /mnt/testShare/loads/cn8.py
START plink cn11 -l capstone python /mnt/testShare/loads/cn11.py
START plink cn13 -l capstone python /mnt/testShare/loads/cn13.py
START plink cn14 -l capstone python /mnt/testShare/loads/cn14.py
START plink cn16 -l capstone python /mnt/testShare/loads/cn16.py
START plink cn20 -l capstone python /mnt/testShare/loads/cn20.py

##### turnOnJumboFrames.bat #####

@echo off
::Turn on Jumbo Frames for Windows Boxes
netsh interface ipv4 set subinterface "Primary" mtu=9000 store=persistent

##### turnOnLoad.bat #####

echo off
echo Mapping share
set str=%computername%
net use v: /delete
net use v: \\testController.byu.edu\share /user:capstone c@pst0n3
echo done.
echo Mapping log folder
net use x: /delete
net use x: \\testController.byu.edu\logs\%str% /user:capstone c@pst0n3
echo done.
echo Starting to run python scripts...

```

```
START python v:\loads\%str%.py
```

```
##### winboxes.txt #####
```

```
cn1.byu.edu  
cn2.byu.edu  
cn7.byu.edu  
cn9.byu.edu  
cn10.byu.edu  
cn12.byu.edu  
cn15.byu.edu  
cn17.byu.edu  
cn18.byu.edu  
cn19.byu.edu
```

