IP Storage and the CPU Consumption Myth

Robert Horst
3ware, Inc.
701 E. Middlefield Rd.
Mountain View, CA 94043

Abstract

This paper addresses a key issue that arises when attaching storage devices directly to IP networks: the perceived need for hardware acceleration of the TCP/IP networking stack. While many implicitly assume that acceleration is required, the evidence shows that this conclusion is not well founded. In the past, network accelerators have had mixed success, and the current economic justification for hardware acceleration is poor given the low cost of host CPU cycles. The I/O load for many applications is dominated by disk latency, not transfer rate, and hardware protocol accelerators have little effect on the I/O performance in these environments. Application benchmarks were run on an IP storage subsystem to measure performance and CPU utilization on Email, database, file serving, and backup applications. The results show that good performance can be obtained without protocol acceleration.

1. Introduction

The growing popularity of gigabit Ethernet has prompted increasing interest in using standard IP networks to attach storage devices to servers. These Ethernet Storage Area Networks (E-SANs), have significant advantages in cost and management ease compared with Fibre Channel SANs. Some IP storage products are already on the market, and work to standardize the protocols is progressing in the IP Storage working group of the IETF [1].

Networks customized to storage networking, such as Fiber Channel, were developed largely due to the perception that standard networking protocols are too heavyweight for attaching storage. Conventional wisdom says that IP storage is impractical without special purpose NICs to accelerate the TCP/IP protocol stack. This papers shows that the need for hardware acceleration is largely a myth. Several different lines of reasoning show that the future of storage networking will rely heavily on storage devices connected to servers without special purpose hardware accelerators.

2. The Historical Argument

There are many historical examples of hardware accelerators to offload processing tasks from the primary CPU. Some examples, such as graphics processors, have been successful, but the history of successful communications processors is filled with examples of unmet expectations.

Examples of front-end communications processors date from the early days of mainframe computing. In many systems, the primary CPU was accompanied by an I/O processor to offload low-level protocol operations. However, it has become increasingly difficult for this type of architecture to deliver real performance gains given the rapid pace of technology evolution.

A specific recent example is the Intel I^2O (Intelligent I/O) initiative. The idea was to have a communications processor, such as an Intel i960, on the motherboard to serve as an I/O processor to offload and isolate the CPU from its attached I/O devices. At the time the initiative started, the i960 embedded processor was adequate to the task, but its performance did not increase at the same rate as the main CPU. If the performance does not keep up, at some point an accelerator becomes a decelerator. Somewhere in between, performance is about equal with or without the attached processor, but the development and support costs become a burden. The accelerator is usually a different CPU architecture than the main CPU, and it usually has a different software development environment. Maintaining two such environments is costly, and even if they were identical, there is overhead for inventing and testing the software interface between the processors. The software development cost eventually kills the front-end processor architecture, until the next generation of engineers rediscovers the idea and repeats the cycle.

Some may argue that the problem was that the accelerators should have been optimized hardware instead of embedded programmable processors. Unfortunately, every protocol worthy of acceleration continues to evolve, and it is difficult to stay ahead of the moving target. The new protocols proposed for IP storage, iSCSI and iFCP, are far from stable, and even after the standards have been formally approved, there will likely be a long series of enhancements and bug fixes. It seems extremely

0-7695-1432-4/01 $10.00 © 2001 IEEE

premature to commit hardware to accelerating these protocols.

There may be more cause for optimism for general-purpose TCP acceleration, but history has not been kind to companies attempting this idea either. Many companies were unsuccessful in getting products to market. One company, Alacritech, is currently marketing a product for fast Ethernet acceleration [2], but this product did not have a gigabit accelerator at the time when the general-purpose gigabit NIC market was developing. This example points out the difficulty in keeping up with the blistering progress in networking and CPU speeds. Once the NIC vendors have skimmed off the most beneficial ideas (such as checksum offload, interrupt coalescing and jumbo frame support), there may not be enough performance gain for the special purpose accelerator vendors to compensate for the added development and product costs.

Accelerators for IPsec have shown a similar trend. They were initially popular, but are now disappearing because host-based software is improving and the CPUs are getting faster. Also, as processing needs are better understood, the most time consuming operations are gradually incorporated into standard CPUs and their support chips.

3. The Economic Argument

The economic argument for protocol acceleration is based on the premise that the computing power in the NIC is less expensive than computing power in the host. Examined closely, this premise is on shaky ground. Until the accelerator can be a single-chip ASIC, it will inevitably have multiple components including a processor, memory, and network interface components. The size of the accelerator market may not warrant development of a fully integrated solution, and the parts cost may increase as a result. Low volume also affects the amortized development cost. Today, Fiber Channel host bus adaptors sell in the range of $600 to $1000 while higher volume Gigabit Ethernet card costs have fallen to less than $150. The 4-port 100BASE-T Alacritech TCP accelerator sells for a list price of $699, compared to a standard gigabit 1000BASE-T NIC that sells for under $150 and will soon be a standard feature of most servers.

