Performance Analysis of Virtualised Head Nodes Utilising Cost-Effective Network Attached Storage

A. P. Gerdelan, M. J. Johnson, and C. H. Messom

Institute of Information and
Mathematical Sciences
Massey University, Albany, New Zealand
Email: {A.Gerdelan | M.J.Johnson | C.H.Messom}@massey.ac.nz

In modern systems local disk I/O has significantly lower bandwidth than network I/O. This has lead to the development of Storage Area Networks (SANs) often implemented with expensive, high bandwidth switching fabrics. With the development of Infiniband and 10 Gigabit Ethernet switching fabrics, high bandwidth I/O is becoming commoditised, but as yet is not cost effective as compared to Gigabit Ethernet solutions. This paper analyses the performance of iSCSI and AoE based network attached storage over multiple Gigabit Ethernet channels. After this analysis we consider I/O performance of a virtual machine accessing storage on the SAN. We show that I/O performance is significantly degraded in a virtual machine to the point that virtualisation should be reserved for CPU-intensive jobs rather than I/O intensive ones.

Keywords: Storage Area Network, AoE, iSCSI, Clustering, High Performance Computing, RAID, Virtual Machines.

1 Introduction

Commodity hardware has contributed significantly in the history of grid computing systems, from the original Beowulf clusters based on commodity processing and networking to more recently the high throughput computing based on workstation and desktop resources that have enabled grid-based science.

The advent of faster, higher bandwidth access to remote storage as well as hardware support for machine virtualisation has allowed novel approaches to GRID computing to be adopted. These include the use of virtual machines to house workloads that access data on a remote file server. The virtual machine helps isolate the host machine from potentially malicious applications while the remote file access isolates the local file system. File system virtualisation also allows workload and data to be migrated independently allowing optimal configurations to be constructed [1].

High throughput computing applications (based on Condor etc.) make use of commodity components forming *ad hoc* clusters of workstation and desktop machines. Modern corporate and educational networks make use of Gigabit Ethernet as the *de facto* transport. Commodity server-based arrays of disks can provide cost effective networked attached storage (NAS) in these environments. The key bottleneck in a multi-user environment is the single gigabit Ethernet link from the switch to the NAS. To overcome this bottleneck, this paper investigates the alternative strategies and protocols that can be adopted to significantly increase the throughput bandwidth to the disk array.

One area that presents a performance bottleneck to grid-enabled clusters is storage read/write access over a Storage Area Network (SAN). Our HPC users demand high bandwidth storage and we need to provide the highest possible read/write access to low-cost shared storage.

Other researchers have identified the use of CPU for network packet processing as the cause of a networking bottleneck, and present novel means for dedicating the use of one processor on multi-core machines for this task [2]. An area that has not been thoroughly examined, however, is the comparative efficiency and effectiveness of the various storage export protocols. An analysis of the overall throughput efficiency and effectiveness, and also efficiency in terms of the CPU resource consumption footprint of these storage access protocols would present the architects of high-performance clusters and grid computing networks with a valuable template for further system optimisation.

We have thus investigated and, in this paper compare various software technologies for the export of storage from a SAN:

- iSCSI
- AoE

iSCSI implements the Small Computer Systems Interface (SCSI) protocol over a TCP/IP network and has become the *de facto* standard for software access to disk resources. iSCSI is available as a cost-free, open-source implementation that will run on most systems, in many cases eliminating the need for expensive hardware controllers [3], [4].

AoE is a similar approach which implements the Advanced Technology Attachment (ATA) protocol over an Ethernet network. AoE was developed by Coraid Ltd, a producer of storage hardware solutions and blade devices, and is available for implementation on systems other than those produced by Coraid. Note that we will be analysing the performance of the software AoE implementation with virtual blades, and not the proprietary hardware blades.

