PARALLEL DATA LABORATORY
CARNEGIE MELLON UNIVERSITY

THE
PDL Packet
THE NEWSLETTER ON PARALLEL DATA SYSTEMS • SUMMER 1996

I•N•S•I•D•E

CONSORTIUM MEMBERS

Data General

Digital Equipment Corporation

EMC Corporation

Hewlett-Packard

International Business Machines

Seagate

Storage Technologies, Inc.

Symbios Logic

YEAR•IN•REVIEW

September 95
CMU Steve Blumenau, et al. EMC.
December 95
SOSP15 Hugo Patterson presented TIP.
January 96
CMU Tom Cormen, Dartmouth, on ViC*.
February 96
DARPA Garth reviewed NASD.

A Taxonomy for Network-Attached Storage

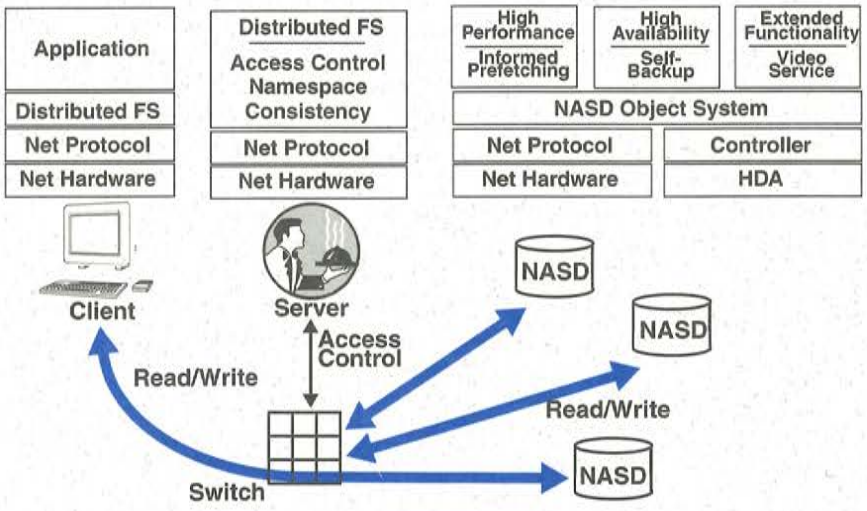
Researchers in the NASD group, in an effort to better organize their work, have developed a simple taxonomy of current and reasonable offerings for network-attached storage. There are four functional elements in this taxonomy: server-attached disk, server-integrated disk, network SCSI, and network-attached secure disks.

Most storage today comes in server-attached disk (SAD) configurations. Disks are privately attached to general-purpose machines that are dedicated to running the distributed file system code. A client wanting data from storage sends a message to the file server, which parses it and sends a message to storage, which accesses the data and sends it back to the file server, which finally sends the requested data back to the client. Server-integrated disk (SID) is logically the same structure, except that the hardware and software in the file server machine is specialized to the file service function.

NASD raises the storage interface abstraction, enabling automatic storage management and transparent optimizations.

—Garth Gibson

... continued on p.4



NASD's direct transfers between client and disk allow scalable bandwidth, lower latency, and off-load much of the file server's work without integrating file system policy into the disk.

... continued on p.6

Year 3: The Year of the Interface

From the Director's Chair

GARTH GIBSON

The third year of the Parallel Data Lab has been a year of building. Notably, underway or in place are: a network-TIP experiment, a prototype NASD storage command and security interface, a low-level parallel file system interface proposal, and a RAIDframe release. Moreover, the School of Computer Science more than doubled our lab space. Of course, the crack in the ceiling of the new lab led to our worst problem of the year — water dripping onto machines — but we're told that this has been repaired :-)

At the top of our stack has been the Network-Attached Secure Disk (NASD) project proposed in the Spring of 1995 and funded just before last year's retreat. With the addition of Eugene, Dana, Chen, Patty, Berend, and Marc, we have about 12 in our NASD team today. This team has efforts underway in NASD performance modeling; NASD drive, file system, and embedded video application prototyping; NASD security analysis; informed prefetching to the drive experiments; and storage interconnect analysis.

