THE

PDL Packet

THE NEWSLETTER ON PDL ACTIVITIES & EVENTS • FALL 2001

http://www.pdl.cmu.edu/

PARALLEL DATA LABORATORY

CARNEGIE MELLON UNIVERSITY

AN INFORMAL PUBLICATION FROM ACADEMIA'S PREMIERE STORAGE SYSTEMS RESEARCH CENTER DEVOTED TO ADVANCING THE STATE OF THE ART IN STORAGE SYSTEMS AND INFORMATION INFRASTRUCTURES.

CONTENTS

CONSORTIUM MEMBERS

EMC Corporation
Hewlett Packard Labs
Hitachi, Ltd.
IBM
Intel Corporation
LSI Logic
Lucent Technologies
Network Appliance
Panasas, Inc.
Platys Communications
Seagate Technology
Snap Appliances
Sun Microsystems
Veritas Software Corp.

Blurring the Line Between Operating Systems and Storage Devices

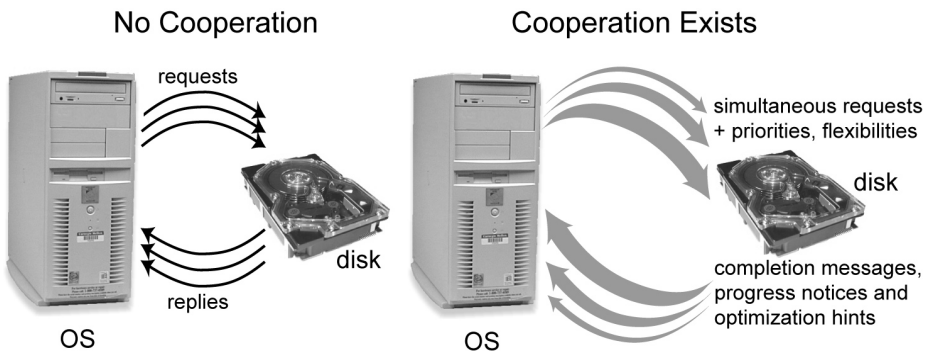
Greg Ganger & Joan Digney

Getting Disks and Operating Systems to Cooperate

Independently, both device firmware and OS software engineers aggressively utilize their knowledge and resources to mitigate disk performance problems. At the same time, the disk firmware folks complain of short request queues; frustrating to them because they can do efficiency-scheduling better than host software, while the file system people have given up on detailed data placement, focusing instead on just trying to use "large" requests to amortize positioning times over more data transfer. An overall goal of some recent PDL projects is to increase the cooperation between these two sets of engineers, significantly increasing the end-to-end performance and robustness of the system as a whole.

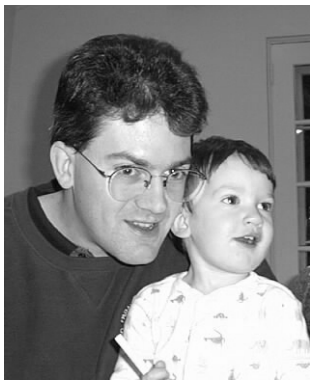
The fundamental problem is that the storage interface hides details from both sides and prevents communication. For example, storage devices can schedule requests to maximize efficiency, but host software tends not to expose much request concurrency because the device firmware does not know about host priorities and considers only efficiency. Likewise, host software can place data and thus affect request locality in a variety of ways, but currently does so with only a crude understanding of device strengths and weaknesses, because detailed device knowledge is not available.

All of these difficulties could be avoided by allowing the host software and device firmware to exchange information. The host software knows the relative



Both systems allow the host OS to make simultaneous requests to the disk, but cooperative interfaces also allow the host to tell the disk what is important and what options are acceptable (e.g., read these 10 blocks in any order or write to one of these 3 places). With this information, the disk can better specialize its actions to host needs. Cooperative interfaces also allow the disk to tell the host OS about data layout and access patterns that will work particularly well or particularly badly. The host OS can then tune its policies to match storage device strengths and avoid weaknesses.

... continued on pg. 12



FROM THE DIRECTOR'S CHAIR

Greg Ganger

Hello from fabulous Pittsburgh!

2001 has been another exciting year in the Parallel Data Lab. We were joined by three new faculty (Natassa Ailamaki, Mike Reiter, and Chenxi Wang) and several new students and staff. Three students won Fellowships (one from Intel, one from IBM, and one from the USENIX Association), and several spent the summer with PDL Consortium companies. Our big proposals received government funding, providing a solid foundation for PDL's financial stability. We prototyped and taught a new storage systems class. And, most importantly, there is our research.

The PDL continues to pursue a broad array of storage systems research, from the underlying devices to the applications that rely on storage. The past year brought excellent progress in existing projects and the initiation of exciting new ones. Let me highlight a few things.

Last year, we proposed freeblock scheduling as a new approach to utilizing more of a disk's potential media bandwidth. This year, we developed a software-only implementation that works from outside the disk drive, achieving 65% of the potential free bandwidth. As a result, 15 to 35% of the disk's total media bandwidth can be provided to background tasks with a less than 2% increase in foreground response times. We are excited that an outside-the-disk freeblock scheduler works, and we believe that disk firmware can do even better; we are working with Seagate to explore in-firmware implementations.

Database systems have suddenly become a strength area in the PDL. Of course, Christos Faloutsos remains well-known for his contributions. Also, in the past couple of years, Todd Mowry's group began exploring architectural support for database systems. This year, Natassa Ailamaki came to Carnegie Mellon and joined the PDL, significantly building up our database systems expertise. Both Natassa's and Todd's groups published excellent papers in the top 2001 database systems conferences – Natassa's VLDB paper was named Best Paper and Shimin Chen's SIGMOD paper was runner-up for the award. I am very excited with these developments, and I expect database systems will be a growing PDL focus in years to come.

PDL's survivable storage project (PASIS) continues to explore distributed storage systems that can survive failures and malicious attacks by encoding and distributing data across independent servers. In addition to building a prototype, we are exploring the trade-offs among the designs being promoted by various researchers. Most such designs differ mainly in their choice of data distribution scheme (how data is encoded and broken into pieces). We have developed and are refining an approach to evaluating and visualizing the trade-offs among performance, security, and availability inherent to this choice. The visualization, in particular, exposes a lot of the interesting features of the trade-off space.

PDL's self-securing storage and NIC-based firewall projects have grown and merged into a new vision of infrastructure security based on *self-securing devices.* The article on page 8 describes the self-securing device concept. In a nutshell, each device is augmented with relevant security functionality and made intrusion-independent from the OS and other devices. We think that this architecture will make systems more intrusion-tolerant and more manageable

… continued on pg. 3

THE PDL PACKET

The Parallel Data Laboratory
School of Computer Science
Deptartment of ECE
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
VOICE 412•268•6716
FAX 412•268•3010

PUBLISHERS
Greg Ganger & David Nagle

EDITOR
Joan Digney

The PDL Packet is published once per year and provided to members of the Parallel Data Consortium. Copies are given to other researchers in industry and academia as well. A pdf version resides in the Publications section of the PDL Web pages and may be freely distributed. 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. It is presently being run as a luxury hotel.

CONTACTING US

WEB PAGES

PDL Home: http://www.pdl.cmu.edu/

Please see our web pages at
http://www.pdl.cmu.edu/PEOPLE/
for further contact information.

FACULTY

Greg Ganger (director)
412•268•1297
greg.ganger@ece.cmu.edu

David Nagle (director)
david.nagle@cs.cmu.edu

Anastassia Ailamaki
natassa@cs.cmu.edu

Christos Faloutsos
christos.faloutsos@cs.cmu.edu

Garth Gibson
garth.gibson@cs.cmu.edu

Seth Goldstein
seth.goldstein@cs.cmu.edu

Mor Harchol-Balter
mor.harchol-balter@cs.cmu.edu

Todd Mowry
todd.mowry@cs.cmu.edu

Mike Reiter
reiter@ece.cmu.edu

Srinivasan Seshan
srini@cs.cmu.edu

Chenxi Wang
chenxi@ece.cmu.edu

Hui Zhang
hui.zhang@cs.cmu.edu

STAFF MEMBERS

Karen Lindenfelser
(pdl business administrator)
412•268•6716
karen@ece.cmu.edu

Mike Bigrigg
John Bucy
Joan Digney
Gregg Economou
Semih Oguz
Melissa Puryear
Joseph Slember
Ken Tew

GRADUATE STUDENTS

Mukesh Agrawal
Aditya Akella
Mehmet Bakkaloglu
Hemant Bhanoo
Angela Demke Brown
Shimin Chen
David Friedman
Garth Goodson
John Griffin
Stavros Harizopoulos
Andrew Klosterman
Chris Lumb
Amit Manjhi
Vijay Pandurangan
Efstratios Papadomanolakis
David Petrou
Asad Samar
Jiri Schindler
Steve Schlosser
Craig Soules
John Strunk
Mengzhi Wang
Ted Wong
Jay Wylie
Shuheng Zhou

FROM THE DIRECTOR'S CHAIR

… continued from pg. 2

when under attack. This vision brings with it many interesting challenges and a healthy source of funding, so stay tuned.