Both AoE and iSCSI can export storage block devices over Ethernet, which are encapsulated by the protocol to mimic physical hard disks on the machine mounting the storage; SCSI disks in the case of iSCSI and ATA disks in the case of AoE. The machine exporting the storage is called the *target* and machine remotely accessing the storage is the *initiator*. The initiator can treat remote storage devices as if they were local attached disks and create file systems or combine them using software RAID [5].

It has been previously ascertained that the performance of iSCSI depends on the balance of various componential system resources [6] in order to accommodate different types of network traffic. We have therefore collected performance results for a range of different network operations including character and block operations, random seeks, and CPU utilisation per unit throughput, in order to determine if our system configuration has a different effect on these different properties of performance with iSCSI that it does with the AoE protocol.

We did not consider NFS in our comparison since other studies have shown that iSCSI-based file access provides higher performance than NFS-based file access [7], [8], and current implementations of NFS scale poorly and it thus provides poor performance when used concurrently by many clients. For concurrent access to shared storage a cluster file system such as GFS, OCFS or Lustre must be used. For simplicity however we tested using XFS and a single initiator.

For our tests we used bonnie++ to measure file system performance on the initiator, with problem sizes greater than twice the total size of available RAM to avoid caching effects, and also used netperf to test the available bandwidth. We have also used custom written scripts to find raw block transfer rates. Each of our tests comprises 5 runs, which are repeatable as a set of 4 results within 10 %, excluding 1 spurrious outlying result.

Virtual machines are being increasingly adopted, particularly in grid environments, as cluster gateways, network servers, and storage head nodes, where separate physical machines would normally be delegated for these rôles. This approach can drastically reduce the costs of building a cluster, allowing several virtual machines to be hosted simultaneously on one physical server machine; a realistic scenario with the growing availability of multi-core server machines.

Xen virtual machines are reported to operate at 90-100% of the host machine's CPU capacity, under various test conditions [9]. Whilst the virtual machine approach is becoming widely adopted, there is very little information available on the I/O performance hit that it incurs. We have therefore investigated the performance of storage access throughput for a typical virtual machine mounted on our storage configuration, and in this paper also analyse the performance this approach compared to storage mounted directly on a physical machine.

2 Conceptual Comparison of iSCSI and AoE

Fig. 1 shows the protocol stacks for AoE and iSCSI. AoE requires a much smaller stack than iSCSI. The TCP/IP layer is eliminated altogether by AoE, which uses an Ethernet-level protocol.

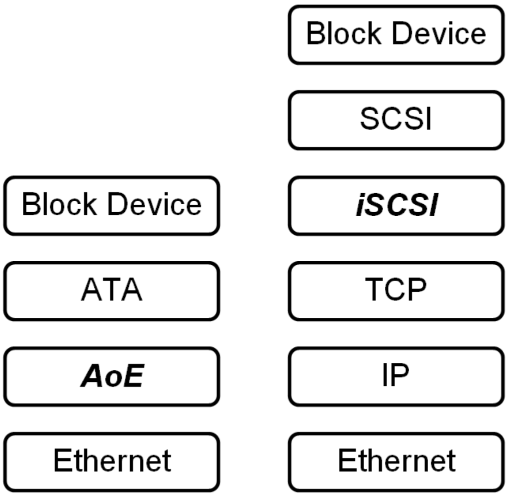

Figure 1: Comparison of AoE and iSCSI Protocol Stacks.

Depending on the efficiency of the AoE protocol, AoE should be faster than iSCSI as it does not incur the overhead of TCP/IP. The speed increase of AoE is at the expense of functionality however; because of its use of the TCP/IP stack iSCSI is routable over a network, AoE is not. Cluster storage will almost never need to be routed and where it does NFS can be used alongside AoE.

If the storage does not need to be routed, as in the case of our Storage Area Network, then comparative performance is the key defining difference. For this reason we present here our performance evaluation of AoE and comparative evaluation of iSCSI.

