Obtaining High Performance for Storage Outsourcing

Wee Teck Ng and Bruce K. Hillyer
Bell Laboratories
Murray Hill, NJ 07974
{weeteck,bruce}@research.bell-labs.com

ABSTRACT

The viability of storage outsourcing is critically dependent on the access performance of remote storage. We study this issue by measuring the behavior of a broad variety of I/O-intensive benchmarks as they access remote storage over an IP network. We measure the effect of network latencies that correspond to distances ranging from a local neighborhood to halfway across a continent. We then measure the effect of latency-hiding mechanisms. Our results indicate that, in many cases, the adverse effects of network delay can be rendered inconsequential by clever file system and operating system techniques.

1. INTRODUCTION

This paper reports on the results of an initial study that investigates whether a remote-storage architecture is reasonable for data-intensive file system workloads. The goal of our evaluation is twofold. First, we seek to understand the trade-offs between network delays and application performance. Second, we want to understand the impact of file system design and storage system parameters on application performance when using remote storage. This enables us to investigate latency hiding techniques to overcome large network delays. We study these questions by measuring the performance of various benchmarks.

Our experimental platform consists of PCs (700 MHz) running FreeBSD OS 3.4, connected by a fiber-based gigabit Ethernet, with our own SCSI over IP implementation. We performed benchmarking on actual fiber links up to 40 km, giving speed of light propagation delays up to 0.2 ms. We then conducted additional experimentation using a short fiber link plus an accepted kernel tool (the FreeBSD dummynet package) that introduces "network delay" into the protocol stack. In this environment we measured the benchmark performance for delays up to 8 ms, corresponding to 1600 km of fiber. We assume the properties of dedicated network capacity or virtual private networking between the storage service provider and the storage user: we do not investigate performance under the arbitrary network delays of the public internet.

We use two classes of benchmarks. We use microbenchmarks that measure the performance of sequential and random reads and writes to the raw disk device (i.e., a remote disk that is made to look like a local disk by our SCSI over IP software.) We also use a collection of macrobenchmarks that represent a variety of file system workloads. The SSH benchmark [1] unpacks, configures, and builds a medium-sized software package. The SPEC SDET benchmark [2] emulates a timesharing workload in a software development environment via a collection of scripts that invoke various editing and compilation tools. The Surge benchmark [3] emulates a collection of users accessing a web server (we set the parameters to represent 400 clients accessing a 1 GB data set). The Postmark benchmark [4] models the workload seen by an internet service provider under heavy load—a combination of email, netnews, and e-commerce transactions.

2. MICROBENCHMARK RESULTS

Our measurements indicate that I/O response times for our SCSI over IP storage system are well modeled (error < 10%) by equations for a local disk access plus the additive network latency, as shown in the equations below. (Additional explanations and supporting measurements are available in an unpublished draft at http://www.bell-labs.com/~weeteck/papers/SOTechReport.ps.)

In these equations, T_{seek}, $T_{rotation}$, and T_{cmd} are the latencies for seek, rotation and command processing in the disk; T_{net} is the network latency, and T_{trans} is the disk transfer time (i.e.,number of bytes divided by media transfer rate). These equations hold for large TCP windows that minimize unnecessary fragmentation of SCSI data and the ensuing round-trip delays that fragmentation incurs. In the case of small network latency, the performance of remote storage accessed via our SCSI over IP implementation is approximately the same as for a local disk. In the case of large network latency, I/O response time is dominated by the network latency. We note that the FreeBSD OS 3.4 SCSI target driver does not support tagged queueing, so only one I/O is outstanding. As a concrete example of the speed of SCSI over IP, our system obtains 26 MB/s for 64KB I/Os given a small network latency, and 1.8 MB/s when the network latency is 8ms.

$$T_{seqRead/Write} = T_{cmd} + T_{trans} + 2\times T_{net}$$

$$T_{randRead} = T_{diskDelay} + 2\times T_{net}$$

$$T_{randWrite} = \begin{cases} T_{diskDelay,} & \text{for } T_{net} < T_{thresh} \\ T_{cmd} + T_{trans} + 2\times T_{net}\,, & \text{for } T_{net} \geq T_{thresh} \end{cases}$$

where

$$T_{diskDelay} = T_{cmd} + T_{seek} + T_{rotation} + T_{trans},$$

$$T_{thresh} = \begin{cases} T_{diskDelay} / 2, & \text{for a single disk} \\ \dfrac{T_{diskDelay}}{2 \times N_{disk} \times N_{seg}} & \text{for a disk array with } N_{disk} \text{ disks and } N_{seg} \text{ cache segments.} \end{cases}$$

Table 1: Application benchmark results.