PDL's strength in security received a big protein boost with the arrivals of Mike Reiter and Chenxi Wang. Both of these new PDL faculty are security experts; please find more details about them in their biographies starting below.

Other PDL projects are also producing exciting results. For example, the DIXtrac disk characterization tool has been used to explore the use of disk-specific knowledge in file systems. The CHIPS center continues to develop hardware and process technologies to realize MEMS-based storage devices, and PDL researchers are looking at reliability issues and system-level performance issues. For the latter, we developed a timing-accurate storage emulator that looks to systems like a real MEMS-based storage device. The compiler-directed prefetching work has developed adaptive software pipelining techniques that do a better job of hiding disk latencies. The bandwidth management for IP Networks project is using smart storage-over-IP "host-bus" adapters to provide important storage flows (e.g., backup or reconstruction) with the bandwidth they need while avoiding specialized (costly) routers and starvation of other flows. This newsletter and the PDL website offer more details and additional research highlights.

I'm always overwhelmed by the accomplishments of the PDL students and staff, and it's a pleasure to work with them. Given this year's growth and progress, I look forward to great things to come.

NEW PDL FACULTY

Chenxi Wang

Chenxi is a new research faculty member of Electrical and Computer Engineering at CMU. She holds a Ph.D. in Computer Science from the University of Virginia. Her research interests include security solutions for large-scale, distributed systems, and security issues with respect to software engineering. Chenxi held a research associate position at Citibank from 1996 to 1997. She is the recipient of ACM's DC Chapter's 1999 Samuel Alexander award for Doctoral Candidates and the 1999 outstanding student research award from the Department of Computer Science at the University of Virginia. Chenxi served on the program committee for ACM's New Security Paradigms workshop from 1997 to 1999. She is the work-in-progress chair for ACM's ACSAC 2001 and serves on an advisory committee for the FCC.

Mike Reiter

Michael Reiter is a Professor of ECE and CS at CMU. He received his BS degree in mathematical sciences from the University of North Carolina in 1989, and his MS and Ph.D. degrees in computer science from Cornell University in 1991 and 1993, respectively. He joined AT&T Bell Labs in 1993 and became a founding member of AT&T Labs Research when NCR and Lucent Technologies (including Bell Labs) split from AT&T in 1996. He returned to Bell Labs in 1998 to become Director of Secure Systems Research.

… continued on pg. 4

YEAR IN REVIEW

November 2001
- Ninth Annual PDL Retreat & Workshop.

October 2001
- Jim Gray of Microsoft Research visits the Database seminar to speak on "Rules of Thumb in Data Engineering."
- Mike Bigrigg on "Testing the Portability of Desktop Applications to a Networked Embedded System" at the Workshop on Reliable Embedded Systems, in conjunction with the 20th IEEE SRDS in New Orleans, LA.
- Mike Reiter comes to CMU.

September 2001
- SDI speaker and PDL Alumni Howard Gobioff, of Google on "Google - A Systems Overview."
- Natassa Ailamaki attended the 27th International Conference on Very Large Data Bases (VLDB) speaking on "Weaving Relations For Cache Performance" (and received Best Paper Award) in Rome, Italy.
- Chenxi Wang comes to CMU.
- Three PDL papers accepted at FAST 2002.

August 2001
- SDI speaker PDL Alumni Erik Riedel, of HP Labs on "A Framework for Evaluating Storage System Security."

July 2001
- John Griffin spent the summer working with Liddy Shriver and Phil Bohannon at Bell Laboratories in Murray Hill, NJ.
- Steve Sclosser worked with EMC's Performance Group this summer.
- Craig Soules interned with the K42 operating system group at IBM's T.J. Watson research center.

June 2001
- Greg attended the USENIX Technical Conference in Boston as a program committee member and the chair of the Work In Progress session.

May 2001
- Third annual PDL Spring Open House, this year in partnership with the Center for Highly Integrated Information Processing and Storage Systems (CHIPS).
- Greg spoke on "Better Security via Smarter Devices" at HotOS-VIII in Germany.
- Thesis proposal: Ted Wong on "Dynamic Decentralized Recovery for Survivable Storage Systems."

April 2001
- SDI Speaker Ted Wong, SCS, on "My Cache or Yours?"

February 2001
- SDI Speaker Eitan Bachmat, of EMC Corporation on "The Merits of the IRM Model as a Replacement for Trace Driven Simulations in System Design."

January 2001
- Natassa Ailamaki attended the 4th CAECW speaking on "Walking Four Machines By the Shore" in Monterrey, Mexico.

December 2000
- SDI Speaker Liddy Shriver, of Bell Laboratories on "Storage Management for Web Proxies."

November 2000
- Eighth Annual PDS Retreat & Workshop.
- Steve Schlosser on "Designing Computer Systems with MEMS-based Storage" at ASPLOS-IX, Cambridge, MA.

NEW PDL FACULTY

… continued from pg. 3

Dr. Reiter's research interests include all areas of computer and communications security, and distributed computing. He has served as Program Chair of the flagship computer security conferences of both the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronic Engineers (IEEE), and serves on numerous other conference program committees in the areas of computer security and distributed computing. He presently serves on the editorial boards of ACM Transactions on Information and System Security, IEEE Transactions on Software Engineering, and the International Journal of Information Security.

Anastassia Ailamaki

Anastassia Ailamaki received her Ph.D. from the Computer Sciences Dept. of the University of Wisconsin-Madison in November 2000 and joined the School of Computer Science Faculty at CMU this spring. Anastassia's research focuses on the interaction of DBMSs with computer architecture, using innovative interdisciplinary methods to improve DBMS performance by studying the hardware behavior of modern commercial systems. Currently, she is designing indexing data structures and data/instruction placement algorithms to achieve optimal cache and memory bandwidth utilization. On multiprocessor platforms, her work focuses on cache coherence protocols as they influence DBMS behavior. Her long-term goal is to explore the interaction between database systems and other related areas in order to improve performance and expand functionality. She is also interested in making application logic a first class citizen in database systems, and has demonstrated that active DBMS functionality can be used to fully support scientific workflows modeling soil science and biochemistry experiments.

PASTENSE: a Fast Start-up Algorithm for Scalable Video Libraries

Harizopoulos & Gibson

Carnegie Mellon University Technical Report CMU-CS-01-105, March, 2001.

Striping video clip data over many physical resources (typically disk drives) balances video server load with less data replication. Current striped video delivery algorithms can have high start-up latency if the load is high. We propose a new, fast start-up algorithm, PASTENSE. This algorithm minimizes start-up latency by using aggressive prefetching to exploit disk idle time, and using available RAM to dynamically optimize the newly requested video's schedule. Our proposed method (a) does not require changes in the existing striped data placement, (b) it never performs worse than alternate designs, and (c) it achieves significant benefits: up to 9 times faster start-up times under high loads.

Enabling Dynamic Security Management of via Device-Embedded Security

Ganger & Nagle

Carnegie Mellon University Technical Report CMU-CS-00-174, December 2000.

This report contains the technical content of a recent funding proposal. In it, we propose a new approach to network security in which each individual device erects its own security perimeter and defends its own critical resources. Together with conventional border defenses (e.g., firewalls and OS kernels), such *self-securing devices* provide a flexible infrastructure for dynamic prevention, detection, diagnosis, isolation, and repair of successful breaches in borders and device security perimeters.

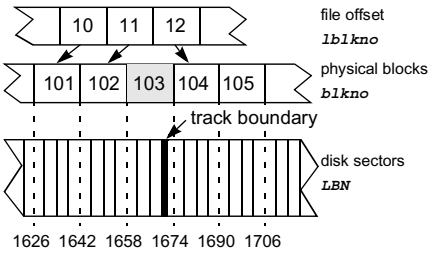
Managing network security is difficult in current systems, because a small number of border protections are used to protect a large number of resources. We plan to explore the fundamental principles and practical costs/benefits of embedding security functionality into infrastructural devices, such as network interface cards (NICs), network-attached storage (NAS) devices, video surveillance equipment, and network switches and routers. The report offers several examples of how different devices might be extended with embedded security functionality and outlines some challenge of designing and managing self-securing devices.

Track-Aligned Extents: Matching Access Patterns to Disk Drive Characteristics

Schindler, Griffin, Lumb & Ganger

To appear, Conference on File and Storage Technologies (FAST) January 28-30, 2002. Monterey, CA.

Track-aligned extents (traxtents) utilize disk-specific knowledge to match access patterns to the strengths of modern disks. By allocating and accessing related data on disk track boundaries, a system can avoid most rotational latency and track crossing overheads. Avoiding these overheads can increase disk access efficiency by up to 50% for mid-sized requests (100-500 KB). This paper describes *traxtents*, algorithms for detecting track boundaries, and the use of traxtents in file systems and video servers. For large file workloads, a modified version of FreeBSD's FFS implementation reduces application run times by 20% compared to the original version. A video server using traxtent-based requests can support 56% more concurrent streams at the same startup latency and buffer space. For LFS, 44% lower overall write cost for track-sized segments can be achieved.



Mapping system-level blocks to disk sectors. Physical block 101 maps directly to disk sectors 1626-1641. Block 103 is an excluded block because it spans the disk track boundary between LBNs 1669-1670.