3 System Configuration

3.1 Equipment Details

The storage system we are using is based on 4 Storage Area Network target machines each containing 16 SATAII 7200 RPM hard disks of 500 Gigabytes capacity. Each SAN machine therefore provides a raw capacity of 8 Terabytes before RAID and all 4 SAN machines have a combined gross capacity of 32 Terabytes. The SAN targets each use a single Dual core XEON 5130 CPUs running at 2 GHz and run CentOS 4.4 x86_64 Linux with kernel version 2.6.20.

3.2 Channel Bonding

Each of the SAN machines is equipped with 4 Gigabit Ethernet adapters. In order to take full advantage of the network capability, we have investigated the use of channel bonding, which would introduce redundancy to overcome any Ethernet link failures, and potentially combine Ethernet channels into a single conceptual link.

Using netperf, with one bond of two MTU 9000 Ethernet channels between our storage and server machines (using one half of the available Ethernet channels), we were able to produce storage to head node traffic at up to 1575 Megabits per second, at up to 26% usage of SAN machine CPU, and head node to storage traffic at up to 1570 Megabits per second, using up to 15% of head node CPU.

With 2 bonds of two MTU 9000 channels each between the server and storage machines (using all of the available channels), operating simultaneously, we found that the SAN machines were using up to 44% of CPU, and up to 19% of CPU on the head node. From storage machine to head node we were able to produce a combined peak of 3092 Megabits per second, and from head node to storage machine a peak of 3536 Megabits per second.

Because channel bonding is able to facilitate use of just over 3/4 of the available bandwidth, we have decided not to use channel bonding, but rather have designed our SAN machines to make use of all four Ethernet channels separately and independently, which we have shown to operate at no less than 95% of available bandwidth.

3.3 Design of Gigabit SAN Access

We have divided the physical hard disks in each SAN machine into 4 logical groups of 4 disks each (see Fig. 2). Each of these logical groups is a block device on the target, created using software RAID-5, which provides a redundancy of 1 out of every 4 disks. We dedicated one of our Ethernet channels (labeled eth0-eth3 in Fig. 2) to each of our block devices; each channel is allocated a separate IP subnet to enforce this separation. For performance analysis and comparison, we exported the block devices from the SAN machines using both AoE vblade version 14, and iSCSI target version 0.4.14. The Server machine in our cluster mounts the exported storage, and has a compatible set of 4 gigabit Ethernet interfaces.

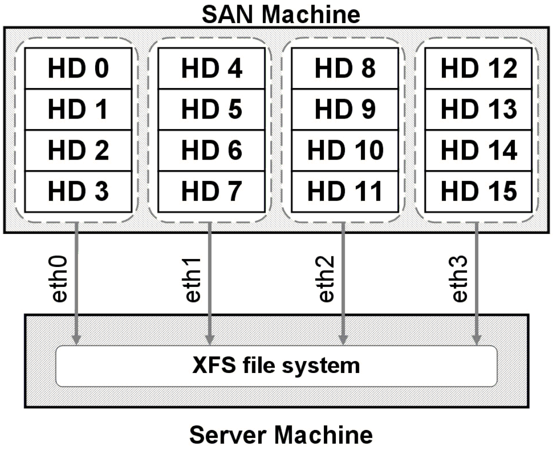

Figure 2: Storage Export Design.

In the case of iSCSI or AoE, the Server discovers or initiates each of the exported or exposed block devices using the appropriate software; aoetools version 14, and open-iSCSI initiator version 2.0-754, respectively, and combines the 4 smaller block devices into one large block device using software RAID-0. The Server creates the XFS file system on the combined block device, and mounts it on the local system.

We analysed the network performance of this system configuration by running concurrent versions of the netperf benchmarking utility. We found that the combined Ethernet interfaces could sustain a concurrent bandwidth of 960 Megabits per second i.e. 3.8 Gigabits/s or 480 Megabytes per second.

3.4 Jumbo Frames

Jumbo Frames are network packets larger than the default 1500 bytes. Jumbo Frames must be supported by the network device drivers and the network switch being used. Since larger frames lead to fewer interrupts at the device layer they should provide a performance increase to both iSCSI and AoE. We measured the performance of both AoE and iSCSI targets with and without 9000 byte Jumbo frames.