We have structured our thinking around a simple taxonomy of reasonable network-attached storage systems:

- today's general purpose workstation running server code, Server-Attached Disk (SAD);
- today's special-purposed file server boxes, Server-Integrated Disk (SID);
- the minimum changes to SCSI disks needed for secure, efficient network storage, Network SCSI (NetSCSI); and
- a "SCSI" much evolved to allow clients to directly initiate transfers from storage, Network-Attached Secure Disks (NASD).

Using NFS and AFS trace data and direct measurements of SAD implementations, we have estimated that NetSCSI drives might reduce NFS server load by 9% and AFS server load by 50%, while NASD drives might reduce NFS server load by 90% and AFS server load by 75%. Within this taxonomy, we have defined a set of reasonable security thresholds based on the security mechanism at the drive:

- simple accident avoidance, like NFS today (no mechanism);
- storage integrity assurance, where the validity of all commands and data is checked (message digests);
- transfer privacy assurance, where commands and data cannot be eavesdropped (encryption); and
- physically insecure environments tolerated, where even probe and scope attacks are countered (tamper-resistant secure co-processors).

In addition to our DARPA-sponsored NASD research, we continue collaboration with the National Storage Industry Consortium's working group on Network-Attached Storage Devices (NSIC/NASD). Meeting about four times a year, with public and private sessions, the core working group has been hammering out an intellectual property rights agreement to allow early sharing of pre-standards architectural R&D results. Current participants, in addition to CMU, are IBM, HP, StorageTek, and Quantum.

... *continued on p.4*

THE PDL PACKET

The Parallel Data Laboratory
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15217-3891
412•268•3835 VOICE
412•268•3010 FAX
412•268•5576 FAX ALTERNATE

PUBLISHER
Garth Gibson

EDITOR/DESIGNER
LeAnn Neal Reilly

PRODUCTION EDITOR
Dale A. James

The *PDL Packet* is published one or more times a year and mailed to members of the Parallel Data Consortium. Copies are given to other researchers in industry and academia as well. Postscript versions will reside in the Library section of the PDL Web pages. Contributions are welcome.

COVER ILLUSTRATION

Skibo Castle and the lands that comprise its estate are located in the Kyle of Sutherland in the northeastern part of Scotland. Both 'Skibo' and 'Sutherland' are names whose roots are from Old Norse, the language spoken by the Vikings who began washing ashore regularly in the late ninth century. The word 'Skibo' fascinates etymologists, who are unable to agree on its original meaning. All agree that 'bo' is the Old Norse for 'land' or 'place.' But they argue whether 'ski' means 'ships' or 'peace' or 'fairy hill.'

Although the earliest version of Skibo seems to be lost in the mists of time, it was most likely some kind of fortified building erected by the Norsemen. The present-day castle was built by a bishop of the Roman Catholic Church. Andrew Carnegie, after making his fortune, bought it in 1898 to serve as his summer home. In 1980, his daughter, Margaret, donated Skibo to a trust that later sold the estate.

PARALLEL DATA LABORATORY

MISSION OF THE LAB

To advance the state of the art in storage subsystems and to integrate them efficiently into parallel file systems, high bandwidth networks, and multicomputers.

CONTACTING US

WEB PAGES

http://www.cs.cmu.edu/Web/Groups/PDL/

FACULTY

Garth Gibson
(director)
garth.gibson@cs.cmu.edu

David Nagle
david.nagle@ece.cmu.edu

STAFF MEMBERS

Anne Byrne
anne.byrne@cs.cmu.edu

Chris Demetriou
chris.demetriou@cs.cmu.edu

Chen Lee
chen.lee@cs.cmu.edu

Patty Mackiewicz
(consortium administrator)
412•268•6716
patty.mackiewicz@cs.cmu.edu