Data Mining Meets Performance Evaluation: Fast Algorithms for Modeling Bursty Traffic

Wang, Madhyastha, Chan, Papadimitriou & Faloutsos

To appear 18th International Conference on Data Engineering (ICDE) 2002, San Jose, CA.

Network, web, and disk I/O traffic are usually bursty, self-similar and therefore cannot be modeled adequately with Poisson arrivals. However, we do want to model these types of traffic and to generate realistic traces, because of obvious applications for disk scheduling, network management, web server design. Previous models (like fractional Brownian motion, ARFIMA etc.) tried to capture the 'burstiness'. However the proposed models either require too many parameters to fit and/or require prohibitively large (quadratic) time to generate large traces. We propose a simple, parsimonious method, the b-model, which solves both problems: it requires just one parameter, *b*, and it can easily generate large traces. In addition, it has many more attrac-

… continued on pg. 6

… continued from pg. 5

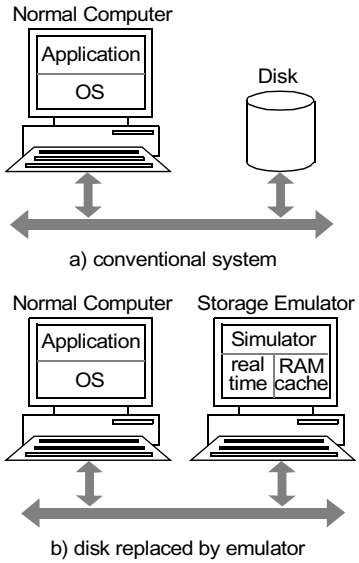
tive properties: (a) with our proposed estimation algorithm, it requires just a single pass over the actual trace to estimate *b*. For example, a one-day-long disk trace in milliseconds contains about 86Mb data points and requires about 3 minutes for model fitting and 5 minutes for generation, and (b) the resulting synthetic traces are very realistic: our experiments on real disk and web traces show that our synthetic traces match the real ones very well in terms of queuing behavior.

Timing-Accurate Storage Emulation

Griffin, Schindler, Schlosser & Ganger

To appear, Conference on File and Storage Technologies (FAST), January 28-30, 2002. Monterey, CA.

Timing-accurate storage emulation fills an important hole in the set of common performance evaluation techniques for proposed storage designs: it allows a researcher to experiment with not-yet-existing storage components in the context of real systems executing real applications. As its name suggests, a timing-accurate storage emulator appears to the system to be a real storage component with service times matching a simulation model of that component. This paper promotes timing-accurate storage emulation by describing its unique features, demonstrating its feasibility, and illustrating its value. A prototype, called the Memulator, is described and shown to produce service times within 2% of those computed by its component simulator for over 99% of requests. Two sets of measurements enabled by the Memulator illustrate its power: (1) application performance on a modern Linux system equipped with a MEMS-based storage device (no such device exists at this time), and (2) application performance on a modern Linux system equipped with a disk whose firmware has been modified (we have no access to firmware source code).



A system with (a) real storage or (b) emulated storage. The emulator transparently replaces storage devices in a real system. By reporting request completions at the correct times, the performance of different devices can be mimicked, enabling full system-level evaluations of proposed storage subsystem modifications.

Tri-Plots: Scalable Tools for Multidimensional Data Mining

Traina, Traina, Papadimitriou & Faloutsos

7th ACM Intl. Conference on Knowledge Discovery and Data Mining (KDD) 2001, San Francisco, CA, August 2001.

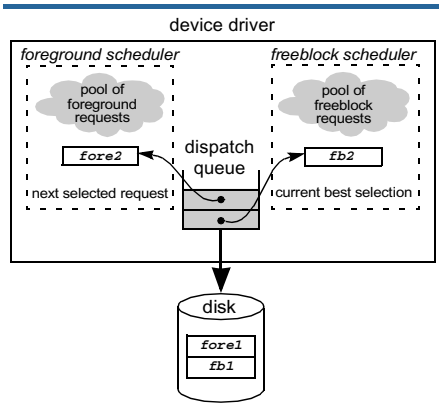
We focus on the problem of finding patterns across two large, multidimensional datasets. For example, given feature vectors of healthy and of non-healthy patients, we want to answer the following questions: Are the two clouds of points separable? What is the smallest/largest pairwise distance across the two datasets? Which of the two clouds does a new point (feature vector) come from? We propose a new tool, the tri-plot, and its generalization, the pq-plot, which help us answer the above questions. We provide a set of rules on how to interpret a tri-plot, and we apply these rules on synthetic and real datasets. We also show how to use our tool for classification, when traditional methods (nearest neighbor, classification trees) may fail.

Freeblock Scheduling Outside of Disk Firmware

Lumb, Schindler & Ganger

To appear, Conference on File and Storage Technologies (FAST), January 28-30, 2002. Monterey, CA.

Freeblock scheduling replaces a disk drive's rotational latency delays with useful background media transfers, potentially allowing background disk I/O to occur with no impact on foreground service times. To do so, a freeblock scheduler must be able to very accurately predict the service time components of any given disk request – the necessary accuracy was not previously considered achievable outside of disk firmware. This paper describes the design and implementation of a working external freeblock scheduler running either as a user-level application atop Linux or inside the FreeBSD kernel. This freeblock scheduler can



Freeblock scheduling inside a device driver.

… continued on pg. 7

… continued from pg. 6

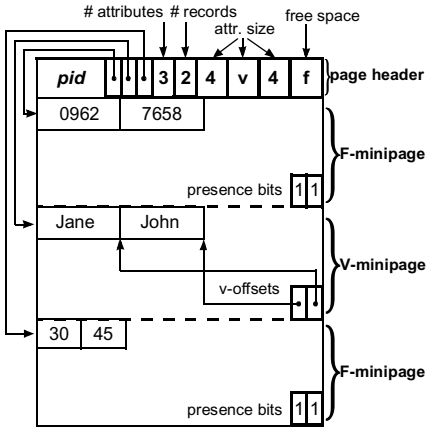
give 15% of a disk's potential bandwidth (over 3.1MB/s) to a background disk scanning task with almost no impact (less than 2%) on the foreground request response times. This increases disk bandwidth utilization by over 6x.

Weaving Relations For Cache Performance

Ailamaki, DeWitt, Hill & Skounakis

The 27th International Conference on Very Large Data Bases (VLDB), Rome, Italy, September 2001

Relational database systems have traditionally optimzed for I/O performance and organized records sequentially on disk pages using the N-ary Storage Model (NSM) (a.k.a., slotted pages). Recent research, however, indicates that cache utilization and performance is becoming increasingly important on modern platforms. In this paper, we first demonstrate that in-page data placement is the key to high cache performance and that NSM exhibits low cache utilization on modern platforms. Next, we propose a new data organization model called PAX (Partition Attributes Across), that significantly improves cache performance by grouping together all values of each attribute within each page. Because PAX only affects layout inside the pages, it incurs no storage penalty and does not affect I/O behavior. According to our experimental results, when compared to NSM (a) PAX exhibits superior cache and memory bandwidth utilization, saving at least 75% of NSM's stall time due to data cache accesses, (b) range selection queries and updates on memory-resident relations execute 17-25% faster, and (c) TPC-H queries involving I/O execute 11-48% faster.



An example PAX page.

Walking Four Machines By the Shore

Ailamaki, DeWitt & Hill

The 4th Workshop on Computer Architecture Evaluation using Commercial Workloads. Monterrey, Mexico, January 2001

Conceptually, all of today's processors follow the same sequence of logical operations when executing a program. Nevertheless, there are internal implementation details that critically affect the processor's performance, and vary both within and across compute vendor products. To accurately identify the impact of variation in processor and memory subsystem design on DBMS performance, we need to identify the impact of the microarchitectural parameters on the performance of database management systems.

This study compares the behavior of a prototype database system built on top of the Shore storage manager across three different processor design philosophies. In order to evaluate the different design decisions and trade-offs in the execution engine and memory subsystems of the above processors, we ran several range selections and decision-support queries on a memory-resident TPC-H dataset. The insights gained are indications that, provided that there are no serious hardware implementation concerns, DSS would exploit the following designs towards higher performance: 1. A processor design that employs (a) out-of-order execution to more aggressively overlap stalls, (b) a high-accuracy branch prediction mechanism, and (c) the opportunity to execute more than one load/store instruction per cycle, and 2. A memory hierarchy with (a) non-inclusive (at least for instructions) caches (b) a large (>2MB) second-level cache, and (c) a large cache block size (64-128 bytes) without sub-blocking, to exploit spatial locality.

System Design Considerations for MEMS Actuated Magnetic-Probe Based Mass Storage

Carley, Ganger, Guillou & Nagle

IEEE Transactions on Magnetics, January 2001.

This paper presents common system design considerations imposed on magnetic storage devices that employ MEMS devices for positioning of a magnetic probe device over a magnetic media. The paper demonstrates that active servo control of the probe tip to media separation can be achieved with sub-nanometer accuracy. It also demonstrates that reasonable-size capacitive sensors can resolve probe tip motions with a noise floor of roughly 22 picometers, allowing them to be used as position sensors in magnetic force microscope (MFM) readout approaches. In addition, this paper demonstrates that although MEMS media actuators can achieve scanning ranges of ∓ 50 µm, the mass of the media sled imposes important access time and data rate constraints on such MEMS-actuated mass storage devices.