4 Performance Results

4.1 iSCSI Performance

We are primarily interested in the block read and write speeds over our SANs since other measures are more dependent on the file system. At a Maximum Transmission Unit (MTU) of 1500 bytes per frame, iSCSI produced read and write speeds up to 194 MB/s. With Jumbo frames of MTU 9000 bytes per frame we were able to produce block read and write speeds of up to 227 MB/s, a 17% improvement. Figs. 3 and 4 show comparative performance results from iSCSI and AoE for character and block operations at MTU 1500 and 9000, respectively.

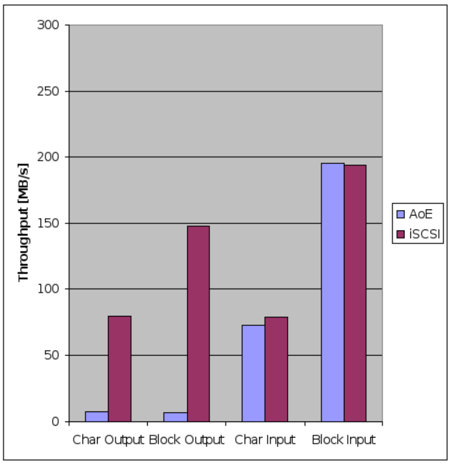

Figure 3: Performance Peaks for iSCSI and AoE at MTU 1500.

Random access to data is also a key performance indicator for Storage Area Networks. We found that our iSCSI configuration allowed us to perform 405.2 random seeks per second at a MTU of 1500 bytes per frame, and 436.2 random seeks per second at 9000 bytes per frame, a 7.6% improvement with Jumbo frames. Character read and write speeds are of some interest also, which we found to peak at 80 MB/s with a 1500 byte MTU. Character I/O appears to be limited by the OS/File system layer because using Jumbo frames did not increase the performance of character reads and writes. These peaked at 77 MB/s with the MTU set to 9000 bytes per frame. CPU usage by iSCSI during block operations was minimal; up to 14% without Jumbo frames, and up to 30% with MTU set to 9000 bytes per frame. CPU usage during character operations was close to 100%, which also indicates that the protocol is not the bottleneck in these cases.

4.2 AoE Performance

Interestingly, the AoE software did not operate correctly at smaller MTU sizes. At the standard MTU of 1500 bytes per frame, we were only able to produce block and character write speeds of 6-10 MB/s under AoE, when we would expect 100 MB/s read and write speeds with our hardware. Fig. 3 illustrates the unusually poor performance of AoE write operations at MTU 1500. Sequential character and block read speeds were, however, comparable to those of iSCSI at MTU size 1500 bytes per frame; 77 MB/s and 195 MB/s, respectively.

The low performance of AoE output with the MTU set to less than 9000 bytes per frame was puzzling. At an intermediate setting; MTU 3000 bytes per frame, AoE still produced poor results; only 20 MB/s. Further research may be required to determine if a different system configuration is necessary to produce better MTU 1500 write results with AoE.

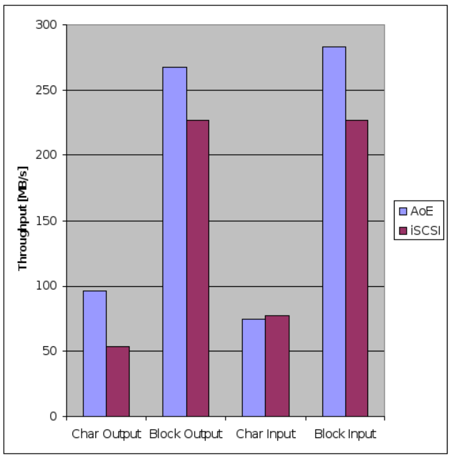

Figure 4: Performance Peaks for iSCSI and AoE at MTU 9000.