Paul Mazaitis
paul.mazaitis@cs.cmu.edu

Victor Ortega
victor.ortega@andrew.cmu.edu

Tammo Spalink
tammo.spalink@cs.cmu.edu

Marc Unangst
marc.unangst@cs.cmu.edu

Jim Zelenka
jim.zelenka@cs.cmu.edu

GRADUATE STUDENTS

Khalil Amiri
khalil.amiri@cs.cmu.edu

Fay Chang
fay.chang@cs.cmu.edu

Bill Courtright
bill.courtright@cs.cmu.edu

Eugene Feinberg
eugene.feinberg@andrew.cmu.edu

Eka Ginting
eka.ginting@cs.cmu.edu

Howard Gobioff
howard.gobioff@cs.cmu.edu

Dana Hustins
dana.hustins@andrew.cmu.edu

Qingming Ma
qingming.ma@cs.cmu.edu

Berend Ozceri
berend.ozceri@cs.cmu.edu

Hugo Patterson
r.hugo.patterson@cs.cmu.edu

Erik Riedel
erik.riedel@cs.cmu.edu

David Rochberg
david.rochberg@cs.cmu.edu

Chang-Ming Wu
chang-ming.wu@cs.cmu.edu

RAIDframe Released in Fall 1996

RAIDframe, a rapid prototyping tool for RAID system designers, was released prior to the 1996 PDL workshop and retreat. Although the software is being made widely available unsupported, Parallel Data Consortium members are encouraged to contact the PDL for assistance with configuration and use of RAIDframe.

As a rapid prototyping tool, RAIDframe offers designers the ability to work in three environments: a stand-alone user application, an event-driven simulator, and an in-kernel device driver. Studies conducted by the developers show that RAID levels 0, 1, 4, 5, 6, and parity declustering implemented in RAIDframe perform as expected. Equally important, the developers found that the stand-alone application performs as well as the in-kernel version — a key benefit for designers who want to avoid programming in the kernel.

RAIDframe simplifies code development in three powerful ways. First, RAIDframe expresses RAID architectures in directed, acyclic graphs that can be executed by a scheduling engine unaware of the RAID features. Second, having libraries that include full descriptions of multiple architectures allows a new experimental architecture to be compared to many existing architectures easily. Finally, by including a simple commit node into graphs, rollaway recovery can mechanically resolve the correct action for all in-flight RAID operations at the time of a failure.

RAIDframe has been tested extensively on Alphas running Digital Unix 3.2. The simulator version works on many platforms, including Alphas running Digital Unix 2.0 or 3.2x or NetBSD1.2, i386 running NetBSD1.1or Linux 1.2 or 2.0, PARISC running HPUX 9.x, RS6000 running AIX 3.2.5 or 4.1, SPARC running Solaris 5.4 or SunOS 4.1, DEC MIPS running ULTRIX 4.2, and SGI running IRIX 5.3.

Documentation detailing the installation, use, and extension of RAIDframe is also available. Questions can be sent to raidframe@cs.cmu.edu.

Annual Workshop and Retreat Set for Sept. 23-25

The fourth annual Parallel Data Systems Workshop and Retreat is scheduled for September 23 through 25 at the Wisp Resort in Deep Creek Lake, Maryland. Industry sponsors from the Parallel Data Consortium and academic researchers from the CMU community will be joined on the final day by attendees of the National Storage Industry Consortium's (NSIC) working group on Network-Attached Storage Devices.

These workshops are small and informal. Most speakers will be CMU researchers in the Parallel Data Lab or their friends researching related topics. Questions and discussion are encouraged, and visitors are asked to report informally their impressions and to make suggestions.

Although the days are long, this is a retreat, and as such the schedule ensures long breaks (inside and outside) for personal interaction among CMU and guest researchers.

... continued from p.1

Network-Attached Storage Taxonomy

SAD storage suffers from a bottleneck in the file server machine; all data is copied through its protocol stacks. SID avoids this by integrating the server into storage, but to do so it must bind to a particular distributed file system.

By network SCSI (NetSCSI), the next level in the taxonomy, NASD researchers define the minimal extensions to existing SCSI disks that yield useful network-attached storage. A file manager machine translates its clients' file system requests into SCSI commands for its disks, but rather than returning data to the file manager to be forwarded, a NetSCSI disk sends data directly to the client (similar to a SCSI COPY command). The biggest challenge for NetSCSI disks is security; it cannot trust all commands it receives in a real network.

NASD researchers identify two approaches to correcting the security problem for NetSCSI disks: the disk can provide a second network port physically private to the file server or it can provide cryptography support for virtually private channels to the file server. If cryptography is used, it can range from message digest software for authenticating commands and data, to encryption hardware for ensuring privacy, to secure co-processors able to ensure that a stolen drive does not reveal its data or security keys.

With network-attached secure disks (NASD), the last member of the taxonomy, the focus is on changing the (SCSI) command interface to off-load more of the file server's work onto the disk without integrating file system policy into the disk. Fast-path operations, like reads and writes, go straight to the disk, and less-common ones, like namespace manipulations, go to the file server. With this direct communication approach, disks must authenticate a client's commands without file server intervention, so the second physically private network is not an option and cryptographic support is required.