The accelerator cost must be weighed against the cost of running the protocol in software in the main processor. The cost of using the main processors depends on assumptions of the incremental cost of processing, and the amount of CPU required to implement a storage networking protocol. Today, the Dell Poweredge 1400SC server is offered in single and dual processor versions. The cost to upgrade to a second 800MHz CPU is $399. With this system, the cost of the second processor and 1000BASE-T card ($149) is much less than the $699 cost of the 4-port Alacritech NIC.

Comparing the merit of the alternatives requires an estimate of how much processing power (percent CPU utilization) it takes to run the storage and TCP protocols in the main processor. This percentage varies by operating system, because some systems have more efficient TCP and IP storage driver implementations that avoid extra data copies. Merits of different solutions also depend on the I/O workload and the storage protocol efficiency. Conventional wisdom says that it takes one high-speed processor of about 800 MHz to stream TCP continuously at Gbit rates. If this is the case, then running the TCP protocol would take the entire second CPU, but would still be a better deal than the accelerator NIC, even if that accelerator offloaded 100% of the protocol (not likely). Moreover, the balance shifts more towards the second CPU as real processing workloads are examined, because it is extremely rare for an application to simultaneously stream I/O at full speed and compute at full speed. Also, when a complete application is considered, the second processor can be used for other tasks at times when I/O workloads are low, and may accelerate those compute-intensive phases of the application.

An I/O accelerator has other drawbacks as well. It takes up an extra valuable PCI slot in systems with built-in GbE. The accelerator reduces the choices in NICs, and may lack other features needed by the server, such as data encryption. There is usually a time lag in availability of new features and performance improvements in hardware accelerators compared with software solutions. Over time, CPU performance will improve at a greater rate than the accelerator due to larger market for general-purpose microprocessors.

Similar economic arguments apply to System Area Network such as ServerNet, Scalable Coherent Interface (SCI), Giganet, Myrinet and Infiniband. While these networks have demonstrated greatly improved bandwidth and latency relative to Ethernet, none have yet been widely accepted in mainstream computing markets. They have had the same difficulty demonstrating enough benefit to mainstream applications to justify their extra cost.

4. The Disk Argument

The preceding discussion assumed a system spends 100% of its time transferring a maximum I/O streaming rate. Real applications have much different behavior due to the mix of sequential and random I/O.

Today's disk drives can transfer sequential blocks at 20-40 MB/s, but can perform only about 100 I/Os (seeks) per second [3]. If each seek transfers a 4 Kbyte block, then the random transfer rate is just 0.4 MB/s, or nearly 100 times slower than the sequential rate. Most real applications access disk storage via a file system with data organized in noncontiguous pages, or via a database with relatively small record sizes. These applications transfer at closer to the random rate than the sequential rate.

When storage is connected through IP networks without acceleration, CPU consumption for the IP stack is determined primarily by the number data copies and the total amount of data moved. When an application is doing mostly random I/O, there is little data moving through the IP stack, keeping CPU consumption low. This type of application gets little benefit from hardware acceleration.

On Oct. 6, 2000, Compaq set a new transaction processing record on the industry-standard TPC-C benchmark [4]. The record was 505,302 transactions per minute obtained using 2568 disk drives and 24 8-processor servers. Assuming about 0.5 IOs/sec/tpmC [5], each disk performs 98 I/O accesses per second, keeping the disks very busy. Measured from the CPU side, each of the 192 CPUs performs 1315 IOs per second. If each IO averages 8 KB, the average data rate is just 10.3 MB/s per CPU. If this benchmark had been run with IP storage and no protocol acceleration, the relatively low data rate would have consumed only a few percent of the CPU time. The net result is that the low cost of IP storage subsystems should give better TPMC price/performance than SCSI or Fibre Channel solutions.

A similar case can be made for scientific applications. A 1997 study by Rich Martin measured the sensitivity of ten different scientific applications with respect to communication bandwidth and latency [6]. This study shows that many scientific applications do very little I/O and are much less sensitive to communications bandwidth than to latency or overhead, indicating that the applications are primarily CPU bound, not I/O bound. One might think CPU-bound applications are ideal for protocol offloading, but if the amount of IO is small compared to the amount of computation, very little of the run time is subject to acceleration. Also, many scientific programs perform their IO at the beginning and end, with primary computation in between. During the I/O phases, the CPUs are often lightly loaded. The net result is that when IP storage is used for the backing store for scientific applications, network acceleration should show little performance gain.