With reference to Fig. 4, performance results with MTU set to 9000 bytes per frame were very good, in relation to those produced by iSCSI. Block read and write speeds for AoE peaked at 283 MB/s. Character read and write speeds peaked at 96 MB/s. CPU usage by AoE during block operations was up to 23% for block operations at MTU set to 1500 bytes per frame, and at the same level, 23%, with MTU set to 9000 bytes per frame. CPU Usage during character operations was close to 100% for all results; AoE was clearly reaching a peak of its possible performance for character I/O.

5 Comparative Analysis

5.1 Sequential Block Operations

Considering the best performance scenario of sequential block operations using Jumbo frames, we can say that AoE has a significant performance advantage over iSCSI of 25%. This is clearly illustrated in Fig. 4. It must be noted that the block read and write performance of AoE at MTU 9000 fluctuated over a range of 81 MB/s, but the median of results obtained, 256 MB/s, still exceeded the peak result produced by iSCSI; 227 MB/s.

With reference to Fig. 3, we can see that there is very little difference in block input and character input performance between iSCSI and AoE at the standard 1500 bytes per frame MTU size. Aside from the output problems with AoE at low MTU sizes, there is no significant difference in throughput between AoE and iSCSI at this frame size.

5.2 CPU and Protocol Efficiency

In order to determine if the AoE protocol stack does in fact, make more efficient use of CPU resources than iSCSI by reducing the overhead incurred by TCP/IP, we can divide our best block access speeds, for each protocol, by the percentage of CPU that was occupied during these operations. Tables 1 and 2 present the result as a Throughput per CPU Utilisation measure.

Table 1: CPU Usage Efficiency Comparison at MTU 1500

Protocol	**Peak Block Throughput [MB/s]**	**CPU %**	**Throughput per CPU Utilisation**
iSCSI	194	14	14
AoE	195	23	8.5

Table 2: CPU Usage Efficiency Comparison at MTU 9000

Protocol	**Peak Block Throughput [MB/s]**	**CPU %**	**Throughput per CPU Utilisation**
iSCSI	227	30	7.5
AoE	283	23	12

From Tables 1 and 2, we can see that in the case of MTU set to 9000 bytes per frame AoE not only enables faster storage access than iSCSI, but also requires lower CPU usage, proving that AoE can be a more efficient and more effective mechanism than iSCSI for block operations when Jumbo frames are enabled. At at standard MTU of 1500 bytes per frame, however, we observe that AoE is significantly less efficient than iSCSI.

5.3 Random Seeks

With reference to Fig. 5, we can see that, where MTU is set to either 1500 or 9000 bytes per frame, iSCSI is faster than any result produced by AoE.

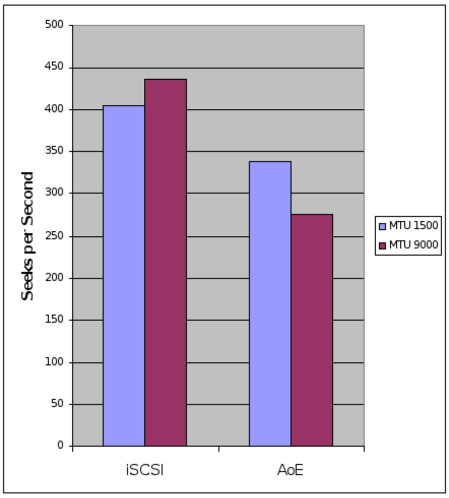

Figure 5: Random Seek Performance Peaks for iSCSI and AoE.

5.4 Comparison of Protocols to Theoretical Maximum System Performance

The theoretical maximum storage access speed of our system can be calculated in separate components; storage, network, and CPU, and then given as the maximum speed of the slowest component. The maximum access speed on the SAN machine is the sum of individual maximum disk speeds minus redundancy due to RAID. This is approximately 70 MB/s multiplied by 4 disks, minus one disk due to RAID-5 redundancy. We then have 4 groups of these 210 MB/s software RAID blocks, which are accessed simultaneously due to a top-level RAID-0 stripe. This gives us a maximum of approximately 840 MB/s storage access speed for our SAN machines.