… continued on pg. 17

BETTER SECURITY VIA SMARTER DEVICES

Greg Ganger & David Nagle

Despite enormous effort and investment, it has proven nearly impossible to prevent computer security breaches. Between our growing dependence upon on-line information and wide-area networking, an enormous security risk to our national economic and defense infrastructures exists. To protect critical information infrastructures, we need defenses that can survive determined and successful attacks, allowing security managers to dynamically detect, diagnose, and recover from breaches in security perimeters.

To attack the security dilemma, the PDL has embarked on a long-term research effort to re-architect computer systems into *"Self-Securing Devices."* Funded by the Department of Defense's Critical Infrastructure Protection program for $4.7 million over 5 years, PDL draws on our expertise in Network-Attached Storage, Self-Securing Storage, PASIS, and Scalable Firewalls, to promote a security architecture where individual system components erect their own security perimeters and protect their resources (e.g., network, storage, or video feed) from intruder tampering. The "self-securing devices" architecture distributes security functionality amongst *physically distinct* components, avoiding much of the fragility and manageability inherent in today's border-based security.

Specifically, this new architecture addresses three fundamental difficulties: it simplifies each security perimeter (e.g., consider NIC or disk interfaces), it reduces the power that an intruder gains from compromising just one of the perimeters, and (3) it distributes security enforcement checks among the many components of the system.

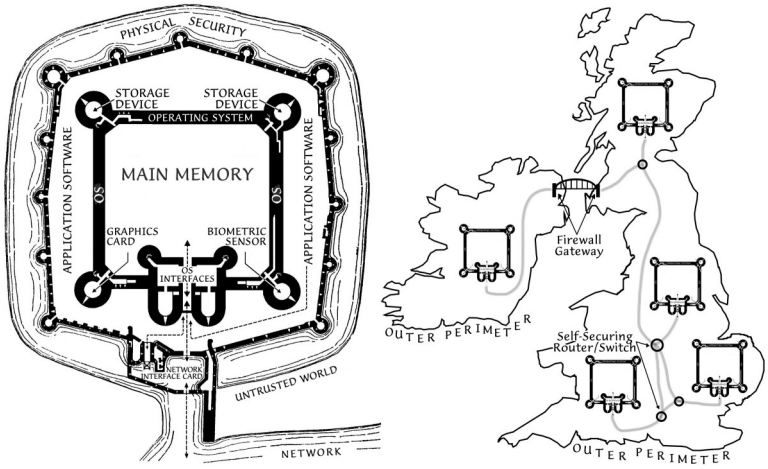
Current security mechanisms are based largely on singular border protections. This roughly corresponds to defense practices during Roman times, when defenders erected walls around their camps and homes to provide protective cover during attacks. Once inside the walls, however, attackers faced few obstacles to gaining access to all parts of the enclosed area. Likewise, a cracker who successfully compromises a firewall or OS has complete access to the resources protected by these border defenses. Of course, border defenses were a large improvement over open camps, but they proved difficult to maintain against determined attackers – border protections can be worn down over time and defenders are often spread thin at the outer wall.

As the size and sophistication of attacking forces grew, so did the sophistication of defensive structures. The most impressive such structures, constructed to withstand determined sieges in medieval times, used multiple tiers of defenses. Further, tiers were not strictly hierarchical in nature – rather, some structures could be defended independently of others. This major advancement in defense capabilities provided defenders with significant flexibility in defense strategy, the ability to observe attacker activities, and the ability to force attackers to deal with multiple independent defensive forces.

Applying the same ideas to computer and network security, border protections (i.e., firewalls and host OSes) can be augmented with security perimeters erected at many points within the borders. Enabled by low-cost computation (e.g., embedded processors, ASICs), security functionality can be embedded in most device microcontrollers, yielding "better security via smarter devices." We refer to devices with embedded security functionality as *self-securing devices* (see figure).

Self-securing devices can significantly increase network security and manageability, enabling capabilities that are difficult or impossible to implement in current systems. For example, independent device-embedded security perimeters guarantee that a penetrated boundary does not compromise the entire system. Uncompromised components continue their security functions even when other system components are compromised. Further, when attackers penetrate one boundary and then attempt to penetrate another,

… continued on pg. 9



The self-securing device architecture illustrated via the siege warfare constructs that inspired it. On the left, (a) shows a siege-ready system with layered and independent tiers of defense enabled by device-embedded security perimeters. On the right, (b) shows two small intranets of such systems, separated by firewall-guarded entry points. Also note the self-securing routers/switches connecting the machines within each intranet.

… continued from pg. 8

uncompromised components can observe and react to the intruder's attack; from behind their intact security perimeters, they can send alerts to the security administrator, actively quarantine or immobilize the attacker, and wall-off or migrate critical data and resources. Pragmatically, each self-securing device's security perimeter is simpler because of specialization, which should make correct implementations more likely. Further, distributing security checks among many devices reduces their performance impact and allows more checks to be made.

By augmenting conventional border protections with self-securing devices, substantial increases in both network security and security manageability can result. As with medieval fortresses, well-defended systems conforming to this architecture could survive protracted sieges by organized attackers.

Device-Embedded Security Examples

Network Interface Cards: NICs in computer systems move packets between the system's components and the network. Thus, the natural security extension is to enforce security policies on packets forwarded in each direction. Like a firewall, a self-securing NIC does this by examining packet headers and simply not forwarding unacceptable packets into or out of the computer system. A self-securing NIC can also act as a machine-specific gateway proxy, achieving the corresponding protections without scalability or identification problems; by performing such functions at each system's NIC, one avoids the bottleneck imposed by current centralized approaches.

Storage Devices: The role of storage devices in computer systems is to persistently store data. Thus, the natural security extension is to protect stored data from attackers, preventing undetectable tampering and permanent deletion. Self-securing storage devices do this by managing storage space from behind its security perimeter, keeping an audit log of all requests, and keeping clean versions of data modified by attackers. Since a storage device cannot distinguish compromised user accounts from legitimate users, the latter requires keeping all versions of all data. Finite capacities limit how long such comprehensive versioning can be maintained, but 100% per year storage capacity growth will allow modern disks to keep several weeks of all versions. If intrusion detection mechanisms reveal an intrusion within this *detection window*, security administrators will have this valuable audit and version information for diagnosis and recovery.

Biometric Sensors: Biometric sensors provide input to biometric-enhanced authentication processes, which promise to distinguish between users based on measurements of their physical features. The natural security extension is to ensure the authenticity of the information provided to these processes. A self-securing sensor can do this by timestamping and digitally signing its sensor information. Such evidence of when and where readings were taken is critical to secure use of biometric information because, unlike passwords, biometrics are not secrets. For example, anyone can lift fingerprints from a laptop with the right tools or download facial images from a web page. Thus, evidence is needed to prevent straightforward forgery and replay attacks. Powerful self-securing sensors may also be able to increase security and privacy by performing the identity verification step from within their security perimeter and only exposing the results (with the evidence). By embedding mechanisms for demonstrating authenticity and timeliness inside sensor devices, one can verify sensor information (even over a network) even when intruders gain the ability to offer their own "sensor" data.

Graphical Displays: The role of graphical displays to visually present information to users. Thus, a natural security extension would be to ensure that critical information is displayed. A self-securing display could do this by allowing high-privilege entities to display data that cannot be overwritten or blocked by less-privileged entities. Thus, a security administrator could display a warning message when there is a problem in the system (e.g., a suspected trojan horse or a new e-mail virus that must not be opened). By embedding this screen control inside the display device, one gains the ability to ensure information visibility even when an intruder gains control over the window manager.

Routers and Switches: The role of routers and switches in a network environment is to forward packets from one link to an appropriate next link. Thus, a natural security extension for such devices is to provide firewall and proxy functionality. Many current routers already provide this. Some routers/switches also enhance security by isolating separate virtual LANs (VLANs). More dynamic defensive actions could provide even more defensive flexibility and strength. For example, the ability to dynamically change VLAN configurations would give security administrators the ability to create protected command and control channels in times of crisis or to quarantine areas suspected of compromise. When under attack, self-securing routers/switches could also initiate transparent replication of data services, greatly reducing the impact of denial-of-service attacks. Further, essential data sites could be replicated on-the-fly to "safe locations" or immediately isolated via VLANs to ensure security.

This article has been reprinted from our spring "PDL Updates" edition.

November 2001

Seagate Builds Experimental Drive for PDL Research

Over the summer, Seagate engineers developed a modified version of their high-end drive firmware to allow PDL researchers to experiment with in-drive scheduling algorithms. Specifically, the modified firmware will provide PDL researchers with hooks to try new algorithms, including forms of freeblock scheduling and other priority-based schemes. We thank Seagate for teaming up with us to develop these exciting research ideas.

