Cost-Effective Remote Mirroring Using the iSCSI Protocol

Ming Zhang, Yinan Liu, and Qing (Ken) Yang
Department of Electrical and Computer Engineering
University of Rhode Island
Kingston, RI 02874
{mingz, yinan, qyang}@ele.uri.edu
tel +1-401-874-5880
fax +1-401-782-6422

Abstract

This paper presents a performance study of the iSCSI protocol in the context of remote mirroring. We first integrate our caching technology called DCD (disk caching disk) into a standard iSCSI target device to form a high performance storage system for mirroring purpose. Performance measurements are then carried out using this storage system as well as standard iSCSI targets as mirroring devices. We consider remote mirroring on a LAN (local area network) and on a commercial WAN (wide area network). The workloads used in our measurements include popular benchmarks such as PostMark and IoMeter, and real-world I/O traces. Our measurement results show that iSCSI is a viable approach to cost-effective remote mirroring for organizations that have moderate amount of data changes. In particular, our DCD-enhanced iSCSI target can greatly improve performance of remote mirroring.

1. Introduction

Remote data mirroring has become increasingly important as organizations and businesses depend more and more on digital information [1]. It has been widely deployed in financial industry and other businesses for tolerating failures and disaster recovery. Traditionally, such remote mirroring is done through dedicated SAN (storage area network) with FC (Fiber Channel) connections that are usually very costly in terms of installation and maintenance. A newly emerging protocol for storage networking, iSCSI [2], was recently ratified by the Internet Engineering Task Force [3]. The iSCSI protocol is perceived as a low cost alternative to the FC protocol for remote storage [4][5][6][7]. It allows block level storage data to be transported over the popular TCP/IP network that can cover a wide area across cities and states. Therefore, the iSCSI lends itself naturally to a cost-effective candidate for remote mirroring making use of the available Internet infrastructure.

The viability of iSCSI protocol for remote mirroring depends, to a large extent, on whether acceptable performance can be obtained to replicate data to a remote site. While there are studies reported very recently on the iSCSI performance on LAN networks, campus networks, and emulated WAN [4][5][6][7], the open literature lacks technical data on the performance of the iSCSI protocol for remote mirroring over a realistic commercial WAN. The objective of this paper is two fold. First, we incorporate our new storage architecture to an iSCSI target to enhance write performance specifically for remote mirroring purposes. Secondly, we carry out measurement experiments to study the performance of the iSCSI protocol for remote mirroring on both a LAN and a commercial WAN network.

Our new storage architecture is referred to as DCD (disk caching disk) [8][9]. The idea is to use a log disk, called cache-disk, as an extension of a small NVRAM to cache file changes and to destage cached data to the data disk afterward when the system is idle.

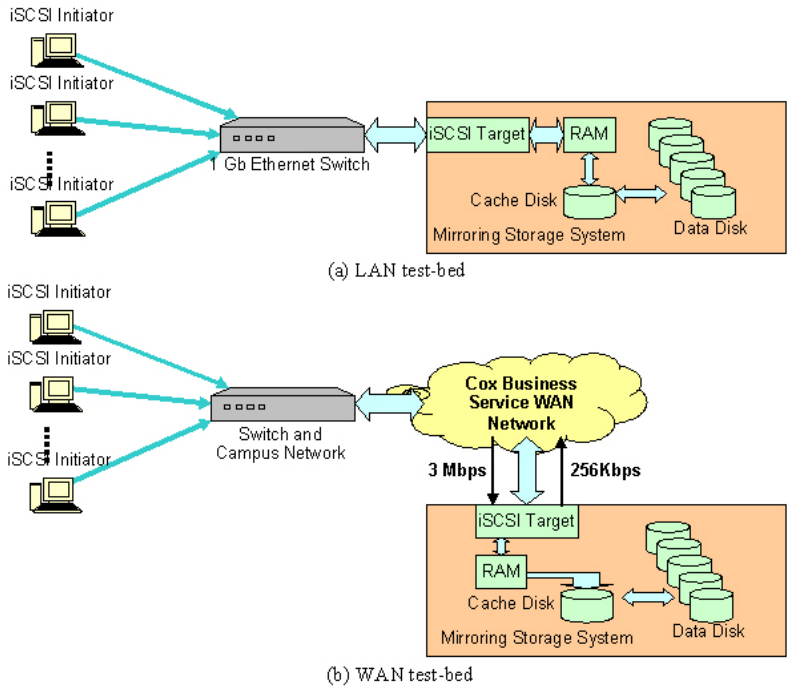

Figure 1. Experimental settings for performance measurements of iSCSI remote mirroring.

Small and random writes from the iSCSI network are first buffered in the small NVRAM buffer. Whenever the cache-disk is idle or the RAM is full, all data in the RAM buffer are written sequentially in one data transfer into the cache-disk. The RAM buffer is then made available quickly to absorb additional requests so that the two-level cache appears to the iSCSI network as a huge RAM with size of a disk. When the data disk is idle, a destage operation is performed, in which the data is transferred from the cache-disk to the normal data disk. Since the cache is a disk, it is cost-effective and highly reliable. In addition, the log disk is only a cache that is transparent to the file system and upper layer applications.

We incorporated our DCD into the iSCSI target program [10] that is used as a storage device for remote mirroring purpose. We carried out measurement experiments in two different settings: one inside our laboratory and the other over a commercial WAN through Cox-business Internet services. Our experiments with the real world commercial WAN give us great insightful experiences that may not be possible using emulated WAN in a laboratory [4][5][6][7]. We measured a variety of benchmarks and real world I/O traces widely used in the file system and storage system communities. Measured results show that the iSCSI protocol is a viable and cost-effective approach for remote mirroring. It is particularly useful to small to medium size organizations to deploy economical remote mirroring for tolerating site failures and disaster recovery when the cost of losing data matters.