```
##### winBurn.txt #####
```

```
cn1.byu.edu  
cn2.byu.edu  
cn7.byu.edu  
cn17.byu.edu  
cn18.byu.edu
```

APPENDIX C: NETWORK PERFORMANCE DATA

TCP, Jumbo Frames, Blade to Blade

Test Type: TCP, Jumbo Frames On, 300s Blade to Blade		
Name	Xsigo Bandwidth (Mbps)	Cisco Bandwidth (Mbps)
CN1	0.44	0.44
CN2	182	219
CN3	4,310	1,699
CN4	2,406	1,462
CN5	812	NA
CN6	3,200	2,685
CN7	148	90.4
CN8	NA	NA
CN9	146	185
CN10	80.5	0.44
CN11	0.44	0.44
CN12	182	219
CN13	4,310	1,699
CN14	2,406	1,462
CN15	812	NA
CN16	3,200	2,685
CN17	148	90.4
CN18	NA	NA
CN19	146	185
CN20	80.5	0.44

TCP, Standard Frames, Blade to Blade

Test Type: TCP, Jumbo Frames Off, 300s Blade to Blade		
Name	Xsigo Bandwidth (Mbps)	Cisco Bandwidth (Mbps)
CN1	241	2.51
CN2	257	252
CN3	1,464	651
CN4	2,406	750
CN5	418	887
CN6	1,060	633
CN7	165	114
CN8	578	711
CN9	202	212
CN10	96.4	1.76
CN11	241	2.51
CN12	257	252
CN13	1,464	651
CN14	2,406	750
CN15	418	887
CN16	1,060	633
CN17	165	114
CN18	578	711
CN19	202	212
CN20	96.4	1.76

TCP, Jumbo Frames, Chassis to Chassis

Test Type: TCP, Jumbo Frames On 300s, Chassis to Chassis		
Name	Xsigo Bandwidth (Mbps)	Cisco Bandwidth (Mbps)
CN1	0.43	0.44
CN2	232	153
CN3	3,583	2,265
CN4	NA	2,052
CN5	NA	NA
CN6	4,486	2,270
CN7	119	89.7
CN8	NA	NA
CN9	212	119
CN10	0.44	0.44
CN11	0.43	0.44
CN12	232	153
CN13	3,583	2,265
CN14	NA	2,052
CN15	NA	NA
CN16	4,486	2,270
CN17	119	89.7
CN18	NA	NA
CN19	212	119
CN20	0.44	0.44

TCP, Standard Frames, Chassis to Chassis

Test Type: TCP, Jumbo Frames Off, 300s, Chassis to Chassis		
Name	Xsigo Bandwidth (Mbps)	Cisco Bandwidth (Mbps)
CN1	270	3.24
CN2	336	151
CN3	788	1,048
CN4	NA	NA
CN5	241	2,230
CN6	1,656	1,196
CN7	165	151
CN8	318	1,129
CN9	305	151
CN10	88.5	3.93
CN11	270	3.24
CN12	336	151
CN13	788	1,048
CN14	NA	NA
CN15	241	2,230
CN16	1,656	1,196
CN17	165	151
CN18	318	1,129
CN19	305	151
CN20	88.5	3.93

UDP, Jumbo Frames, Blade to Blade

Test Type: UDP, Jumbo Frames On, 300s, Blade to Blade						
Name	Xsigo Bandwidth (Mbps)	Jitter(ms)	Datagram Loss	Cisco Bandwidth (Mbps)	Jitter(ms)	Datagram Loss
CN1	29.4	14.600	51%	272	1.586	3%
CN2	27.7	0.932	60%	383	NA	9%
CN3	66.9	21.687	96%	619	0.124	57%
CN4	27.0	23.652	97%	1,509	0.168	47%
CN5	3,044	NA	NA	315	0.020	88%
CN6	63.0	15.511	97%	740	0.003	63%
CN7	30.2	0.915	55%	69.5	0.915	46%
CN8	2,432	NA	NA	409	0.095	80%
CN9	7.1	0.915	83%	18.3	0.915	38%
CN10	18.6	2.276	84%	101	1.118	41%
CN11	29.4	14.600	51%	272	1.586	3%
CN12	27.7	0.932	60%	383	NA	9%
CN13	66.9	21.687	96%	619	0.124	57%
CN14	27.0	23.652	97%	1,509	0.168	47%
CN15	3,044	NA	NA	315	0.020	88%
CN16	63.0	15.511	97%	740	0.003	63%
CN17	30.2	0.915	55%	69.5	0.915	46%
CN18	2,432	NA	NA	409	0.095	80%
CN19	7.1	0.915	83%	18.3	0.915	38%
CN20	18.6	2.276	84%	101	1.118	41%

UDP, Standard Frames, Blade to Blade

Test Type: UDP, Jumbo Frames Off, 300s, Blade to Blade						
Name	Xsigo Bandwidth (Mbps)	Jitter(ms)	Datagram Loss	Cisco Bandwidth (Mbps)	Jitter(ms)	Datagram Loss
CN1	29.4	14.600	51%	272	1.586	3%
CN2	27.7	0.932	60%	383	NA	9%
CN3	66.9	21.687	96%	619	0.124	57%
CN4	27	23.652	97%	1,509	0.168	47%
CN5	3,044	NA	NA	315	0.02	88%
CN6	63	15.511	97%	740	0.003	63%
CN7	30.2	0.915	55%	69.5	0.915	46%