October 2001

3 PDL Faculty Receive HP/Intel Itanium-Based System Grant*

Hewlett-Packard Company (HP) and Intel Corporation have awarded three Carnegie Mellon computer science faculty members $153,162 in equipment grants from their new Itanium-based Systems Grant program. Carnegie Mellon is one of 40 universities worldwide whose faculty was selected based on how they would deploy the Itanium-based systems to strengthen their research. One of the largest awards made under the HP/Intel program goes to Anastassia Ailamaki (PI), Todd Mowry and David Nagle. They will receive six one-way workstations, one two-way Workstation and one RX4610 server valued at $153,162. "The Itanium architecture is extremely promising for database applications – the software that dominates the commercial server market," said Ailamaki, an assistant professor of computer science. "Inventing ways for database systems to take advantage of the Itanium processor's characteristics opens an exciting new research area that will lead in revolutionizing database software technology to deliver superb performance on this cutting-edge computer architecture."

August 2001

Professor Srinivasan Seshan Receives IBM Faculty Partnership Award and NSF CAREER Award*

Srinivasan Seshan has been awarded an IBM Faculty Partnership Award (FPA) and an NSF CAREER award. An IBM FPA of $20,000 was granted for his work related to networking protocols and infrastructure for ubiquitous computing. He earned a $487,651 NSF Career Award for his proposal, "Towards an Efficient Ubiquitous Computing Infrastructure," in which he will attempt to design a new networking and operating system infrastructure for the next generation of ubiquitous computing applications. Congratulations Srini!

August, 2001

A New Mowry!

Todd Mowry and his wife Karen Clay are pleased to announce the arrival of their second son, Davis Foster Clay Mowry, born on August 15 at 12:23 a.m. Congratulations!

July 2001

Ailamaki Receives Carnegie Mellon Berkman Faculty Development Award and VLBD Best Paper*

Anastassia Ailamaki recently received a Carnegie Mellon Berkman Faculty Development Award for $5,000. Carnegie Mellon established the award to aid junior faculty in their professional development and provide funding for projects that have difficulty attracting outside support. Recently honored with the Best Paper Award in the prestigious International Conference on Very Large Data Bases (VLDB 2001), Ailamaki researches database management systems (DBMSs) and on their interaction with computer architecture. Her recent work is highly interdisciplinary and explores innovative ways to improve DBMS performance by studying the hardware behavior of commercial database systems running on modern processors.

July 2001

2 PDL Faculty Receive IBM Faculty Partnership Award*

Anastassia Ailamaki has received an IBM Faculty Partnership Award, totaling $40,000. Also receiving the award, for a third year, is PDL Director Greg Ganger. The IBM Faculty Partnership Award recognizes and fosters novel, creative work as well as strengthens the relationships between leading universities and the IBM research and development community.

June 2001

Computer Magazine Features Research on Active Disks for Large-Scale Data Processing*

In the June edition of Computer Magazine (Vol. 34, No. 6), Erik Riedel, ECE alumnus, Christos Faloutsos, professor, Garth A. Gibson, associate professor, and David Nagle, senior research computer scientist, report on their research on an active disk storage device that can accelerate an existing database system and eliminate the need for the PC processor.

From the online abstract: "With active disks, application-specific functions access the excess computation power in drives. Active disks combine the requisite processing power of general-purpose disk-drive micro-

… continued on pg. 11

… continued from pg. 10

processors with the special-purpose functionality of end-user programmability. The authors' experiment showed that active disks can accelerate an existing database system by moving data-intensive processing to the disks, thereby reducing the server CPU's processing load."

May 2001

Craig Soules Named USENIX Scholar for Second Year

Our congratulations to Craig Soules, a PDL graduate student working toward his Ph.D. in ECE. He was named a "USENIX Scholar" for a second year by the USENIX association. As a part of this award, USENIX is supporting Craig's tuition and stipend.

May 2001

Premier Database Research Conference ACM SIGMOD 2001 Recognizes CS Ph.D. Student's Work*

The paper "Improving Index Performance through Prefetching," by Shimin Chen, Phillip B. Gibbons, and Todd C. Mowry was selected as a runner-up for the Best Paper Award for 2001 at the ACM SIGMOD 2001 conference. The premier conference for database research and practice, ACM SIGMOD 2001 conference organizers received 290 paper submissions, but chose only 44 papers to be included in the proceedings. "The task for selection of best paper was extremely difficult," explained the ACM SIGMOD selection committee. "We ended up with three papers (out of 44) that were clearly the top ones, all deserving the best paper award. At the end, the decision was a very hard one to make; [Chen's] paper was finally ranked as one of the two runners-up." Chen is a second year CS Ph.D.; his advisor is Todd Mowry, associate professor of computer science. A copy of "Improving Index Performance through Prefetching" is linked from Chen's web page.

April 2001

Steve Schlosser wins Intel Fellowship

We'd like to congratulate Steve Schlosser, who has been awarded an Intel fellowship. Nationally, Intel awards thirty-five Ph.D. fellowships each year, providing a cash award (tuition/fees/stipend), an Intel CPU-based PC, an Intel Mentor, and the opportunity to conduct research or an internship at Intel.

Steve's research focuses on MEMS-based storage's design and application. This non-volatile storage technology merges magnetic recording material and thousands of recording heads to provide storage capacity of 1-10 GB of data in an area under 1 cm^2 in size with access times of less than a millisecond and streaming bandwidths of over 50 MB per second. Further, because MEMS-based storage is built using photolithographic IC processes similar to standard CMOS, MEMS-based storage has per-byte costs significantly lower than DRAM and access times an order of magnitude faster than conventional disks.

March 2001

EMC Donates $250K to PDL

We thank EMC for its generous donation of $250K (above and beyond membership fees) to support PDL research. Such donations enable PDL to initiate and seed its long-range research activities before government agencies are prepared to believe in them. Examples include the early MEMS-based storage work, early web server scheduling work, early new interfaces work, and early self-securing devices work.

March 2001

CMU's Grad Program in Computer Engineering Number 1†

In the annual U.S. News & World Report survey of engineering school deans, CMU's graduate program in Computer Engineering has been ranked *the* top program of its kind in the country. This is the first time that any of CMU's graduate or undergraduate programs have ranked number one. Carnegie Mellon's College of Engineering graduate programs overall once again ranked 8th among the nation's best.

March 2001

Mor Harchol-Balter Receives Anna McCandless Chair*

Mor Harchol-Balter, assistant professor of computer science, has been awarded the Anna McCandless Chair, a three-year term career development professorship that provides funding for travel and sabbaticals, including partial costs of academic-year teaching and research and programs. Jim Morris, dean of the School of Computer Science, said, "Mor arrived here, hit the ground running and has already launched an exciting program of research and education in computer system performance."

… continued on pg. 20

PROPOSALS & DEFENSES

THESIS PROPOSAL:

Decentralized Recovery for Survivable Storage Systems

Thesis proposal, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 2001

Theodore Wong, SCS

Survivable storage systems provide persistent, secure, and reliable access to data objects. These systems are designed to protect data against the failure or subversion of the backend storage devices. To achieve this, survivable storage systems employ sharing schemes to distribute shares of objects to the devices. Unfortunately, many current systems provide only static distribution of shares. Others support limited redistribution of shares by first reconstructing the original object. Survivable storage systems require a recovery service that guarantees persistent access to objects and that does not introduce single points of vulnerability into the system.

I claim that decentralized recovery services for survivable storage systems can execute in a secure and efficient manner. To demonstrate the validity of the thesis, I will implement a recovery service that regenerates shares lost to device failures, and repudiates shares compromised by adversaries. To replace lost shares, I will employ a reactive recovery protocol that does not require reconstruction of the original object. To repudiate stolen shares, I will use proactive recovery protocols that update shares of objects. In theory, the decentralized reactive and proactive protocols preserve the security properties of objects during recovery. Experimental results will demonstrate the efficiency of the decentralized recovery service.

OPERATING SYSTEMS & STORAGE DEVICES

… continued from pg. 1

importance of requests and has some ability to manipulate the locations that are accessed. The device firmware knows what the device hardware is capable of in general and what would be most efficient at any given point. Thus, the host software knows what is important and the device firmware knows what is fast. By exploring new storage interfaces and algorithms for exchanging and exploiting the collection of knowledge, and developing cooperation between devices and applications, we hope to eliminate redundant, guess-based optimization. The result would be storage systems that are simpler, faster, and more manageable.

Understanding When and How Cooperation Helps

For the past 15 years or so, the most common storage interfaces (SCSI and IDE) have consisted mainly of the same simple read and write commands. This consistent high-level interface has enabled great portability, interoperability, and flexibility for storage devices and their vendors. In particular, the resulting flexibility has allowed such very different devices as disk drives, solid state stores, and caching RAID systems to all appear the same to host operating systems.

In the continuing struggle to keep storage devices from becoming a bottleneck to system performance and thus functionality, system designers have developed many mechanisms for both storage device firmware and host OSes. These mechanisms have almost exclusively been restricted to only one side of the storage interface or the other. In fact, evolution on the two sides of the storage interface has reached the point where each has little idea of or input on the detailed operation of the other. We believe that this separation, which once enabled great advances, is now hindering the development of more cooperative mechanisms that consist of functionality on both sides of the interface.

The goal of cooperation between host software and device firmware raises a number of questions: (1) what should change in the host to better match device characteristics? (2) what should change in device firmware to better match what the host views as important? (3) how should the storage interface be extended to make these changes possible and effective? (4) how much device-specific information can be obtained and used from outside the device? and (5) how much complexity and overhead is involved with extending disk firmware functionality?