2. Experimental Methodology

Figure 1 shows the two experimental setups for our experiments. Our first experiment was carried out inside our laboratory as shown in Figure 1(a). Several server hosts are connected to a mirroring storage system through an Intel NetStructure 470T Gigabit Ethernet switch. The server hosts act as iSCSI initiators while the mirroring storage system acts as an iSCSI target. Our second experimental setting is over a realistic commercial WAN through Cox Business Services. The iSCSI initiators in the LAN inside our laboratory are connected through our campus network and leased lines to the educational Internet. They are then connected to a website of a business office, Elake Data Systems, Inc., on the Cox Communications Inc.

cable network. The business office is used as a remote mirroring site that is located in a different town several miles away from our university campus. The down stream speed to the mirroring site is theoretically 3 Mbps and the upstream speed is theoretically 256 Kbps. Because of sharing of cables, the actual speed varies depending on network traffic and the time of a day. We found that during our experiments that the actual speeds vary from 40.7 KBps to 294 KBps. The cost of such a business connection is less than $100/month in New England area. We believe that such a connection and its cost represent a typical network connection for small to medium size businesses that mirror moderate amount of their business data at a remote site for failure tolerance and disaster recovery. As indicated in [1], the cost of leasing a WAN connection with speed of 155Mbps could cost about $460,000/year. Our objective here is to analyze the backup performance of the iSCSI protocol over an inexpensive WAN network where the iSCSI protocol is likely to be used for cost effectiveness.

All machines used in the experiments are Dell servers equipped with a single Pentium III 866MHz CPU, 512MB SDRAM, and the Intel Pro1000T Gigabit NIC (network interface card). We run Redhat 9 as the operating system with recompiled Linux kernel 2.4.20. Our iSCSI implementation is based on the implementation ref20_19b from University of New Hampshire [10]. The SCSI controllers we used are Adaptec AIC7899 Ultra 160 controller. All SCSI disks we used in our experiments are 18GB Seagate ST318452LW Ultra 160 disks. When two disks need to be connected to the same SCSI controller, we connect them to different channels.

At the iSCSI target, our DCD system is integrated with the iSCSI target software. Random and small write requests coming from the network are first buffered in the 32MB NVRAM buffer and the target acknowledges immediately to the server host for write completion. These small write requests are collected to form a log to be moved sequentially to the cache disk as soon as the cache disk is idle or the data in the NVRAM exceeds a predetermined watermark. As a result, there is always a room in the NVRAM buffer to accept new requests and the two-level hierarchy consisting of the RAM buffer and the cache disk appears to the network as a large RAM absorbing write requests quickly. In our current implementation, we use part of the host memory to emulate the NVRAM. Our previous experiments have shown that the maximum time before the data in the NVRAM buffer are moved to a disk is usually less than 100 milliseconds, which guarantees the safety of mirrored data even a DRAM buffer is used instead of the NVRAM provided that a UPS is used. Destaging operations between the cache disk and the data disk are done when the target storage system is idle.

Our remote mirroring software is based on the RAID1 code in the Linux kernel. There are two devices in a typical mirroring configuration, one is a primary device and the other is a secondary device. In a production implementation, there might be one or more spare devices available. Since we are interested in the performance of remote mirroring, we only consider the two-device configuration here. The mirroring software exports a block device to the operating system and applications that is very similar to the normal RAID1 device such as /dev/md0 or /dev/mdx. All read requests are sent to the primary device only while all write requests are sent to both devices. A write request sent from a upper layer is acknowledged as being finished only after the mirroring software receives acknowledgments from both devices. Therefore, our mirroring software falls into the category of ”lock-step” or ”synchronous” mode as defined by Ji et *al* [1]. The following is a list of hardware configurations that our experiments are based on:

- A primary SCSI disk with another SCSI disk as the mirroring device (S-S). We use a local SCSI disk to mirror another SCSI disk in the same system. This is a baseline configuration as a reference for comparison purpose.
- A primary SCSI disk with an iSCSI target device on a LAN as the mirror device (S-iL). In this configuration, we use an iSCSI target device on a LAN to mirror a local SCSI disk.
- A primary SCSI disk with an iSCSI target device on a WAN as the mirror device (S-iW). In this configuration, we use an iSCSI target device on a WAN to mirror a local SCSI disk.
- A primary SCSI disk with a DCD-enhanced iSCSI device as the mirroring device (S-iD). This

	TPC-C20	Financial-1	Financial-2
Number of requests	10,000,000	2,452,167	2,733,121
Number of write requests	2,965,750	1,502,641	480,529
Request size range	4096B-126,976B	1024B-17,116,160B	1024B-262,656B
Mean Request Size	47,063B	3,855B	2,508B
Write Size Range	4096B-126,976B	1024B-17,116,160B	1024B-262,656B
Mean Write Size	4,710B	4,838B	3,107B
Request per second	192	57	67
Write per second	57	35	12

Table 1. Characteristics of the traces used in our experiments

configuration incorporates our DCD technology into the iSCSI target device for mirroring purpose. Similarly, such DCD-enhanced iSCSI storage can be either located on the LAN designated as S-iDL, or located on the WAN referred to as S-iDW.

The workloads used in our experiments consist of popular storage benchmarks such as PostMark [11], IoMeter [12], and real world traces. PostMark [11] is a widely used [13][14] file system benchmark tool from Network Appliance, Inc.. It measures performance in terms of transaction rates in an ephemeral small-file environment by creating a large pool of continually changing files. Once the pool has been created, a specified number of transactions occur. Each transaction consists of a pair of smaller transactions, i.e. Create file or Delete file and Read file or Append file. Each transaction's type and files it affected are chosen randomly. The read and write block size can be tuned. On completion of each run, a report is generated showing some metrics such as elapsed time, transaction rate, total number of files created and so on.

The IoMeter is another highly flexible and configurable synthetic benchmark tool that is also widely used in various research works. IoMeter can be used to measure the performance of a mounted file system or a block device. For a mounted file system, it generates a large size file as the workplace and performs various configurable operations. For a block device, for example, a SCSI disk /dev/sda1, a RAID or mirroring device /dev/md0, IoMeter treats it as a normal file and directly reads or writes on it after opening it.