Because clients directly access file regions, NASD drives maintain disk-layout metadata internally, enabling smart drives to better exploit detailed knowledge of their own resources to optimize data layout, readahead, and cache management. This is precisely the type of value-added opportunity that nimble storage vendors can exploit for market and, more importantly, customer advantage.

To better understand the tradeoffs among the new architectures in this taxonomy, the NASD researchers have recently analyzed AFS and NFS file-system traces to 1) characterize the behavior and cost of distributed-file-server functionality and 2) provide data for analytic models of the Net-

... continued on p.6

... continued from p.2

From the Director's Chair

While NASD is in the forefront of most of our efforts today, our previously established projects have also made significant progress. RAIDframe, whose release was delayed until September 4, 1996, by surprisingly extensive debugging and documenting, now includes Bill Courtright's automatic recovery for RAID operations in-flight at the time of a failure and Chang-Ming Wu's implementation of IBM's EVEN-ODD RAID level 6 architecture. Bill has a first draft of his dissertation written and is back at Symbios Logic.

The informed prefetching project has also matured this year. Hugo did a super job of presenting TIP to the premier operating systems conference in December, 1995, and has been closeted with his dissertation since. Distracting him have been the interesting efforts of Eka Ginting, to develop automatic methods of extracting hints from complex, non-array codes like databases, and those of Andrew Tomkins, whose detailed comparison of TIP's prefetching and caching strategy with the Washington/Princeton/Wisconsin approach has led to a joint paper with this other group. Most interesting for NASD, however, are David Rochberg's experiments with TIP over the network, because the object-based NASD interface has enough information for NASD drives to prefetch based on TIP hints without filesystem help.

Our parallel file system efforts have been refocused by the departure of Dan Stodolsky. We have invested more heavily in the low-level parallel file system programmer's interface working group within the Scalable I/O Initiative. This effort was sparked by our ideas about a performance-critical small interface on which can be built efficient application-specific high-level parallel file system libraries such as NASA's MPI-IO, IBM's PIOFS, or Intel's PFS. The API, drafted by CMU, Princeton, Arizona, Illinois, and IBM, will be released for public review October 1 and a roundtable feedback session will be held at Supercomputing 96 in Pittsburgh November 17-20. We are pleased to report that support for hints and client caching are included with scatter/gather, asynchronous transfers, atomic controls during open, collective transfers, fast copy, and space preallocation. It is our desire to minimize the codepath and semantic difference between the NASD interface and the SIO low-level parallel file system interface so that clusters of client machines can use clusters of NASD drives for parallel processing of large files.

I look forward to explaining all of this and more at our 1996 Parallel Data Systems Retreat and Workshop, September 23-25, at Wisp resort in Deep Creek Maryland. See you there!

PDL Faculty Win Honors

Garth Gibson, Director of CMU's Parallel Data Lab, was appointed the Litton Junior Faculty Fellow in the Computer Science Department at Carnegie Mellon, in recognition of his many research achievements during 1995. He was acknowledged by his colleagues and friends at a CS Departmental Meeting. When asked to join Jim Morris [CSD department head] at the podium to receive his plaque and honors, it was noted "that he couldn't make the meeting because he was busy working...!"

from SCS-Today, Jan. 29, 1996

Garth was not the only PDL researcher winning an award this year. Both David Nagle and Hui Zhang, members of the NASD project, received Career Awards from the National Science Foundation.

NSIC Working Group on Network-Attached Storage

The mission statement for NSIC's Network-Attached Storage Device working group is: "To develop, explore, validate, and document the technologies required to enable the development and adoption of network-attached storage devices and systems."

Quoting from the emerging working group agreement, "The intent of creating this project is to develop sufficient early understanding of Network-Attached Storage Device technology that we can make recommendations to guide a future standards-defining effort." In this sense we are investing in the creation and appropriate shaping of a marketplace that some or all of us would like to compete in. The reasons, then, for an intellectual rights sharing agreement is that, inevitably, convincing each other of the correctness of some aspect of the pre-standards definition will require divulging compelling information that is pertinent to the interface issue in question.