Over the years, the host-level software that manages storage devices has lost touch with the details of device mechanics and firmware algorithms. Unlike with many other components, however, these details can have dramatic, order-of-magnitude effects on system performance. Identifying specific examples where the host-level software can change in relatively small ways to better match device characteristics will represent one important step towards actually realizing greater cooperation. One example that we are exploring is Track-Aligned Extents. Several disk

… continued on pg. 13

… continued from pg. 12

drive advances in recent years have conspired to make the track a sweet spot in terms of access unit, but it only works when accesses are aligned on and sized to track boundaries. Accomplishing track-aligned extents in file systems will require several changes, including techniques for identifying the boundaries and file system support for variable sized allocation units. Our upcoming paper on this topic [Schindler01] shows that track-aligned extents can provide significant performance and predictability benefits for large file and video applications.

Agressive cache management and request scheduling policies available in today's systems, which typically go largely unused, could be active participants in scheduling if the firmware could differentiate between efficiency and system priorities. With minor extensions to the current storage interface, it should be possible to convey simple priority information to the device firmware. Freeblock Scheduling is one mechanism being explored for using this information in the firmware. By accurately predicting the rotational latencies of high-priority requests, it becomes possible to make progress on background activity with little or no impact on foreground request access times. Preliminary results [Lumb00] indicate that this can increase media bandwidth utilization by an order of magnitude and provide significant end-to-end performance improvements.

Our future research will explore interfaces and algorithms that allow even more cooperation between host software and device firmware. For example, we envision an interface that would allow the host system to direct the device to write a block to any of several locations (whichever is most efficient); the device would then return the resulting location, which would be recorded in the host's metadata structures. Such an interface would allow the host (e.g., database or file system) and the device to collaborate when making allocation decisions, resulting in greater efficiency but no loss of host control.

Another important practical research question that must be answered in pursuing this vision is how much device-specific knowledge and low-level control is available from outside the device firmware. In particular, if complete knowledge and control is available externally, then it may be unnecessary for host software to cooperate with device firmware – instead, the host software can directly do exactly what needs to be done, bypassing the firmware engineers entirely. Although we do not believe complete control to be possible, it is important to work on understanding how close one can get. An upcoming paper [Lumb01] describes our experiences with OS-level freeblock scheduling.

An equally important question relates to the practical limitations involved with working within disk firmware, for example those related to ASIC interactions, timeliness requirements, and limited runtime support. In some sense, this is not deep research, since disk manufacturers have been extending firmware for years. However, it is a critical practical consideration for any work that proposes extensions. We are working with Seagate to experiment with freeblock scheduling inside their disk firmware.

Example: Improving Host Software with Track-Aligned Extents

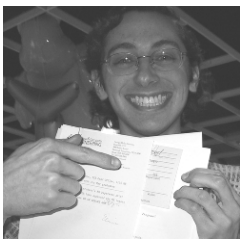
Modern host software (e.g., file systems or databases) performs aggressive on-disk placement and request coalescing, but generally do so with only a vague view of device characteristics – generally, the focus is on the notion that "bigger is better, because positioning delays are amortized over more data transfer." However, there do exist circumstances where considering specific boundaries can make a big difference.

Track-aligned extents is a new approach to matching system workloads to device characteristics and firmware enhancements. By placing

… continued on pg. 14

PDL giant comes to town.

David Friedman celebrates the first signature on his thesis.



… continued from pg. 13

and accessing largish data objects on track boundaries, one can avoid most rotational latency and track crossing overheads. Specifically, accessing a full track's data has two significant benefits: avoiding track switches and thus positioning delays, and eliminating rotational latency by accessing sectors on the media in the order that they pass under the read/write head instead of ascending LBN order. Thus, all sectors on a track can be accessed in a single rotation, regardless of which sector passes under the head first.

Combined, these benefits can increase large access efficiency by 25 to 55%, depending on seek penalties. They can also make disk access times much more predictable, which is important to real-time applications (e.g., video servers). However, these benefits are fully realized only for accesses that are track-sized and track-aligned. We believe that such accesses can be made much more common if file systems were modified to use track-aligned extents, specifically sized and aligned to match track boundaries.

Track-aligned extents are most valuable for workloads that involve many large requests to distinct portions of the disk. Video servers represent an ideal application. Although envisioned as a "streaming media," video storage access patterns show requests for relatively large segments of data read individually at a rate that allows the video to be displayed smoothly. A video server interleaves the segment fetches for several videos such that they all keep up. The result is non-contiguous requests, where the discontinuities are due to timeliness requirements rather than allocation decisions. The number of videos that can be played simultaneously depends both on the average-case performance and the bounds that can be placed on response times. Track-aligned extents can substantially increase video server throughput by making video segment fetches both more efficient and more predictable.

Example: Improving Disk Firmware with Freeblock Scheduling

Disk firmware includes support for aggressive scheduling of media accesses in order to maximize efficiency. In its current form, however, this scheduling concerns itself only with overall throughput. As a result, host software does not entrust disk firmware with scheduling decisions for large sets of mixed-priority requests. Freeblock scheduling is a new approach to media bandwidth utilization and request scheduling that uses the accurate predictions needed for aggressive scheduling to combine minimized response times for high-priority requests with improved efficiency and steady forward progress for lower-priority requests. Specifically, by interleaving low priority disk activity with the normal workload, freeblock scheduling replaces the rotational latency delays of high-priority requests with background media transfers. With appropriate freeblock scheduling, background tasks can receive 20 to 50% of a disk's potential media bandwidth without any increase in foreground request service times.

Fundamentally, the only time the disk head cannot be transferring data sectors to or from the media is during a seek. In fact, in most modern disk drives, the firmware will transfer a large request's data to or from the media "out of order" to minimize wasted time; this feature is sometimes referred to as zero-latency or immediate access. While seeks are unavoidable costs associated with accessing desired data locations, rotational latency is an artifact of not doing something more useful with the disk head. Since disk platters rotate constantly, a given sector will rotate past the disk head at a given time, independent of what the disk head is doing up until that time, offering an opportunity for something more useful to be done. Freeblock scheduling consists of predicting how much rotational latency will occur before the next foreground media transfer, squeezing some additional media transfers into that time, and still getting to the destination track in time for the foreground transfer.

Anticipated applications for freeblock scheduling include scanning large portions of disk contents, e.g. data mining of an active transaction processing systems, which showed that over 47 full scans per day of a 9GB disk can be made with no impact on OLTP performance, and internal storage optimization, e.g. placing related data contiguously for sequential disk access or segment cleaning in log-structured file systems which resulted in a 300% speedup for application benchmarks.

References

[Schindler01] *Track-aligned Extents: Matching Access Patterns to Disk Drive Characteristics*. Jiri Schindler, John Linwood Griffin, Christopher R. Lumb, Gregory R. Ganger. To appear, Conference on File and Storage Technologies (FAST), January 28-30, 2002, Monterey, CA.

[Lumb00] *Towards Higher Disk Head Utilization: Extracting "Free" Bandwidth From Busy Disk Drives*. Christopher R. Lumb, Jiri Schindler, Gregory R. Ganger, David F. Nagle and Erik Riedel. Proc. of the 4th Symposium on Operating Systems Design and Implementation, 2000.

[Lumb01] *Freeblock Scheduling Outside of Disk Firmware*. Christopher R. Lumb, Jiri Schindler, Gregory R.Ganger. To appear, Conference on File and Storage Technologies (FAST), January 28-30, 2002. Monterey, CA.

FACULTY

Please see biographies for our three new faculty members Anastassia Ailamaki, Mike Reiter and Chenxi Wang beginning on page 3.

STAFF

John Bucy joined the PDL this summer as a Systems Programmer to develop PDL's tools such as DIXtrac, DiskSim and Memulator.

Shelby Davis left us in August to join Panasas as a programmer.

Gregg Economou joined the PDL this summer as a Systems Programmer to work on the self-securing devices project.

Nat Lanza joined Panasas this spring as a programmer.

Paul Mazaitis has taken a leave of absence to try his hand at writing fiction.

Stephen Miller left the PDL in May after deciding to devote his full attention to his degree in Computer Science.

Nitin Parab left the PDL last fall to move to California.

Joseph Slember began working as research assistant on the PASIS project this spring.

Ken Tew joined the PDL this spring as a research programmer on the PASIS project. He came to us from the University of Pittsburgh.

GRAD STUDENTS

Mukesh Agrawal is a 2nd year Ph.D. student in CS exploring bandwidth-management for storage networks with Srini Seshan.

Aditya Akella is in his 2nd year in CS working on the Congestion Manager project with Srini.

Hemant Bhanoo is a Master's student working with Srini Seshan and Dave Nagle.

Lesley Leposo graduated with his MS in ECE and has moved on to industry.

Amit Manji has begun work on his Ph.D in CS with Todd Mowry and wants to finish before what he is working on becomes obsolete.

Efstratios Papadomanolakis is a new grad student in CS studying databases with Natassa Ailamaki.

Asad Samar has joined the PDL as a graduate student pursuing his Master's Degree in Computer Science.