Network bandwidth is across 4 Gigabit Ethernet connections, which gives us a theoretical maximum of about 400 Megabytes per second. Providing therefore, that the CPU is not fully utilised, then we could expect an ideal storage access protocol to approach a throughput of 400 MB/s.

Given this, we can say that iSCSI is, at peak performance in our system, operating at 56% throughput efficiency, and that AoE is, at peak performance in our system, operating at 70% throughput efficiency.

6 Virtual Machines

6.1 Virtual Machine Configuration

In order to support a virtual machine on mounted storage we have used 4 RAID-5 blocks from one of our SAN machines as iSCSI targets over individual Gigabit Ethernet channels. As illustrated in Fig. 6, these have then been striped together with a RAID-0 block on the head node. This configuration is identical to the best-performance MTU 1500 iSCSI configuration that we identify in this paper, as this is the most typical of the various system configurations that we have analysed. We have created a physical volume on the RAID-5 block, and subsequently also a volume group. This allows us to create logical volumes on the block of realistic partition sizes, allocating only a 20 GB segment of the 8 TB block for the virtual machine.

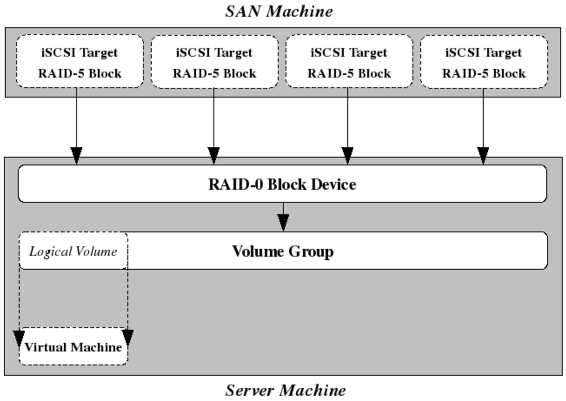

Figure 6: Storage-Mounted Virtual Machine Design.

Using a Xen version 1, x86_64 kernel 2.6.20-1.2948 under Fedora Core 6 as a Domain0 hypervisor host machine on Intel virtualisation-supported hardware, we have created an Linux x86_64 kernel 2.6.9-42 Centos 4.4 machine on the 20 GB logical partition with a typical Linux partitioning scheme. This consists of another volume group with another logical volume containing a root and a swap partition, and a separate boot partition. We allocated the virtual machine only a single core for processing, and 1 GB of RAM.

Storage access from the root mount of the virtual machine then must transition through 4 logical layers more than the equivalent storage mounted directly on the head node, is using an ext3 file system, rather than XFS, and has only a subset of the head node's system resources to process storage access.

6.2 Storage Access Throughput

Preliminary analsis of storage throughput using bonnie++ on the virtual machine produced some interesting results. Virtualised tests were run on DomainU. Output results fluctuated almost randomly, as an unavoidable amount of memory buffering occurred, producing unreliable results. Excluding these extreme results, character input results for the virtual machine peaked at 49 MB/s, 75% of the same operation on the host machine at MTU set to 1500 bytes per second, and block input throughput peaked at 50 MB/s, 25% of maximum block input performance on the host machine at MTU 1500. At most, data access operations used no more than 40% of the one virtual CPU allocated to the virtual machine.

Table 3: CPU Usage Efficiency Comparison of Physical and Virtual Head Node

Head Node	**Peak Block Throughput [MB/s]**	**CPU %**	**Throughput per CPU Utilisation**
Domain0 Physical Host	192	14	14
DomU Virtual Machine	50	40	1.25

Using custom-built scripts to overcome buffering of test data in system RAM, we have written and read randomly sized chunks of data to storage through the virtual machine. On the virtual machine we were able to produce data throughput rates of up to 76.2 MB/s, which is 54% of the maximum throughput achieved from the same test scripts, with the storage mounted directly to the head node machine. If we examine Table 3, we can see that the virtual machine has lower overall throughput, more than double the CPU usage, and is less than one tenth as efficient as mounting directly to the physical host.