Although members of the working group proper are defined as signatories to the intellectual rights agreement, in practice each working group meeting contains a public session lasting about half a day. The public sessions facilitate input from related industries and pre-standard forums.

The next NSIC working group meeting is being held on the evening of September 24 (private session) and in the morning of September 25 (public session at Wisp Resort in Deep Creek Lake, Maryland. The following meeting is planned for the first week of January 1997 in Maui, Hawaii; it will be held in conjunction with a task force in Storage Architecture, a component of the Hawaii International Conference on Systems Sciences (HICSS).

For more information, visit the working group's Web page at:

http://www.hpl.hp.com/SSP/NASD

Thesis Proposal

Since the last *PDL Packet*, one graduate student has proposed his Ph.D. thesis topic and been accepted. The abstract for his proposal follows.

Automatic Hint Generation for I/O Optimizations proposed by Eka Ginting

A classic computer system problem is the continuously widening performance gap between CPU and disk. One powerful strategy to mitigate the effect of this gap is through disk array technology, which directly benefits applications that have explicit parallelism in their I/O accesses. Applications that do not have this characteristic, however, cannot realize the full benefit of the disk array. Patterson's informed prefetching and caching (TIP) system comes to the rescue by exposing the hidden I/O parallelism that exists in these applications. The TIP system clearly demonstrates that applications can derive significant performance improvement by disclosing their future accesses in the form of hints.

Patterson's TIP system currently requires programmers to annotate the application code with hints, limiting the acceptance of this powerful technology to those applications whose programmers are explicitly worried about I/O performance. This thesis proposes that we can automatically generate hints in applications for disk I/O prefetching through three mechanisms: first, static analysis similar to optimizing compiler analysis to extract knowledge of file accesses; second, dynamic analysis by pre-execution of parts of applications to discover future file accesses; and third, historical analysis to predict future file accesses based on learned past accesses. To successfully employ these mechanisms, the TIP system must be strengthened with a measure to model hint quality and to increase the robustness of hint handling. This thesis research will demonstrate that I/O access hints that enable powerful prefetching and caching can be generated automatically by these mechanisms.

Lab Status

In late March, the Parallel Data Lab expanded its facilities by adding a lab space next door to its original lab in 3607 Wean Hall. The original lab has become the group's machine room while 3606 Wean Hall has become its main communal workspace. Good timing, too, because 25 new workstations arrived, filling the 3607 space formerly occupied by people.

... continued from p.4

Network-Attached Storage Taxonomy

work SCSI (NetSCSI) and NASD storage architectures.

Armed with the statistics these traces gave them, NASD researchers project that NetSCSI may reduce the workload for AFS file managers by 50% while NASD may lower it up to 75%. For NFS, NASD does even better against NetSCSI—NASD lowers the workload up to a significant 91%, which is a factor of 9 improvement over NetSCSI's 9% reduction.

The group published these results in "A Case for Network-Attached Secure Disks," Technical report CMU-CS-96-142, Carnegie Mellon University., June 1996. An abstract of this report is included in the Publications section on page 7 of *The PDL Packet*; a copy of the report itself can be downloaded from the PDL Web pages.

NEWSBRIEFS CONTINUED

Departures & Arrivals

Three members of the Parallel Data Lab have left: Bill Courtright, LeAnn Neal Reilly, and Daniel Stodolsky.

Bill Courtright is back at Symbios Logic while he writes a dissertation and pursues Garth to read it. While with the PDL, Bill worked on RAID software design and RAIDframe development.

LeAnn left the group on maternity leave in late July; her first child was born shortly thereafter. While with the group, she worked on technical writing and design projects.

Daniel recently accepted a position with Verity, Inc. in Sunnyvale, CA. At Verity Daniel will head up the performance group. As a graduate student, Daniel worked on parity logging, write deferring, SPFS, and the SIO API.

New members have joined the ranks of the PDL in various capacities: Anne Byrne, Howard Gobioff, Dana Hustins, Chen Lee, Patty Mackiewicz, David Rochberg, and Chang-Ming Wu.

Anne Byrne divides her time between Garth and Prof. M. Satyanaranan as an executive assistant. She comes to CMU via the Professional Services Group, a Pittsburgh-based computer-consulting group.

Howard Gobioff, a fourth year graduate student working with Doug Tygar on security, has joined the NASD project's effort on storage security.