UNDERGRADUATES

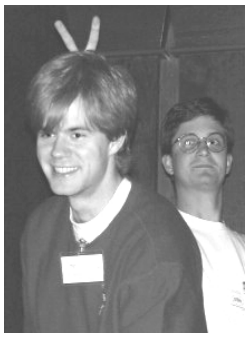
Russ Koenig is an undergrad programmer with the PDL while working on his degree in ECE.

Anand Patel is a freshman in Computer Science at CMU and is helping Karen as her office assistant.

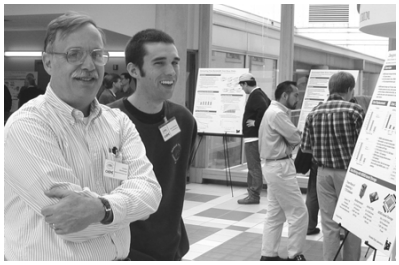
Jacob Vos has joined PASIS as an undergrad research programmer while completing his degree in Computer Science at the University of Pittsburgh.

Cory Williams has joined PASIS as an undergrad research programmer while continuing his degree in Computer and Mathematical Sciences at CMU.

PDL SPRING OPEN HOUSE



Poster session



Dinner with the PDL Consortium

A STORAGE MODEL TO BRIDGE THE PROCESSOR/MEMORY SPEED GAP*

Anastassia Ailamaki & Joan Digney

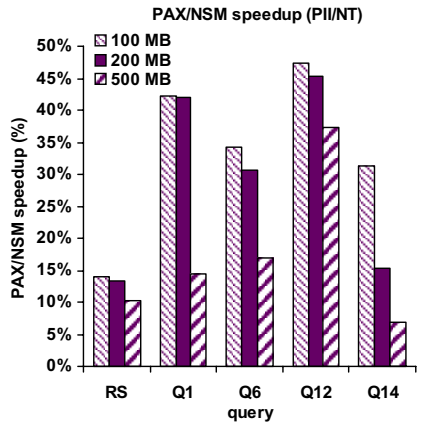
Memory speeds in today's computers have fundamentally lagged behind processor speeds. To alleviate the processor/memory performance gap, computer designers employ a hierarchy of cache memories (e.g., three levels in the recently announced IBM Power 4 processors), in which each level trades-off higher capacity for faster access times. As database applications become increasingly memory-intensive, high performance database management systems (DBMSs) must maximize cache utilization by keeping data that are likely to be referenced in the cache hierarchy. Ideally, the database application should run under the illusion that the database is *cache-resident*.

When optimizing cache utilization, data placement is extremely important. In commercial DBMSs, data misses in the cache hierarchy are a key memory bottleneck. The traditional data placement scheme used in DBMSs, the N-ary Storage Model (NSM), stores records contiguously starting from the beginning of each disk page, and uses an offset (slot) table at the end of the page to locate the beginning of each record. Each record has a record header that contains a null bitmap, offsets to the variable-length values, and other implementation-specific information. Each new record is typically inserted into the first available free space starting at the beginning of the page. Records may have variable length, therefore a pointer (offset) to the beginning of the new record is stored in the next available slot from the end of the page. The n^{th} record in a page is accessed by following the n^{th} pointer from the end of the page. Although NSM provides high intra-record locality, query operators often access only a small fraction of each record causing suboptimal cache behavior. Using NSM wastes bandwidth, pollutes the cache with useless data, and possibly forces replacement of information that may be needed in the future.

An alternative, the decomposition storage model (DSM) [3] partitions an *n*-attribute relation vertically into *n* sub-relations, each of which is accessed only when the corresponding attribute is needed. DSM maximizes inter-record locality; however, the DBMS must join the participating sub-relations together to reconstruct a record. When running queries that involve few attributes from each relation, DSM saves I/O and improves main memory utilization. As the number of participating attributes per relation increases, however, the record reconstruction cost dominates the query execution. For this reason, *all* of today's commercial database systems that we are aware of still use the traditional NSM algorithm for general-purpose data placement. The challenge is to repair NSM's cache behavior, without compromising its advantages over DSM.

Partition Attributes Across (PAX)

PAX is a new layout for data records that combines the best of the two worlds and exhibits superior performance by eliminating unnecessary accesses to main memory. For a given relation, PAX stores the same data as NSM on each page, requiring about the same amount of space as NSM. *Within* each page, however, PAX groups all the values of a particular attribute together on a minipage, increasing inter-record spatial locality with minimal impact on intra-record spatial locality. Each page also has a page header containing the number of attributes, the attribute sizes (for fixed length attributes), offsets to the beginning of the minipages, the current number of records on the page and the total space still available. During sequential scan, PAX fully utilizes the cache resources, because on each miss a number of the same attribute's values are loaded into the cache together. At the same time, all parts of the record are on the same page. To reconstruct a record one needs to perform a *mini-join* among minipages, which incurs minimal cost because it does not have to look beyond the page.



PAX/NSM speedup for DSS queries.

The figure above depicts PAX/NSM speedups when running range selections and four TPC-H queries against a 100, 200, and 500-MB TPC-H database on top of the Shore storage manager [2]. The detailed experimental setup and methodology are described elsewhere [1]. Decision-support systems are especially processor- and memory-bound, and PAX outperforms NSM for all these experiments. The speedups obtained are not constant across the experiments due to a combination of differing amounts of I/O and interactions between the hardware and the algorithms being used.

The results show that PAX outperforms NSM on all TPC-H queries in this workload. For instance, when compared to NSM, PAX reduces L2 cache data misses and stall time by a factor of four. Range selection queries and updates on main-memory tables execute in 17-25% less elapsed time. When running TPC-H queries that perform calculations on the data retrieved and require I/O,

… continued on pg. 17

A STORAGE MODEL TO BRIDGE THE PROCESSOR/MEMORY SPEED GAP*

… continued from pg. 16

PAX incurs a 11-48% speedup over NSM. When compared to DSM, PAX retains all the advantages of NSM and executes queries faster because it combines high cache performance with minimal reconstruction cost.

PAX has several additional advantages. Implementing PAX on a DBMS that uses NSM requires only changes on the page-level data manipulation code. As a low-overhead solution with a high impact on performance, PAX is particularly attractive to existing large DBMSs. In addition, PAX can be used as an alternative data layout, and the storage manager can decide to use PAX or not when storing a relation, based solely on the number of attributes. PAX is orthogonal to other design decisions, because it only affects the layout of data stored on a single page (e.g., one may first use affinity-based vertical partitioning, and then use PAX for storing the 'thick' subrelations). Finally, research [4] has shown that compression algorithms work better with vertically partitioned relations and on a per-page basis, and PAX has both of these characteristics.

References

[1] A. Ailamaki, D. J. DeWitt, M. D. Hill, and M. Skounakis. Weaving Relations for Cache Performance. Submitted for review to the *27th International Conference on Very Large Data Bases (VLDB),* to be held in Rome, Italy, September 2001.

[2] M. Carey, D. J. DeWitt, M. Franklin, N. Hall, M. McAuliffe, J. Naughton, D. Schuh, M. Solomon, C. Tan, O. Tsatalos, S. White, and M. Zwilling, Shoring Up Persistent Applications. In *proceedings of the ACM SIGMOD Conference on Management of Data,* Minneapolis, MN, May 1994.

[3] G. P. Copeland and S. F. Khoshafian. A Decomposition Storage Model. In *proceedings of the ACM SIGMOD International Conference on Management of Data,* pp. 268-279, May 1985.

[4] J. Goldstein, R. Ramakrishnan, and U. Shaft. Compressing Relations and Indexes. In proceedings of *IEEE International Conference on Data Engineering,* 1998.

*Joint work with David DeWitt and Mark Hill at the University of Wisconsin-Madison, submitted to HPTS 2001.

RECENT PUBLICATIONS

… continued from pg. 7

Selecting the Right Data Distribution Scheme for a Survivable Storage System

Wylie, Bakkaloglu, Pandurangan, Bigrigg, Oguz, Tew, Williams, Ganger, Khosla

Carnegie Mellon University Technical Report CMU-CS-01-120. April, 2001.

Survivable storage system design has become a popular research topic. This paper tackles the difficult problem of reasoning about the engineering trade-offs inherent in data distribution scheme selection. The choice of an encoding algorithm and its parameters positions a system at a particular point in a complex trade-off space among performance, availability, and security. We demonstrate that no choice is right for all systems, and we present an approach to codifying and visualizing this trade-off space. Using this approach, we explore the sensitivity of the space to system characteristics, workload, and desired levels of security and availability.

Authentication Confidences

Ganger

Carnegie Mellon University Technical Report CMU-CS-01-123, May 2001.

"Over the Internet, no one knows you're a dog," goes the joke. Yet, in most systems, a password submitted over the Internet gives one the same access rights as one typed at the physical console. We promote an alternate approach to authentication, in which a system fuses observations about a user into a probability (an authentication confidence) that the user is who they claim to be. Relevant observations include password correctness, physical location, activity patterns, and biometric readings. Authentication confidences refincurrent yes-or-no authentication decisions, allowing systems to cleanly provide partial access rights to authenticated users whose identities are suspect.

Verifiable Secret Redistribution

Wong & Wing

Carnegie Mellon University Technical Report, CMU-CS-01-155, November 2000.