CN8	2,432	NA	NA	409	0.095	80%
CN9	7.1	0.915	83%	18.3	0.915	38%
CN10	18.6	2.276	84%	101	1.118	41%
CN11	29.4	14.600	51%	272	1.586	3%
CN12	27.7	0.932	60%	383	NA	9%
CN13	66.9	21.687	96%	619	0.124	57%
CN14	27	23.652	97%	1,509	0.168	47%
CN15	3,044	NA	NA	315	0.02	88%
CN16	63	15.511	97%	740	0.003	63%
CN17	30.2	0.915	55%	69.5	0.915	46%
CN18	2,432	NA	NA	409	0.095	80%
CN19	7.1	0.915	83%	18.3	0.915	38%
CN20	18.6	2.276	84%	101	1.118	41%

UDP, Jumbo Frames, Chassis to Chassis

Test Type: UDP, Jumbo Frames On, 300s, Chassis to Chassis						
Name	Xsigo Bandwidth (Mbps)	Jitter(ms)	Datagram Loss	Cisco Bandwidth (Mbps)	Jitter(ms)	Datagram Loss
CN1	13.9	0.45	60%	257.0	0.321	30%
CN2	23.8	NA	63%	217.0	0.915	29%
CN3	60.2	0.038	97%	1,779.0	0.021	44%
CN4	3,085.0	NA	NA	5,471.0	NA	NA
CN5	2,820.0	NA	NA	379.0	0.013	88%
CN6	48.0	7.764	97%	1,883.0	0.009	43%
CN7	32.4	14.659	54%	159.0	0.917	33%
CN8	578.0	NA	NA	489.0	15.627	83%
CN9	4.0	1.395	89%	123.0	0.915	44%
CN10	15.6	0.55	88%	485.0	0.909	9%
CN11	13.9	0.45	60%	257.0	0.321	30%
CN12	23.8	NA	63%	217.0	0.915	29%
CN13	60.2	0.038	97%	1,779.0	0.021	44%
CN14	3,085.0	NA	NA	5,471.0	NA	NA
CN15	2,820.0	NA	NA	379.0	0.013	88%
CN16	48	7.764	97%	1,883.0	0.009	43%
CN17	32.4	14.659	54%	159.0	0.917	33%
CN18	578.0	NA	NA	489.0	15.627	83%
CN19	4.0	1.395	89%	123.0	0.915	44%
CN20	15.6	0.55	88%	485.0	0.909	9%

UDP, Standard Frames, Chassis to Chassis

Test Type: UDP, Jumbo Frames Off, 300s, Chassis to Chassis						
Name	Xsigo Bandwidth (Mbps)	Jitter(ms)	Datagram Loss	Cisco Bandwidth (Mbps)	Jitter(ms)	Datagram Loss
CN1	198.0	0.949	31%	443.0	0.930	NA
CN2	191.0	0.915	44%	234.0	0.920	44%
CN3	57.2	2.159	93%	2,261.0	NA	NA
CN4	4,821.0	NA	NA	1,231.0	0.020	55%
CN5	22.6	53.986	NA	301.0	0.140	NA
CN6	38.0	0.022	95%	1,320.0	0.020	53%
CN7	155.0	0.915	59%	340.0	0.920	NA
CN8	38.4	31.452	NA	430.0	15.590	NA
CN9	60.4	0.915	39%	232.0	2.770	36%
CN10	80.6	15.608	33%	117.0	1.630	56%
CN11	198.0	0.949	31%	443.0	0.930	NA
CN12	191.0	0.915	44%	234.0	0.920	44%
CN13	57.2	2.159	93%	2,261.0	NA	NA
CN14	4,821.0	NA	NA	1,231.0	0.020	55%
CN15	22.6	53.986	NA	301.0	0.140	NA
CN16	38.0	0.022	95%	1,320.0	0.020	53%
CN17	155.0	0.915	59%	340.0	0.920	NA
CN18	38.4	31.452	NA	430.0	15.590	NA
CN19	60.4	0.915	39%	232.0	2.770	36%
CN20	80.6	15.608	33%	117.0	1.630	56%

APPENDIX D: VMOTION PERFORMANCE DATA

vMotion Times

Host	VM	Cisco vMotion Time	Xsigo vMotion Time	Xsigo Better By (sec)
A	cn1	0:00:21	0:00:13	8
A	cn2	0:00:32	0:00:28	4
A	cn3	0:00:10	0:00:12	-2
A	cn4	0:00:10	0:00:12	-2
A	cn5	0:00:32	0:00:30	2
A	cn6	0:00:14	0:00:17	-3
A	cn7	0:00:15	0:00:11	4
A	cn8	0:00:13	0:00:15	-2
A	cn9	0:00:33	0:00:22	11
A	cn10	0:00:29	0:00:18	11
B	cn11	0:00:10	0:00:15	-5
B	cn12	0:00:23	0:00:14	9
B	cn13	0:00:08	0:00:10	-2
B	cn14	0:00:25	0:00:25	0
B	cn15	0:00:22	0:00:11	11
B	cn16	0:00:11	0:00:13	-2
B	cn17	0:00:25	0:00:11	14
B	cn18	0:00:22	0:00:22	0
B	cn19	0:00:24	0:00:26	-2
B	cn20	0:00:15	0:00:16	-1
Total		00:06:34	00:05:41	53

APPENDIX E: STORAGE PERFORMANCE DATA

Storage Performance

Order	Vendor	Test Type	Iteration	IOps	MBps	Average Response Time (ms)
8	Cisco	db	1	128,671.86	263.52	.93
14	Cisco	db	2	128,652.61	263.48	.93
18	Cisco	db	3	128,640.55	263.46	.93