Besides the above benchmarks, real world traces are also used in our experiments. The first trace is TPC-C20 that is a block level trace downloaded from the Performance Evaluation laboratory at Brigham Young University. They had run TPC-C benchmark with 20 data warehouses using Postgres database on Redhat Linux 7.1 and collected the trace using their kernel level disk trace tool, DTB [15]. The other two traces are I/O traces from OLTP applications running at two large financial institutions, Financial-1 and Financial-2. They represent typical workloads of financial industries and are made available by Storage Performance Council in partnership with the University of Massachusetts who together are hosting a repository of I/O traces for use in the public domain [16]. Table 1 shows the characteristics of the traces that we use.

3. Results and Discussions

3.1. PostMark Results

Our first experiment is to measure the mirroring performances of PostMark benchmark. Figure 2 shows our measured times for finishing 100,000 transactions on 10,000 files of the PostMark benchmark. The total amount of data generated by the PostMark is about 700MB. We varied the proportion of write requests of all transactions in the benchmark between 50% and 30% and plotted them separately as shown in Figure 2. Let us first consider mirroring on the LAN network. As shown in Figure 2(a), it takes longer time to finish all the transactions when mirroring data using iSCSI target than mirroring data using a local

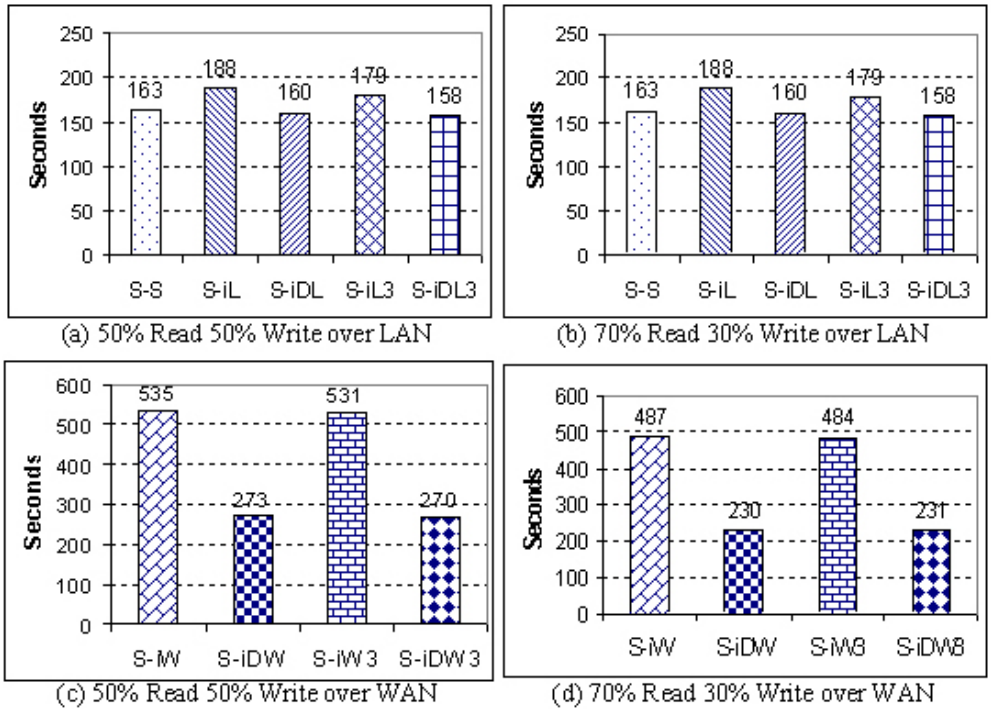

Figure 2. PostMark results: total time to finish 100,000 transactions on 10,000 files

Figure Legend for Figure-2 through Figure-4:
S-S: SCSI disk and SCSI disk mirroring.
S-iL: SCSI disk and iSCSI disk mirroring over a LAN.
S-iDL: SCSI disk and DCD-enhanced iSCSI disk mirroring over a LAN.
S-iL3: SCSI disk and iSCSI disk mirroring over a LAN with 3 parallel iSCSI connections.
S-iDL3: SCSI disk and DCD-enhanced iSCSI disk mirroring over a LAN with 3 parallel iSCSI connections.
S-iW: SCSI disk and iSCSI disk mirroring over a WAN.
S-iDW: SCSI disk and DCD-enhanced iSCSI disk mirroring over a WAN.
S-iW3: SCSI disk and iSCSI disk mirroring over a WAN with 3 parallel iSCSI connections.
S-iDW3: SCSI disk and DCD-enhanced iSCSI disk mirroring over a WAN with 3 parallel iSCSI connections.

SCSI disk. However, the time difference is not as significant as we initially expected, about 16% longer than that of local mirroring. It is interesting to observe that remote mirroring using the DCD-enhanced iSCSI disk takes shorter time than local disk mirroring. This is because DCD hides many seek times and rotation latencies of small writes by combining them into large logs to the cache disk, while with local disk mirroring, every write operation has to wait for two disk writes both requiring seek times and rotation latencies. Similar performance trends were observed for smaller write proportion as shown in Figure 2(b). While the total transaction times are shorter with 30% of writes because only write operations go to remote storage for mirroring, the relative difference between local mirroring and iSCSI mirroring keeps almost the same, about 15%.

iSCSI standard suggests that parallel iSCSI connections in a session may help improving its performance. To observe the effect of parallel connections on iSCSI performance, we measured transaction times with three concurrent connections as shown in the bars graphs marked with suffix 3. From our experiments, it seems that parallel connections do not show significant advantages over single connection. We believe that there could be two possible reasons that lead to the similar performances between parallel connections and single connection. One is the specific iSCSI implementation of UNH and the other is the low traffic intensity of PostMark. It remains open whether and how iSCSI can benefit from parallel and concurrent connections.