Network Delay (ms)	SSH Response Time (sec)			SDET Throughput (Op/sec)			Postmark Throughput (Trans/sec)		
	Soft Upd	FFS	FFS-Sync	Soft Upd	FFS	FFS-Sync	Soft Upd	FFS	FFS-Sync
0	38.8	40.6	48.3	4033.6	1832.1	1054.7	271.5	244.7	142.4
0.1	41.9	50.1	89.3	1948.6	703.0	230.9	224.8	135.5	107.7
1	43.0	54.8	112.8	1437.7	490.7	156.6	126.2	74.4	57.9
2	44.5	61.1	140.7	1122.3	365.6	113.6	81.3	47.4	37.9
4	47.3	73.5	195.3	759.3	238.9	74.2	47.6	26.7	20.9
8	55.5	99.9	305.8	451.9	138.8	43.5	24.9	13.1	10.3

3. MACROBENCHMARK RESULTS

We now consider the effect of network latency on the performance of application-level benchmarks that access remote storage. We measure the UNIX fast file system (FFS), which uses synchronous writes for metadata updates, and two variants: one using synchronous writes for data updates too (FFS-sync), and one using asynchronous writes for most metadata (Soft Updates). To ensure that the bottleneck is in the network, the remote storage is a RAID-5 disk array with 128 MB cache and 8 fast disks.

Table 1 shows the performance of the SSH, SDET and PostMark benchmarks as a function of network delay. The metric for SSH is response time, so smaller values reflect faster execution, whereas SDET and Postmark measure operations per second, so larger values mean faster execution. These measurements show significant performance impairment for large network delays.

We now illustrate how to improve the write performance under high delay. Table 2 uses the Postmark benchmark, which does a large number of file system metadata writes, to demonstrate the effectiveness of a non-volatile write cache. A storage-side write cache (columns labeled SC) fails to mask network delays, whereas a host-side cache (columns labeled HC) is quite effective.

Table 2: Postmark transactions per second.

Delay	Soft Updates		FFS		FFS-Sync	
	SC	HC	SC	HC	SC	HC
0	277.8	312.1	186.9	189.1	160.0	157.5
0.1	235.3	298.5	141.8	181.8	117.6	144.9
1	130.7	303.0	76.9	185.2	62.5	147.1
2	85.5	298.5	48.9	186.9	40.2	149.3
4	49.4	303.0	27.3	186.9	22.7	149.3
8	26.0	298.5	14.2	177.0	11.8	148.1

Similarly, caching improves the read performance under high delay. Table 3 shows the effectiveness of the file system read cache at the host side, by running the Surge benchmark on uncachable and cachable request streams.

Table 1: Seconds to complete Surge benchmark.

Delay (ms)	No FS Cache	FS Cache
0	6.7	0.38
0.1	7.3	0.37
1	12.3	0.37
2	17.1	0.38
4	27.3	0.38
8	48.0	0.38

4. CONCLUSION

Broadly speaking, the performance of remote storage is comparable to that of server-attached disks that are a couple generations out of date. The delay of our SCSI over IP implementation is almost entirely in the speed-of-light propagation delay, so access to remote storage behaves in most respects like access to a local disk having a slower access time. In particular, our measurements validate simple equations stating that the stop-and-wait I/O time increases by an amount equal to the network delay. Because the remote storage acts like disks with a slightly slower access latency, many workloads will benefit from the traditional approaches of caching and prefetching, and these techniques should help remote storage to the same extent that they are effective in hiding the delays associated with access to local disks. In particular, our measurements indicate the benefit of read and write caching at the host side under conditions of high network latency, and the benefit of a file system that uses asynchronous metadata writes.

5. ACKNOWLEDGEMENTS

We thank Liddy Shriver and Banu Özden for helpful suggestions that improved the presentation of this extended abstract.

6. REFERENCES

[1] M. Seltzer, G. Ganger, M. McKusick, K. Smith, C. Soules, C. Stein, "Journaling versus Soft Updates: Asynchronous Meta-data Protection in File Systems", USENIX Conf. Proc., pages 71–84, San Diego, CA, June 2000.

[2] S. Gaede, "Perspectives on the SPEC SDET Benchmarks", http://www.spec.org/osg/sdm91/sdet/index.html.

[3] P. Barford, Mark Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", *Sigmetrics Conf. Proc.*, pages 151-160, Madison, WI, June 1998.

[4] J. Katcher, "PostMark: A New File System Benchmark", Technical Report TR3022. Network Appliance Inc, October 1997.