Average				128,655.01	263.49	.93
4	Xsigo	db	1	129,316.19	264.84	.93
9	Xsigo	db	2	129,349.35	264.91	.93
3	Xsigo	db	3	129,181.18	264.56	.93
Average				129,282.24	264.77	.93
16	Cisco	io	1	129,089.13	66.09	.93
5	Cisco	io	2	128,858.71	65.98	.93
7	Cisco	io	3	128,213.82	65.65	.93
Average				128,720.56	65.90	.93
12	Xsigo	io	1	129,978.33	66.55	.92
11	Xsigo	io	2	130,364.61	66.75	.92
15	Xsigo	io	3	130,058.85	66.59	.92
Average				130,133.93	66.63	.92
17	Cisco	through	1	9,384.59	615.03	12.79
1	Cisco	through	2	9,418.07	617.22	12.74
6	Cisco	through	3	9,294.25	609.11	12.91
Average				9,365.64	613.79	12.81
2	Xsigo	through	1	9,201.82	603.05	13.04
10	Xsigo	through	2	9,178.34	601.51	13.07
13	Xsigo	through	3	9,144.44	599.29	13.12
Average				9,174.87	601.28	13.08

APPENDIX F: USABILITY TESTING TASKS

Cisco Tasks to perform:

1. Create a pool of World Wide Port Numbers (WWPN) Hint: SAN tab.
2. Create a pool of MAC addresses. Hint: LAN tab.
3. Create a Service profile from the Service Template
4. Delete what you have created

Xsigo Tasks to perform:

1. Create a vNIC (hint: network cloud)
2. Create a vHBA (hint: storage cloud)
3. Create a server profile (hint: I/O templates)
4. Delete what you created.

APPENDIX G: USABILITY TESTING GRADE SHEET

NAME: _____

Occupation: _____

Cisco Tasks to perform:

Create a pool of World Wide Port Numbers (WWPN) Hint: SAN tab.

Create a pool of MAC addresses. Hint: LAN tab.

Create a Service profile from the Service Template

Delete what you have created

1st time Start time: _____

1st time End Time: _____

Errors: _____

2nd time Start time: _____

2nd time End time: _____

Errors: _____

Do you like the Interface? YES NO

What is your favorite part of the interface?

How easy was to perform the tasks?

Very Easy 1-----2-----3-----4-----5 Very Hard

Other

comments: _____

APPENDIX H: USABILITY TESTING DATA

Usability Tests

Subject	First Test	Cisco First Time	C1 Errors	Cisco Second Time	C2 Errors	Xsigo First Time	X1 Errors	Xsigo Second Time	X2 Errors	Like Xsigo?	Like Cisco?	Xsigo Ease	Cisco Ease	Favorite	Cisco Comments	Xsigo Comments
1	C	3:52	1	2:50	0	4:00	1	3:30	0	no	yes	3	2	C	like design tabs are good	its good that is web based user friendly
2	X	3:43	1	1:39	0	5:19	1	1:54	1	yes	no	2	4	X		
3	X	2:30	2	1:02	0	3:32	2	1:21	0	yes	yes	1	2	X	tree structure on left is good	create and delete is easy
4	X	3:10	1	1:13	0	4:56	3	1:52	0	no	yes	2	2	C	right click is good	needs development
5	X	2:54	1	1:26	0	3:23	2	1:10	0	no	yes	3	2	C	like java interface	
6	C	2:22	1	0:46	0	3:11	2	1:05	0	no	no	2	2	C	tabs are confusing don't like trees	no right click but + button is good
7	C	3:39	1	1:21	0	4:40	2	1:46	0	no	yes	4	1	C	familiar with other interfaces	no intuitive no feedback
8	C	2:35	2	1:20	1	3:19	1	1:02	0	yes	yes	3	2	C	right click is good	web based its good
9	C	4:38	0	1:08	0	2:26	2	1:01	1	yes	no	2	3	X	drag and drop is not intuitive	no correct placement of buttons
10	C	2:47	1	1:07	0	4:22	3	1:45	0	no	yes	5	1	C	right click is good	deleting stuff is easy
11	C	3:15	0	1:10	0	5:23	1	1:14	0	no	yes	3	2	C	familiar cisco interface	just other web interface