PostMark results for iSCSI mirroring over the WAN are shown in Figure 2(c) and Figure 2(d) for 50% writes and 30% writes, respectively. Clearly, synchronously mirroring data over the WAN increases the total transaction time dramatically. Compared to local mirroring, the total time to finish the same 100,000 transactions is more than tripled, from 189s to 535s. To gain some insight to why it takes such a long time to mirror over the WAN, we measured the RTT (round trip time) between our initiators and the target. While RTT fluctuates from time to time, we found the average RTT value to be around 14 ms. This round trip delay is on the same order as a disk operation including seek time, rotation latency and transfer time. For each write operation issued by the benchmark, it has to wait for the mirroring write that has to experience the RTT and a disk operation. As a result, the total transaction time is increased dramatically. The good news is that our DCD technology can help greatly. As shown in the figure, using DCD-enhanced iSCSI target for mirroring over the WAN, the total transaction time is reduced by half from that of pure iSCSI target. This improvement can be attributed to the effective caching of the DCD technology. Compared to the local disk mirroring, the DCD-enhanced iSCSI mirroring over the WAN shows about 40% increase in terms of total transaction time for pure synchronous/lock-step mode. Compared to iSCSI mirroring over a LAN, the difference is about 24%. We believe this is quite acceptable since this mirroring mode will essentially never lose data. Of course, one can allow some degree of asynchrony to obtain better performance. It would be interesting to compare our results here with the traditional FC-SAN mirroring, which we were not able to do because of lack of such FC-SAN facility. For smaller percentage of write operations as shown in Figure 2(d), similar relative performances were observed though the absolute transaction times are shorter because of smaller number of writes.

3.2. IoMeter Results

Our second experiment is to measure the mirroring performances of various configurations using IoMeter benchmark. We configured the IoMeter to generate two types of synthetic workloads, one is 100% writes and the other is 50% writes and 50% reads. Both workloads use random addresses with fixed block size of 4 KB. To minimize the truncation effect that will be explained shortly, we set duration of measurement for each point to 1 hour and reported the average write response time and the maximum response time for each configuration as shown in Figure 3.

Comparing the performance of the local mirroring with that of iSCSI mirroring on a LAN shown in Figure 3(a), IoMeter showed a much larger difference than PostMark did. We believe such a large difference in terms of average response time is the result of higher traffic intensity of the IoMeter benchmark. With 100% writes, the IoMeter continuously generates write requests one after another to both pri-

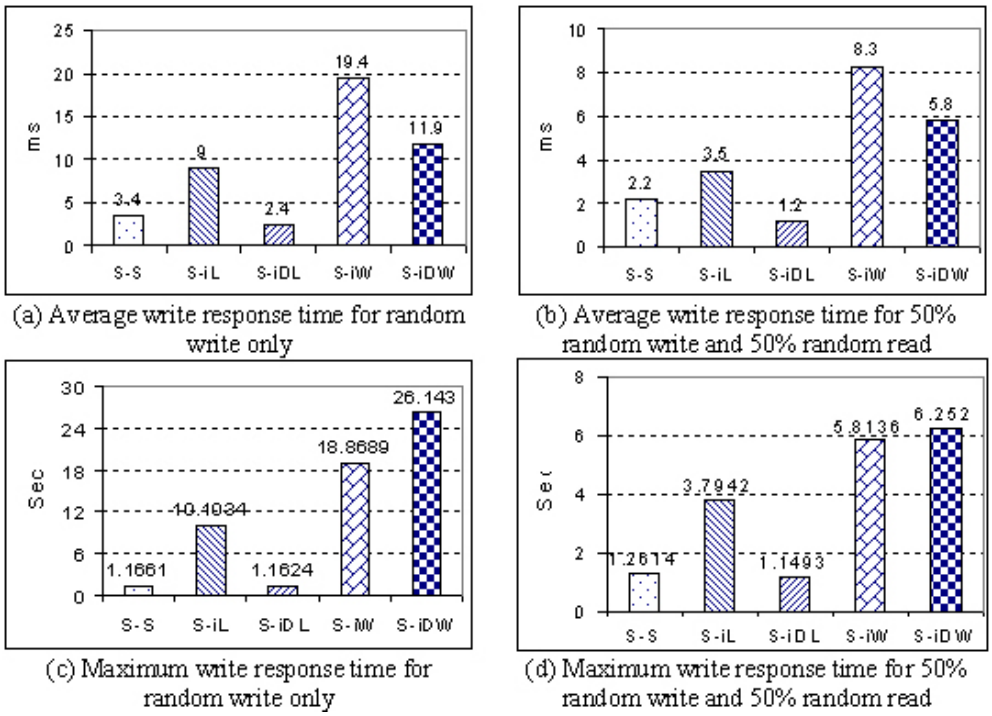

Figure 3. Testing results of IoMeter.

mary disk and the mirroring disk. As a result, it creates a lot of traffic over the network for mirroring data. One may argue that IoMeter generates requests back to back implying that it does not generate a new request until the previous request is acknowledged. However, because IoMeter uses asynchronous writes, it receives an acknowledgment as soon as the write is done in the file system cache. A queue of write requests may be formed giving rise to multiple write requests outstanding on the network. Such a queuing effect increases response time very rapidly. It is also this queuing effect that makes the performance improvement of the DCD-enhanced iSCSI mirroring more pronounced, almost 4 times improvement as shown in Figure 3(a). The queuing effect is substantially reduced if we decrease the write ratio from 100% to 50% as shown in Figure 3(b). In this case, we observed a smaller relative difference between local mirroring and iSCSI mirroring, and between iSCSI mirroring and DCD-enhanced iSCSI mirroring due to reduced write traffic.