Dana Hustins is a new graduate student in ECE, currently working with David Nagle. Dana joins us from Georgia Institute of Technology.

Chen Lee, a project scientist, works on the video service embedded in the NASD project led by Prof. Hui Zhang.

Patty Mackiewicz joins the group from CMU's Robotics Institute. She serves as Consortium Administrator, as well as general information wizard.

David Rochberg is a graduate student who begins his third year this fall. David's speciality is prefetching and security for the NASD project.

Chang-Ming Wu is an Electrical and Computer Engineering graduate student who joined the group to work on EvenOdd for RAIDframe.

YEAR•IN•REVIEW CONTINUED

February 96 (cont.)

RAID Adv. Brd.	Garth reviewed NASD.
CMU	Chris Bajorek, IBM.
Princeton	SIO parallel file system workshop.
MIT	Garth presented TIP.
CMU	Anna Karlin, U. of Wash.,

March 96

CMU	John Wilkes, HP.
NSIC	Dave Nagle on NASD.
CMU	NSF's DSSC review.
U. Mich.	Dave Nagle on NASD.

April 96

Stanford	Garth presented TIP.
Chicago	SIO parallel file systems workshop.
Storage Tek	Garth on NASD.

May 96

CMU	John Wilkes, HP on AFRAID.
Sigmetrics	Bill Courtright on RAIDframe.
Argonne	SIO parallel file systems workshop.

June 96

NSIC	Garth on NASD taxonomy.
Boston	Garth co-chaired I/O Issues Group at ACM Wkshp. on Strategic Directions for Computing Research.

July 96

CMU	SIO parallel file systems workshop.
NSIC	HP hosted WG on NASD.
CMU	Ray Abuzayyed and Denis Mee, IBM, visited.

August 96

Compaq	Garth presented PDL work.

September 96

MSS96	Erik Riedel on AFS; Garth keynote on NASD.
IPDS96	Bill Courtright on recovery in RAIDframe
CMU	PDS Workshop & Retreat
NSIC	PDL hosts WG on NASD following PDS Wkshp.

A Structured Approach to Redundant Disk Array Implementation

In the *Proceedings of the International Computer Performance and Dependability Symposium (IPDS)*, Champaign-Urbana, IL. Sept. 4-6, 1996. Also available as technical report CMU-CS-96-137.

Abstract

Error recovery in redundant disk arrays is typically performed in an ad hoc fashion, requiring architecture-specific code which limits extensibility and is difficult to verify. In this paper, we describe a technique for automating the execution of redundant disk array operations, including recovery from errors, independent of array architecture. Our approach employs a graphical representation of array operations and a two-phase error-recovery scheme we refer to as roll-away error recovery. We demonstrate the validity of this approach in RAIDframe, a prototyping framework that separates architectural policy from execution mechanism. RAIDframe facilitates rapid prototyping of new RAID architectures by localizing modifications. In addition, RAIDframe-implemented architectures run the same code when configured as an event-driven simulator, a user-level application managing raw disks, and as a Digital Unix device-driver capable of mounting a file system. Evaluation shows that RAIDframe performance is equivalent to less complex array implementations and that case studies of RAID levels 0, 1, 4, 5, 6, and parity declustering achieve expected performance. –

A Trace-Driven Comparison of Algorithms for Prefetching and Caching

Available as technical report CMU-CS-96-174.

Abstract

Recently, several researchers have developed systems to make use of application-provided information about access patterns. These systems improve filesystem performance by *prefetching* data when it is advantageous to do so, and by making better-informed *caching* decisions when a block must be ejected from the buffer cache. Currently, two different research groups have reported systems for integrated prefetching and caching: the TIP2 system of Gibson, Patterson et al, and the LRU-SP system of Cao, Felten, Karlin and Li. Published studies of each of these approaches have been incomparable. In this paper we present trace-driven simulations comparing these two systems. Our results can be summarized in three statements. First, the systems perform similarly with respect to prefetching decisions within a single process. Second, TIP2 performs better on average with respect to buffer allocation decisions among multiple processes. And third, as disk bandwidth increases, LRU-SPs aggressive prefetching mechanism results in less caching and requires substantial overhead to submit additional I/Os and service the resulting interrupts.

A Case for Network-Attached Secure Disks

Available as technical report CMU-CS-96-142.