5. Measurement results

When 3ware began to implement the first native IP storage products, the preceding arguments were the only ones available to guide the decision on whether to implement the client-side protocol in the main CPU or in a hardware accelerator. 3ware chose to develop a protocol over TCP/IP that could be implemented very efficiently in software.

IP storage products began shipping in late 2000, and include the Palisade 400 and Palisade 100 products. OS driver support includes Linux, Windows NT, Windows 2000, Solaris and Macintosh [7,8]. Now that the products are complete, measurements of real applications validate the original design decisions. Tests of Email servers, database, file server and backup programs show that the CPU overhead is not excessive and application performance is typically no different than with locally attached SCSI or Fiber Channel storage. Preliminary benchmark results for these applications are given below:

5.1 Email Server

Email comprises over half of the disk storage of many organizations. The Email data often resides on drives internal to the server, or on nearby SCSI RAID arrays. This type of configuration is prone to run out of storage quickly, and it is desirable to seamlessly scale the storage as needed. IP storage provides an ideal solution. A test of Microsoft Exchange was run to determine how well IP storage would serve this application.

Test setup

Server: Dual 866 MHz Pentium III
OS: Windows NT 4.0
Mail server: Microsoft Exchange
Storage: 3ware Palisade 400 IP Storage
640 GB RAID 0 array
Clients: 1400 clients simulated with
5 client systems

Results

Response time: 187 ms avg. per mail message
CPU Utilization: 13%

These results are quite impressive, even measured against typical response time results with internal SCSI arrays. The low CPU utilization means that hardware acceleration of TCP or the IP storage protocol would have shown no significant performance difference.

5.2 Database Server

IP storage will give scalability to existing and future database applications. Databases require the block level access that is native to IP storage and not generally available in network-attached storage (NAS) boxes that communicate using a file protocol such as NFS or CIFS. A database asset tracking application was run on a database server connected through a Gigabit Ethernet switch to a collection of client machines:

Test setup

Server: Dual 866 MHz Xeon
OS: Windows NT 4.0
Database: Oracle 8
Storage: 3ware Palisade 400 IP Storage
640 GB RAID 0 array
Clients: Database transactions simulated with
14 client machines

Results

1.8 Million transactions per hour (500 TPS)
Zero transaction timeouts

In this test, even higher transaction rates would be possible with more clients. The client load was not quite high enough to saturate the server.

These results show that database applications run very well on IP storage, and that there would gain little benefit from hardware acceleration. Very high transaction volumes can be supported, and no transaction timeouts were seen. This is a key point, as experienced database administrators know that some of the most expensive storage subsystems available today are prone to transaction timeout problems.

5.3 File Server

IP storage can provide scalable storage behind a file server. This type of file sharing takes advantage of standard servers acting as the file server, allowing administrators to use familiar operating system administration tools to manage the pool of shared storage. Small reads and writes were run on a Windows 2000 machine to show that the overhead for IP storage does not affect the ability to serve many simultaneous file requests.

Test setup

Server: Dual 866 MHz Pentium III
OS: Windows 2000
I/O load generator: Iometer
Queue depth: 8
Storage: 3ware Palisade 400 IP Storage
640 GB RAID 0 array

Results

Random 2K Reads
806 IOPS
14 % CPU utilization
9.9 ms average response
Random 2K Writes:
1760 IOPS
27 % CPU utilization
4.6 ms average response

This test uses Intel Iometer to simulate the load that would be seen by a file server; no clients were actually simulated. The low CPU utilization at the file server again shows that IP storage can be quite effective even without hardware acceleration. Performance is limited only by the number of seeks per second that can be done by the drives. The random write performance is higher than would be expected from eight physical drives, because this test takes advantage of caching in the IP storage unit.

5.4 Backup

Backup is often cited as the most important application that requires high streaming rates to or from a storage subsystem. However, in most installations, backups are not done during peak processing times, but in times when the system is lightly loaded or quiescent. Almost by definition, the CPU load during backup is not a major concern. A test of the Windows 2000 backup application was done to test the time to backup the system files (C: drive) to either to either an IP Storage device or a network shared volume acting as a NAS device.