For mirroring over the WAN network, the average write response time of using the iSCSI target is 19.4 ms and the response time of using the DCD-enhanced iSCSI target is about 11.9 ms as shown in Figure 3(a). The improvement of the DCD-enhanced iSCSI target over the iSCSI target is about 63%. Putting these results in a different perspective, a computer user would experience 19.4 milliseconds or 11.9 milliseconds delay on average if every write operation were synchronously backed up in a remote site using the iSCSI protocol. Note that these delays correspond to the workload that the user performs write operations continuously one after another. If 50% of continuous disk I/O operations were writes, the average response times would be lower, 8.3 ms and 5.8 ms for the iSCSI target and the DCD-enhanced iSCSI target, respectively as shown in Figure 3(b). Although the average response times are not outrageous, the maximum response times are noticeably large as shown in Figure 3(c) and (d). The maximum response time goes as high as 26 seconds for 100% writes and 6.3 seconds for 50% writes. These high maximum response times suggest that some kind of asynchronous mirroring should be desirable if write traffic is very high. In real world applications, the amount of write opera-

tions is limited and many organizations write less than 3 GB of data per year [17]. We noticed that while the DCD-enhanced iSCSI mirroring shows better average response time, its maximum response time is higher than other configurations. This is because that the target is very busy at that time point and it can hardly find idle time for destage operations. This result is consistent with our previous studies and observations that the DCD is beneficial only when the system can find idle time to carry out destage operations.

In our experiments with the IoMeter benchmark, we found several abnormal phenomena that are hard to explain. One of them is that the DCD-enhanced iSCSI mirroring shows significant lower average response times than the local SCSI disk mirroring. Another phenomenon is the truncation effect as mentioned in the beginning of this subsection. To understand these phenomena, we wrote a micro-benchmark program that continuously performs random writes of 4 KB blocks to a file with the system RAM being set to 256 MB. We measured the total times to finish 50,000 writes and 200,000 writes for three different cases: (1) synchronous writes, (2) asynchronous writes, and (3) asynchronous writes with forced flushing at the end. All the mirroring configurations are in a LAN environment. The asynchronous writes mimic the behavior of IoMeter because IoMeter records time stamp for each request individually and reports performance statistics of all finished transactions at the end of each test run. At the end of each test run, there may be writes done in the cache but no yet written into disks. The asynchronous writes with forced flushing at the end will ensure that all write requests generated in a test run are actually written into disks. As a result, the timing difference between the asynchronous writes and asynchronous writes with forced flushing is the truncation effect. Tables 2 and 3 show the measured results.

	S-S	S-iDL	S-iL
SYNC	273	273	649
ASYNC	161	153	208
ASYNC+Flush	171	155	515

Table 2. Total time in seconds to finish 50,000 transactions by the microbenchmark.

	S-S	S-iDL	S-iL
SYNC	1075	1075	2702
ASYNC	666	623	1666
ASYNC+Flush	677	671	2000

Table 3. Total time in seconds to finish 200,000 transactions by the microbenchmark.

For synchronous writes, we can see that both local mirroring and DCD-enhanced iSCSI mirroring have the same performance which is bounded by the slowest device that we believe is the primary disk. The iSCSI mirroring, however, takes longer time because the iSCSI target may be slower than the primary disk some times during the experiment. For asynchronous writes, the DCD-enhanced iSCSI mirroring shows advantages and even performs better than the local disk mirroring because the performance is no longer bounded by the primary disk because of cache effects. The performance difference increases as the number of transactions increases from 50,000 to 200,000. Similarly, the performance improvement of the DCD-enhanced iSCSI mirroring over iSCSI mirroring also increases as the number of transactions increases from 50,000 to 200,000.

The truncation effects are also shown in the tables as the performance differences between asynchronous and asynchronous with forced flushing. This difference can be as large as 148% as in the case of S-iL column for 50,000 requests. Such truncation effects did show when measuring the IoMeter performance with short measuring time of a few minutes. Therefore, we purposely enlarged the duration of each test run to 1 hour for each point of performance data. As shown in the tables, when we increase the length of test from 50,000 to 200,000 requests, the truncation effect reduced from 148% to 20% for the case of iSCSI mirroring. Similarly, the difference becomes negligible for the local mirroring case for longer test run. However, for some reason that we are not able to explain, the truncation effect increases a little bit (about 7%) for the DCD-enhanced mirroring after increasing the testing time. We believe that it may be the result of the destage process of the DCD system similar to the

phenomenon shown in Figure 3(c,d).

3.3. Traces Results

Figure 4(a) shows the average response times with the three types of the mirroring schemes for the three traces: Financial-1, Financial-2, and TPC-C. Similar performance results to that of the Postmark were observed. If the iSCSI protocol is to be used for remote mirroring, it is important to know the maximum delay caused by iSCSI protocol to determine which mirroring approach to take. We therefore plotted the minimum and maximum response times for three different mirroring schemes as shown in Figure 4(b) and 4(c). It is important to note that the real world workloads are quite different from the benchmarks in that one can always find idle time to take full advantage of the DCD technology. As shown in the figures, the maximum response time of the DCD-enhanced iSCSI is constantly the lowest among the three, implying its smooth and steady performance. The pure iSCSI, on the other hand, takes as much as 2 seconds maximally to mirror a write as shown in the figure. Therefore, it is advisable to use some kind of asynchronous mirroring approaches if an application cannot wait for such a long time.

The performance of remote mirroring over a realistic WAN network is shown in Figure 5. We drew here the response time plots of two traces, Financial-1 and Financial-2, mirroring over the WAN using the DCD-enhanced iSCSI only. As shown in this figure, the response times are noticeably much larger than those in the LAN. The maximum response time is as high as 13 seconds for Financial-1 and 5 seconds for Financial-2. The average response times are 733ms and 405ms for Financial-1 and Financial-2, respectively. However, majority of remote write operations can finish within one second for both cases as shown in this figure.

When converting the traces to SCSI requests, we used 16 parallel threads and each thread generates an SCSI request to the iSCSI initiator according to the time, address and size of one entry of the trace. After a request is issued, the thread waits for the response before generating another SCSI request. Because packet response times fluctuate greatly on a realistic WAN, it may happen that all our 16 threads are busy (blocked) waiting for responses while a new I/O request in the trace is supposed to be issued according to the trace timing. As a result, some write requests in the traces may get skipped until a thread is released. For the experiments reported in Figure 5, we observed that the initiators skipped about 6.55% of writes in the Financial-1 trace and 1.57% in the Financial-2 trace. Note that such skipping would not have happened in real applications but just slowed down the write process. It happened in our experiments because of our applying the traces collected in a fast environment to a slow environment.