Abstract

By providing direct data transfer between storage and client, network-attached storage devices have the potential to improve scalability (by removing the server as a bottleneck) and performance (through network striping and shorter data paths). Realizing the technology's full potential requires careful consideration across a wide range of file system, networking, and security issues. To address these issues, this paper presents two new network-attached storage architectures. (1) Networked SCSI disks (NetSCSI) are network-attached storage devices with minimal changes from the familiar SCSI interface. (2) Network-attached secure disks (NASD) are drives that support independent client access to drive provided object services. For both architectures, we present a sketch of repartitionings of distributed file system functionality, including a security framework whose strongest levels use tamper-resistant processing in the disks to provide action authorization and data privacy even when the drive is in an physically insecure location.

Using AFS and NFS traces to evaluate each architecture's potential to decrease file server workload, our results suggest that NetSCSI can reduce file server load during a burst of AFS activity by a factor of about 2; for the NASD architecture, server load (during burst activity) can be reduced by a factor of about 4 for AFS and 10 for NFS.

... continued on p.8

Work With SIO Initiative Continues

by Erik Riedel

Over the past year, the Parallel Data Lab has worked with other members of the Scalable I/O Initiative (SIO) to consider issues in providing effective I/O performance for large-scale parallel systems. As a forum for parallel I/O researchers, the SIO Initiative gives the PDL fertile ground for interacting with other researchers, industry and the user community.

Within the SIO Initiative, five working groups have been busy exploring I/O-intensive applications, tools for evaluating performance, compiler and language support for I/O, operating- and file-systems questions, integration issues, and testbeds. At meetings around the country, each of the working groups has reported results and brainstormed future directions. The Operating and File Systems group, led by Kai Li from Princeton and including PDL members Jim Zelenka, Chris Demetriou, Garth Gibson, and Erik Riedel, has written a Proposal for a Common Parallel File System Programming Interface, also known as the low-level PFS API.

The group decided at the outset to focus on two levels of interface. For the lowest level, the group chose to specify an API with only the most basic functions necessary to take full advantage of the underlying I/O hardware. At the highest level, another API and application-specific libraries, built on the low-level API, will provide application-level interfaces.

In deciding which functions to include in the low-level API, the group always asked, "Can this function be implemented just as efficiently at a higher level?" If the answer was "Yes," then the function in question was left for library writers working above the portable low-level API. If the answer was "No, performance requires this function at the lowest level," then they added the function into the proposal for the low-level API. The API proposal currently provides functions for basic file management, scatter/gather read and write, asynchronous I/O, file-access hints, client cache control, control operations for inquiring about and specifying file-system parameters, collective I/O, and a fast copy operation.

The API document was largely created in collaboration with the IBM T. J. Watson Research Center, the PDL, Princeton University, the University of Arizona, and the IBM PowerParallel Division. It has been reviewed by the Operating and File Systems group several times and presented twice to the entire SIO Initiative. The current document is available for review from the PDL Web pages and will be publicly discussed in a roundtable session at the upcoming Supercomputing'96 conference in Pittsburgh, Nov. 17-18, 1996.

Recent Publications Continued

Understanding Customer Dissatisfaction With Underutilized Distributed File Servers

In *Proceedings of the Fifth NASA Goddard Space Flight Center Conference on Mass Storage Systems and Technologies.* College Park, MD. September 17-18, 1996. Also available as technical report CMU-CS-96-158.

Abstract

An important trend in the design of storage subsystems is a move toward direct network attachment. Network-attached storage offers the opportunity to off-load distributed file system functionality from dedicated file server machines and execute many requests directly at the storage devices. For this strategy to lead to better performance as perceived by users, the response time of distributed operations must improve.

In this paper, we analyze measurements of an Andrew File System (AFS) server that we recently upgraded in an effort to improve client performance in our laboratory. While the original server's overall utilization was only about 3%, we show how burst loads were sufficiently intense to lead to periods of poor response time significant enough to trigger customer dissatisfaction. In particular, we show how, after adjusting for network load and traffic to non-project servers, 50% of the variation in client response time was explained by variation in server CPU utilization. That is, clients saw long response times in large part because the server was often over-utilized when it was used at all. Using these measures, we see that off-loading file server work in a network-attached storage architecture has the potential to benefit user response time.

Computational power in such a system scales directly with storage capacity, so the slowdown during burst periods should be reduced.