Total		35:25	11	15:02	1	44:31	20	17:40	2	36.36%	72.73%	54.55%	41.82%	C		

APPENDIX I: CAPEX DATA

I/O and Capital Expense Data

	HP Traditional	HP 10 Gb	HP Xsigo	Cisco UCS
Xsigo			\$57,711.52	
UCS				\$64,021.76
Ethernet	\$121,213.62	\$110,704.74	\$17,239.82	\$17,239.82
FC	\$41,150.68	\$41,150.68	\$5,219.52	\$5,219.52
I/O Costs	\$162,364.30	\$151,855.42	\$80,170.86	\$86,481.10
I/O Break out				
Cisco Blade Switch	\$10,599.00			
FC switch	\$13,293.50	\$13,293.50		
Pass through		\$4,399.12		\$5,000.00
IB QDR switch			\$11,435.60	
Quad/Dual 1Gb NIC	\$368.76			\$1,499.00
8GB FC	\$747.12	\$747.12		
10 GbE		\$615.12		
QDR IB			\$919.80	

Cisco 6509 Port Costs

Item Name	Description	Qty	Price	Ext. Price	Ports	Ext. Ports
WS-C6509-E	6509 Chassis	1	\$9,500.00	\$9,500.00		0
VS-S2T-10G	Supervisor	2	\$28,000.00	\$56,000.00		0
S2TAIK9-15001SY	OS Software	1	\$15,000.00	\$15,000.00		0
WS-X6908-10G-2T	8x10Gb line card	7	\$40,000.00	\$280,000.00	8	56
	10 Gb X2 transceiver	56	\$1,995.00	\$111,720.00		0
WS-CAC-6000W	Power supply	2	\$5,000.00	\$10,000.00		0
WS-C6509-E-FAN	Fan Tray	1	\$495.00	\$495.00		0
Total				\$482,715.00		56
Price per port				\$8,619.91		

Nexus Port Costs

Item Name	Description	Qty	Price	Ext. Price	Ports	Ext. Ports
N6004-B-24Q	6004 chassis w/ 24x40GE	1	\$70,000.00	\$70,000.00		0
N6004-M12Q	12x40GE module	6	\$27,000.00	\$162,000.00		0
N2K-C2248PQ	2000 FEX	40	\$12,000.00	\$480,000.00	48	1920
QSFP-H40G-CU3M	40GE TwinAx	80	\$650.00	\$52,000.00		0
	10 GE TwinAx	1920	\$95.00	\$182,400.00		0
Total				\$946,400.00		1920
Price per port				\$492.92		

Fibre Channel Port Costs

Item Name	Description	Qty	Price	Ext. Price	Ports	Ext. Ports
HP San Switch 8/80		1	\$69,906.20	\$69,906.20	80	80
16 port license		2	\$16,522.00	\$33,044.00		0
62.5 LC/LC 3m cable		80	\$18.00	\$1,440.00		0
Total				\$104,390.20		80
Price per Port				\$1,304.88		

Cisco UCS Port Costs

Item Name	Description	Qty	Price	Ext. Price	Ports	Ext. Ports
6248UP	Fabric Interconnect	2	\$32,000.00	\$64,000.00	48	96
6248UP Power supply	Power supply	4	\$1,400.00	\$5,600.00		0
6200 port license	License for ports 13-48	72	\$2,774.00	\$199,728.00		0
8Gb FC SFP+	FC module	8	\$260.00	\$2,080.00		0
10GE SFP+	10 GE module	8	\$995.00	\$7,960.00		0
Total				\$279,368.00		96
Price per Port				\$2,910.08		

Xsigo (OVN) Port Costs

Item Name	Description	Qty	Price	Ext. Price	Ports	Ext. Ports
F1-15	OVN Director	2	\$37,024.00	\$74,048.00		0
Single port 10GE card	1x10GE card	14	\$10,344.00	\$144,816.00		0
Dual port 8Gb FC card	2x8G FC card	16	\$11,947.00	\$191,152.00		0
36 port Sun IB QDR Switch	36x40Gb QDR switch	4	\$15,030.00	\$60,120.00	36	144
QDR SFP+ 3m cable		160	\$75.00	\$12,000.00		0
62.5 LC/LC 3m cable		46	\$18.00	\$828.00		0
Total				\$482,964.00		144
Price per Port				\$3,353.92		

APPENDIX J: OPEX DATA

I/O Power Data

	HP Traditional	HP 10G	HP Xsigo	Cisco UCS
FC HBAs	60.8	60.8		
Ethernet NICs	280.0	344.0		
Converged Adapters			171.2	288.0
FC Switches	56.5	56.5	6.9	6.9
Ethernet Switches	723.7	803.6	74.3	74.3
Converged Switches			268.4	621.9
Overhead (cooling, power loss, etc)	1,120.9	1,264.9	520.7	991.0
Total	2,241.9	2,529.8	1,041.5	1,982.1