To avoid skipping write requests in the traces, we carried out write coalescing at the iSCSI initiator side. Figure 6 shows the response time plots with write coalescing. The write coalescing size is 8 consecutive write requests. That is, each thread collects 8 consecutive writes and issues one write as a batch resulting from coalescing the 8 write operations. With this write coalescing, the 16 threads are able to issue 100% of write requests in both Financial-1 and Financial-2 traces. The response times plotted in Figure 6 correspond to the batch write operations. As shown in Figure 6, the response times for remote mirroring fluctuate but the maximum response times are lower than those in Figure 5. Majority of write mirroring can be done within a half of a second. Our measurement show that about 74.4% of mirroring can be done within one half of a second for financial-1 and about 90% of mirroring can be done within one half of a second for Financial-2. In order word, using the iSCSI protocol with our DCD architecture, one can mirror their business data on transaction-basis to a different town through a very inexpensive Internet connection (less than $100 per month). Mirrored data are safe within 6 seconds in the worst case and over 90% of data can be mirrored safely within half of a second if the transaction rate is as intensive as the Financial-2 trace. For TPC-C traces, we are still experiencing skipped requests of about 1.5% because of high traffic intensity. The response times for TPC-C shown in Figure 6(c) have about 98.5% of write operations of the entire trace.

During our WAN experiments, we also noticed that measured data varies from time to time because of different levels of network contentions. For example, results of daytime measurements may differ from night time and similarly weekdays from weekends.

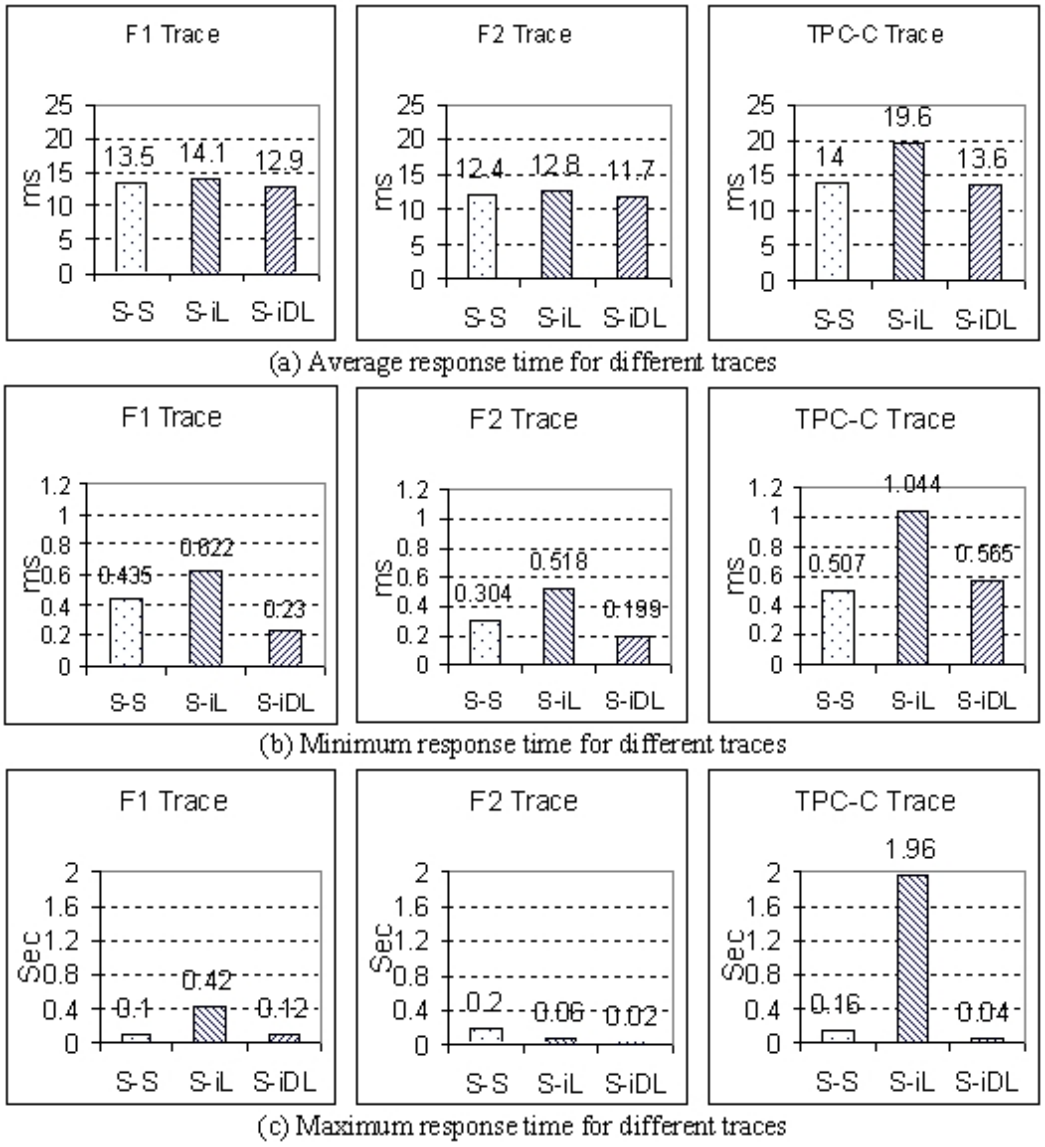

Figure 4. Response time for different mirroring schemes in LAN.