Test setup

Backup client:
500 MHz Pentium III running Windows 2000
Backed up data: 1.6 GB data on "C:" drive
Backup target:
IP Storage: 3ware Palisade 400 IP Storage
640 GB RAID 0 array
NAS: Windows 2000 file sharing
on 500 MHz Pentium III

Results

IP storage backup: 3 minutes 3 seconds
NAS backup: 3 minutes 31 seconds

The backup was significantly faster on the IP storage unit that could take advantage of multiple disk arms seeking simultaneously. In this test, the IP Storage unit was not busy and could have handled many simultaneous backups. Some other backup experiments were severely limited by disk performance of the system being backed up. It is clear that IP storage is well suited as a high-performance backup device. The huge advantage in access time relative to tape will make IP storage increasingly attractive for local and remote online backups [9].

5.5 Other factors affecting performance

All of the preceding tests were configured with Gigabit Ethernet (1000BASE-T) and with RAID level zero. Additional tests were run to evaluate how those parameters affect performance.

Figures 1a and 1b show performance and CPU utilization on a workstation I/O load running on three different configurations of hardware RAID connected through IP. The RAID 0 configuration has data striped across 8 drives. RAID 10 has data striped across four mirrored pairs of drives, and RAID5 rotates 64K parity blocks across the 8 drives, yielding 7 drives of useful capacity.

Test setup

Server: Dual 1GHz Pentium III
OS: Windows NT 4.0
I/O load generator: Iometer
Storage: 3ware Palisade 400 IP Storage
RAID 0, 5 and 10
Controller write caching disabled
Drive write caching enabled
Workstation Access Pattern:
Block size: 8KB
80% Read, 20% Write
80% Random, 20% Sequential

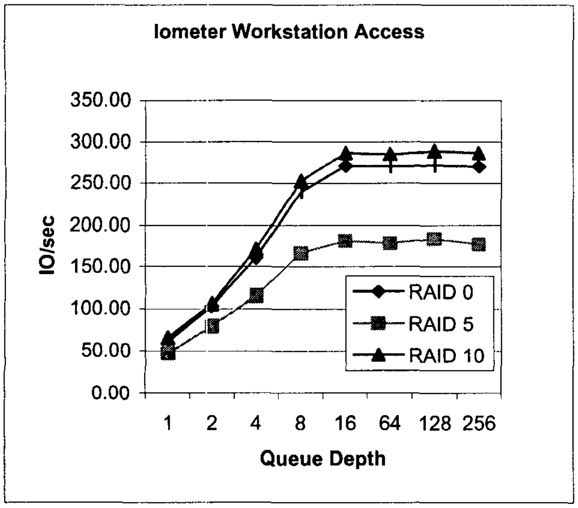

Figure 1a. Effect of RAID level on I/O performance of IP-attached storage.

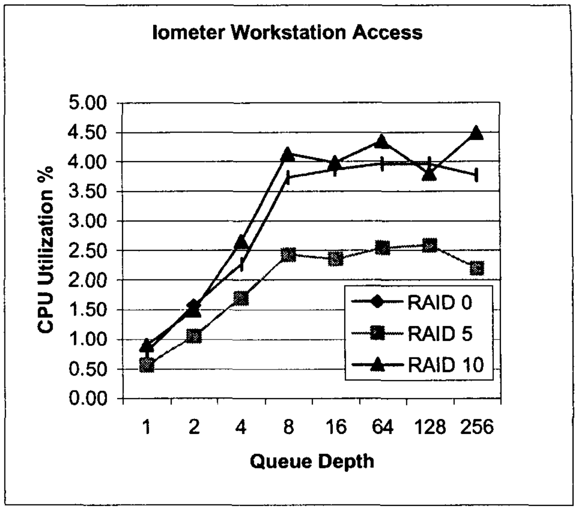

Figure 1b. Effect of RAID level on CPU utilization of IP-attached storage.

In these tests, queue depth simulates the amount of I/O that the system has outstanding at a point in time. Greater queue depth allows more pipelining in the I/O system and results in better performance. Queue depth can be increased either by one application issuing non-waited I/O, or through multiple applications issuing independent I/O operations.

The performance graph in Figure 1a shows that RAID 0 and 10 can perform more I/O per second than RAID 5, mostly due to the parity calculations required to do updates on RAID 5. In all cases, the performance is determined primarily by the drives or disk controller, not by the TCP/IP stack or small latencies introduced by the IP connection. The utilization graphs show that CPU consumption closely tracks the I/O throughput. Total CPU utilization never exceeds 5 % for these tests.

The effect of network speed is illustrated by the graphs in Figure 2a and 2b. This test uses the same test setup as Figure 1, but tests both 100 Mbit and 1 Gbit network connections on an IP storage unit configured with 8 drives in RAID 0 mode. Iometer is used to generate a sequential workload consisting of sequential 256K writes.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you’re on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm’s cloud-native docket research platform finds what other services can’t. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.