7 Conclusions

Whether or not to use AoE or iSCSI to facilitate storage access for a High Performance Computing environment must be decided based on the nature of network access to storage (SAN or LAN), the type of equipment available, and on the types of data being processed. If the storage is to be routed over a LAN, then certainly the routing flexibility of iSCSI makes this protocol superior to AoE, which does not operate over TCP/IP, but rather at the Ethernet level.

We have observed AoE to perform at the same level, or worse than iSCSI at regular frame sizes, where MTU is set to 1500 bytes per frame. For a software vblade setup equivalent to that which we used, without Jumbo frames, then iSCSI would be a better choice of protocol.

Software AoE is high-performing alternative to iSCSI for those systems where hardware is enabled to allow MTU sizes of 9000 bytes per frame. We have shown AoE to not only outperform iSCSI in terms of throughput, but to use less overall CPU in doing so.

For systems that require the fastest possible random access to storage, the results in our system have shown that iSCSI is the faster protocol.

Virtual machines are a popular option for encapsulation of server functionality, and for the reduction of cost. We have shown that, whilst it is possible to use virtual machines to access network attached storage, the virtual machine approach requires several additional layers of software storage architecture, which significantly reduces the performance of access to storage machines, and can consume a much greater amount of CPU than directly mounted storage.

We have shown that it is possible to produce very high rates of storage access throughput at low-cost, making use of open protocols, and shown that whilst the virtual machine approach may significantly reduce equipment costs and reduce the complexities of administration, virtual machines are still no substitute for a physical machine for those demanding maximum storage access performance.

Acknowledgements

We wish to acknowledge and express our thanks to the BeSTGRID project for providing the funding for this research, and to the Institute of Information and Mathematical Sciences at Massey University, for facilitating our research environment.

References

[1] F. Travostino, P. Daspit, L. Gommans, C. Jog, C. de Laat, J. Mambretti, I. Monga, B. van Oudenaarde, S. Raghunath, and P. Y. Wang., "Seamless live migration of virtual machines over the MAN/WAN," *Future Generation Computer Systems*, vol. Volume 22, no. Issue 8, pp. 901–907, October 2006.

[2] T. Brecht, G. J. Janakiraman, B. Lynn, V. Saletore, and Y. Turner., "Evaluating network processing efficiency with processor partitioning and asynchronous I/O," *CM SIGOPS Operating Systems Review*, vol. Volume 40, no. Issue 4, pp. pp. 265–278, October 2006.

[3] S. Aiken, D. Grunwald, A. Pleszkun, and J.Willeke., "A performance analysis of the iSCSI protocol," in *in Proc. 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies*, April 2003, pp. 123–134.

[4] P. Wang, R. Gilligan, H. Green, and J. Raubitschek., "IP SAN - from iSCSI to IP-addressable Ethernet disks," in *in Proc. 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies*, 2003, pp. 189–193.

[5] X. He, P. Beedanagari, and D. Zhou., "Performance evaluation of distributed iSCSI RAID," in *in Proc. International Workshop on Storage Network Architectureand Parallel I/Os (SNAPI)*, New Orleans, LA, Sept. 2003.

[6] D. Xinidis, A. Bilas, and M. Flouris., "Performance evaluation of commodity iSCSI-based storage systems," in *in Proc. 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies*, April 2005, pp. 261–269.

[7] R. J. Recio, "Applications, Technologies, Architectures, and Protocols for Computer Communication," in *in Proc. ACM SIGCOMM workshop on Network-I/O convergence: experience, lessons, implications*, Karlsruhe, Germany, 2003, pp. 163–178.

[8] L. Yingping and D. Du., "Performance study of iSCSI-based storage subsystems," *IEEE Communications Magazine*, vol. Volume 41, no. Issue 8, pp. 76–82, August 2003.

[9] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield., "Xen and the art of virtualization," in *in Proc. SOSP*, 2003.