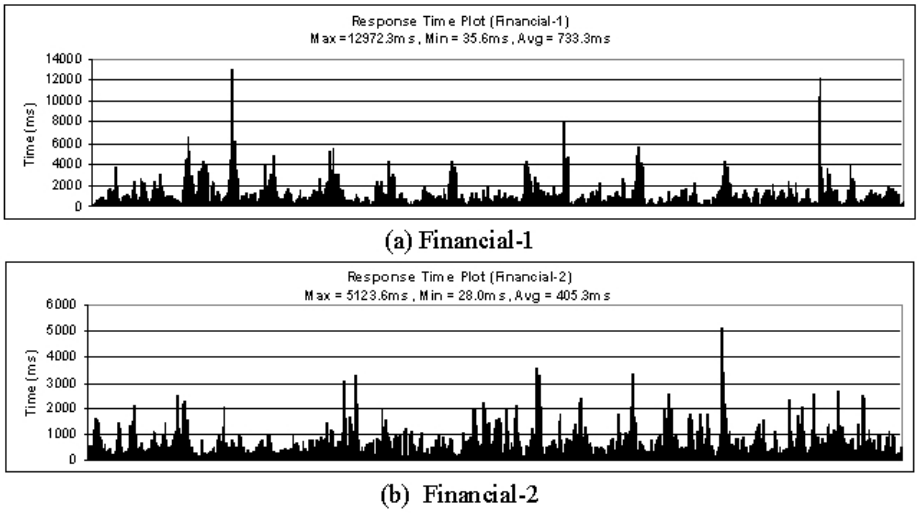

Figure 5. Response time plot of Financial-1 and Financial-2 traces over WAN.

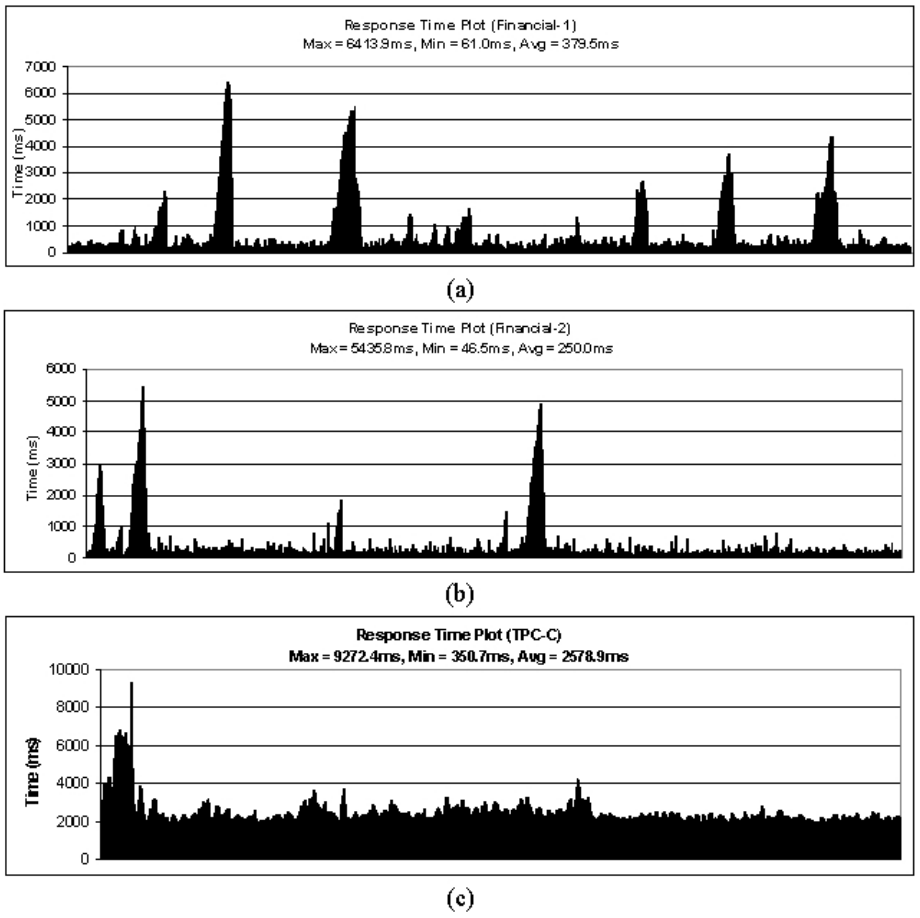

Figure 6. Response time with write coalescing (batch size=8 consecutive writes) over WAN.

4. Related Work

Remote mirroring is not new for data protection and disaster recovery [18]. Companies such as IBM, EMC, Veritas, Computer Associates, and Network Appliance Inc, etc., all provide their own proprietary solutions [19][20][21][22][23]. A good summary of various remote mirroring approaches can be found in [1] including a new asynchronous remote mirroring protocol called Seneca. Myriad [24] uses cross-site checksums instead of direct replication to achieve the same level of disaster tolerance as a typical single mirrored solution requiring less resources. Venti [25] uses a unique hash of a block's contents to act as the block identifier for read and write operations. Thus it enforces a write once policy and can act as a building block for constructing a variety of storage applications with backup and snapshot characteristics.

iSCSI [2] is an emerging IETF standard [3] to provide a mapping for the block level SCSI commands and data over existing TCP/IP networks. Such a technology is supposed to provide a cost-effective alternative to build low cost SAN systems. Meth and Satran [26] discussed some strategies they adopted when designing the iSCSI protocol. Many research works [5][6][7] concentrated on iSCSI performance evaluation in various hardware environments. Most of the WAN performance evaluations are carried out in emulated WAN environments instead of a realistic WAN. Nishan systems (now McData) and other vendors carried out a "Promontory Project" [27] to demonstrate the feasibility of iSCSI in long distance transmission with high speed WAN FC links. Tomonori and Masanori proposed optimization techniques in software iSCSI implementations [28]. A novel cache strategy was proposed to improve the iSCSI performance [4]. It was also proposed to use iSCSI for distributed RAID systems [29]. Many iSCSI software and hardware implementations and products are already available [10][30].

5. Summary and Conclusions

We have carried out measurement experiments to study the viability of using the iSCSI protocol for remote data mirroring for failure tolerance and disaster recovery. To enhance the performance of the iSCSI protocol, a new storage architecture called DCD is incorporated in the iSCSI target software. Measured results show that the DCD-enhanced iSCSI target storage provides smoother and better performance than local disk mirroring in a LAN environment. Experiments on a real commercial WAN show that response times fluctuate and can be very large. Still, data can be mirrored safely at a remote town within a second on average for typical online transaction processing such as Financial-1 and Financial-2 over an inexpensive Internet connection. Our experiments suggest that write coalescing on the initiator side can help in reducing network traffic. Our experience also indicates that measuring performance over a realistic WAN is quite different from an emulated WAN in a laboratory that is more controllable.

Acknowledgment

This research is supported in part by National Science Foundation under grants CCR-0073377 and CCR-0312613. Any opinion, findings and conclusions are those of authors and do not necessarily reflect the views of NSF. We would like to thank our shepherd, Ben Kobler, and the anonymous referees for their valuable comments. The authors would like to thank Elake Data Systems, Inc. (http://www.elakedata.com) for providing the remote site for our experiments.

References

[1] M. Ji, A. Veitch, and J. Wilkes, "Seneca: remote mirroring done write," in *Proceedings of the 2003 USENIX Annual Technical Conference*, San Antonio, TX, June 2003, pp. 253–268.

[2] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, and E. Zeidner, "iSCSI draft standard," http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt.

[3] C. Boulton, "iSCSI becomes official storage standard," http://www.internetnews.com/.

[4] X. He, Q. Yang, and M. Zhang, "Introducing SCSI-To-IP Cache for Storage Area Networks," in *Proceedings of the 2002 International*

Conference on Parallel Processing, Vancouver, Canada, Aug. 2002, pp. 203–210.

[5] W. T. Ng, B. Hillyer, E. Shriver, E. Gabber, and B. Ozden, "Obtaining high performance for storage outsourcing," in *Proceedings of the Conference on File and Storage Technologies (FAST)*, Monterey, CA, Jan. 2002, pp. 145–158.

[6] S. Aiken, D. Grunwald, A. R. Pleszkun, and J. Willeke, "A performance analysis of the iSCSI protocol," in *IEEE Symposium on Mass Storage Systems*, San Diego, CA, Apr. 2003, pp. 123–134.

[7] Y. Lu and D. H. C. Du, "Performance study of iSCSI-based storage subsystems," *IEEE Communication Magazine*, vol. 41, no. 8, Aug. 2003.

[8] Y. Hu and Q. Yang, "DCD—disk caching disk: A new approach for boosting I/O performance," in *Proceedings of the 23rd International Symposium on Computer Architecture*, Philadelphia, Pennsylvania, May 1996, pp. 169–178.

[9] Q. Yang and Y. Hu, "System for destaging data during idle time," U.S. Patent 5 754 888, Sept. 24, 1997.

[10] UNH, "iSCSI reference implementation," http://www.iol.unh.edu/consortiums/iscsi/.

[11] J. Katcher, "PostMark: A new file system benchmark," Network Appliance, Tech. Rep. 3022, 1997.

[12] Intel, "IoMeter, performance analysis tool," http://www.iometer.org/.

[13] K. Magoutis, S. Addetia, A. Fedorova, M. Seltzer, J. Chase, A. Gallatin, R. Kisley, R. Wickremesinghe, and E. Gabber, "Structure and performance of the direct access file system(DAFS)," in *Proceedings of USENIX 2002 Annual Technical Conference*, Monterey, CA, June 2002, pp. 1–14.

[14] J. L. Griffin, J. Schindler, S. W. Schlosser, J. S. Bucy, and G. R. Ganger, "Timing-accurate storage emulation," in *Proceedings of the Conference on File and Storage Technologies (FAST)*, Monterey, CA, Jan. 2002, pp. 75–88.

[15] Performance Evaluation Laboratory, Brigham Young University, "DTB: Linux Disk Trace Buffer," http://traces.byu.edu/new/Tools/.

[16] SPC, "Storage Performance Council I/O traces," http://www.storageperformance.org/.

[17] P. Desmond, "Going the distance for business continuity," http://www.nwfusion.com/supp/2003/ business/1020distance.html, Oct. 2003.

[18] C. Chao, R. English, D. Jacobson, A. Stepanov, and J. Wilkes, "Mime: a high performance parallel storage device with strong recovery guarantees," Hewlett-Packard Laboratories, Tech. Rep. HPL-CSP-92-9 rev1, Nov. 1992.

[19] IBM, "DFSMS SDM Copy Services," http://www.storage.ibm.com/software/sms/sdm/.

[20] EMC, "Symmetrix remote data facility (SRDF)," http://www.emc.com/.

[21] Veritas, "VERITAS storage replicator," http://www.veritas.com.

[22] Computer Associates, "BrightStor ARCserve backup," http://www.ca.com.

[23] Network Appliance Inc., "SnapMirror software: Global data availability and disaster recovery," http://www.netapp.com/.

[24] F. Chang, M. Ji, S.-T. A. Leung, J. MacCormick, S. E. Perl, and L. Zhang, "Myriad: Cost-effective disaster tolerance," in *Proceedings of the Conference on File and Storage Technologies (FAST)*, Monterey, CA, Jan. 2002.

[25] S. Quinlan and S. Dorward, "Venti: a new approach to archival storage," in *Proceedings of the Conference on File and Storage Technologies (FAST)*, Monterey, CA, Jan. 2002.

[26] K. Z. Meth and J. Satran, "Design of the iSCSI protocol," in *IEEE Symposium on Mass Storage Systems*, San Diego, CA, Apr. 2003.

[27] McData, "The Promontory project: Transcontinental IP storage demonstration," http://www.mcdata.com/splash/nishan/.

[28] F. Tomonori and O. Masanori, “Performance optimized software implementation of iSCSI,” in *International Workshop on Storage Network Architecture and Parallel I/Os (SNAPI)*, New Orleans, LA, Sept. 2003.

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

[30] Microsoft, “Microsoft delivers iSCSI support for Windows,” http://www.microsoft.com.