We present a new protocol to perform non-interactive *verifiable secret redistribution* (VSR) for secrets distributed with Shamir's secret sharing scheme. We base our VSR protocol on Desmedt and Jajodia's

… continued on pg. 18

… continued from pg. 17

general redistribution protocol for linear secret sharing schemes, which we specialize for Shamir's scheme. We extend their redistribution protocol with Feldman's non-interactive verifiable secret sharing scheme to ensure that a SUBSHARES-VALID condition is true during redistribution. We show that the SUBSHARES-VALID condition is necessary but not sufficient to guarantee that the new shareholders create valid shares, and present a new SUBSHARES-VALID condition.

Improving Index Performance through Prefetching

Chen, Gibbons & Mowry

2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, California, May 21-24, 2001.

This paper proposes and evaluates Prefetching B^+-Trees (pB^+-Trees), which use prefetching to accelerate two important operations on B^+-Tree indices: searches and range scans. To accelerate searches, pB^+-Trees use prefetching to effectively create wider nodes than the natural data transfer size: e.g., eight vs. one cache lines or disk pages. These wider nodes reduce the height of the B^+-Tree, thereby decreasing the number of expensive misses when going from parent to child without significantly increasing the cost of fetching a given node. Our results show that this technique speeds up search and update times by a factor of 1.2-1.5 for main-memory B^+-Trees. In addition, it outperforms and is complementary to "Cache-Sensitive B^+-Trees." To accelerate range scans, pB^+-Trees provide arrays of pointers to their leaf nodes. These allow the pB^+-Tree to prefetch arbitrarily far ahead, even for nonclustered indices, thereby hiding the normally expensive cache misses associated with traversing the leaves within the range. Our results show that this technique yields over a sixfold speedup on range scans of 1000+ keys. Although our experimental evaluation focuses on main memory databases, the techniques that we propose are also applicable to hiding disk latency.

Better Security via Smarter Devices

Ganger & Nagle

HotOS-VIII (IEEE Workshop on Hot Topics in Operating Systems), May 2001.

This white paper promotes a new approach to network security in which each individual device erects its own security perimeter and defends its own critical resources (e.g., network link or storage media). Together with conventional border defenses, such self-securing devices could provide a flexible infrastructure for dynamic prevention, detection, diagnosis, isolation, and repair of successful breaches in borders and device security perimeters. We overview the self-securing devices approach and the siege warfare analogy that inspired it. We also describe several examples of how different devices might be extended with embedded security functionality and outline some challenges of designing and managing self-securing devices.

My Cache or Yours? Making Storage More Exclusive

Wong, Ganger & Wilkes

Carnegie Mellon University Technical Report CMU-CS-00-157, November 2000.

Modern high-end disk arrays typically have several gigabytes of cache RAM. Unfortunately, most array caches employ management policies in which the same data blocks are cached at both the client and array levels of the cache hierarchy – that is, they are inclusive. As a result, the aggregate cache behaves as if it was only as big as the larger of the client and array caches, instead of as large as the sum of the two.

This paper explores the potential benefits of exclusive caches, in which data blocks are either in the client or array cache, but never in both. Exclusivity helps to create the effect of a single, large unified cache. We propose an operation called DEMOTE for transferring data ejected from the client cache to the array cache, and explore its effectiveness in increasing cache exclusivity using simulation studies. We quantify the benefits of DEMOTE, the overhead it adds, and the effects of combining it with different cache replacement policies across a variety of workloads. The results show that we can obtain useful speedups for both synthetic and real-life workloads.

Analysis of SRPT Scheduling: Investigating Unfairness

Bansal & Harchol-Balter

Proceedings of ACM Sigmetrics 2001 Conference on Measurement and Modeling of Computer Systems, Cambridge, MA, June, 2001.

The Shortest-Remaining-Processing-Time (SRPT) scheduling policy has long been known to be optimal for minimizing mean response time (sojourn time). Despite this fact, SRPT scheduling is rarely used in practice. It is believed that the performance improvements of SRPT over other scheduling policies stem from the fact that SRPT unfairly penalizes the large jobs in order to help the small

… continued on pg. 19

… continued from pg. 18

jobs. This belief has led people to instead adopt "fair" scheduling policies such as Processor-Sharing (PS), which produces the same expected slowdown for jobs of all sizes.

This paper investigates formally the problem of unfairness in SRPT scheduling as compared with PS scheduling. The analysis assumes an M/G/1 model, and emphasizes job size distributions with a heavy-tailed property, as are characteristic of empirical workloads. The analysis shows that the degree of unfairness under SRPT is surprisingly small.

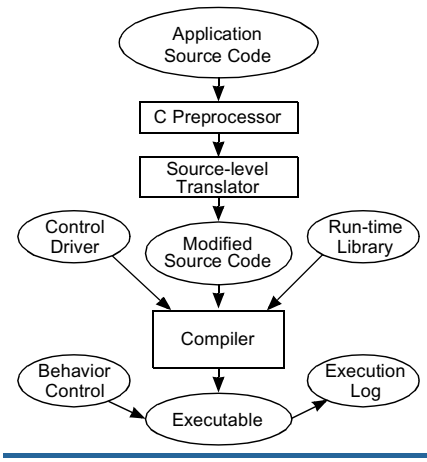
The M/G/1/SRPT and M/G/1/PS queues are also analyzed under overload and closed-form expressions for mean response time as a function of job size are proved in this setting.

Testing the Portability of Desktop Applications to a Networked Embedded System

Bigrigg & Slember

Workshop on Reliable Embedded Systems, with the 20th IEEE Symposium on Reliable Distributed Systems October 28, 2001, New Orleans, LA

Applications that were engineered for desktop environments are often ported to networked embedded systems and mobile environments which have a higher rate of errors due to variable and intermittent connectivity. In embedded systems there is a lack of additional hardware resources, which then requires the software to handle far more and to be increasingly robust. This paper examines the ability of common desktop applications to gracefully handle error conditions when ported to an unreliable networked embedded system. The focus of the testing is the ability of the GNU binutils and textutils to catch and properly handle error return values from the Standard C I/O library.



Test harness phases.

Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture

Chu, Rao, Seshan & Zhang

In response to the serious scalability and deployment concerns with IP Multicast, we and other researchers have advocated an alternate architecture for supporting group communication applications over the Internet where all multicast functionality is pushed to the edge. We refer to such an architecture as End System Multicast. While End System Multicast has several potential advantages, a key concern is the performance penalty associated with such a design. While preliminary simulation results conducted in static environments are promising, they have yet to consider the challenging performance requirements of real world applications in a dynamic and heterogeneous Internet environment. In this paper, we explore how Internet environments and application requirements can influence End System Multicast design. We explore these issues in the context of audio and video conferencing: an important class of applications with stringent performance requirements. We conduct an extensive evaluation study of schemes for constructing overlay networks on a wide-area test-bed of about twenty hosts distributed around the Internet. Our results demonstrate that it is important to adapt to both latency and bandwidth while constructing overlays optimized for conferencing applications. Further, when relatively simple techniques are incorporated into current self-organizing protocols to enable dynamic adaptation to latency and bandwidth, the performance benefits are significant. Our results indicate that End System Multicast is a promising architecture for enabling performance-demanding conferencing applications in a dynamic and heterogeneous Internet environment.

Active Disks for Large-Scale Data Processing

Riedel, Faloutsos, Gibson & Nagle

IEEE Computer, June 2001.

As processor performance increases and memory cost decreases, system intelligence continues to move away from the CPU and into peripherals. Storage system designers use this trend toward excess computing power to perform more complex processing and optimizations inside storage devices. To date, such optimizations take place at relatively low levels of the storage protocol. Trends in storage density, mechanics, and electronics eliminate the hardware bottleneck and put pressure on interconnects and hosts to move data more efficiently.

We propose using an active disk storage device that combines on-drive processing and memory with software downloadability to allow disks to execute application-level functions directly at the device. Moving portions of an application's processing to a storage device signif-

… continued on pg. 20

RECENT PUBLICATIONS

… continued from pg. 19

icantly reduces data traffic and leverages the parallelism already present in large systems, dramatically reducing the execution time for many basic data mining tasks.

The Effects of Wide-Area Conditions on WWW Server Performance

Nahum, Rosu, Seshan & Almeida

WWW workload generators are used to evaluate web server performance, and thus have a large impact on what performance optimizations are applied to servers. However, current benchmarks ignore a crucial component: how these servers perform in the environment in which they are intended to be used, namely the wide area Internet.

This paper shows how WAN conditions can affect WWW server performance. We examine these effects using an experimental testbed which emulates WAN characteristics in a live setting, by introducing factors such as delay and packet loss in a controlled and reproducible fashion. We study how these factors interact with the host TCP implementation and what influence they have on web server performance. We demonstrate that when more realistic wide area conditions are introduced, servers exhibit very different performance properties and scaling behaviors, which are not exposed by existing benchmarks running on LANs. We show that observed throughputs can give misleading information about server performance, and thus find that maximum throughput, or capacity, is a more useful metric. We find that packet losses can reduce server capacity by as much as 50 percent and increase response time as seen by the client. We show that using TCP SACK can reduce client response time, without reducing server capacity.

AWARDS & OTHER PDL NEWS

… continued from pg. 7

March 2001

Garth Goodson Receives IBM Research Fellowship

Congratulations to Garth Goodson, recipient of a Research Fellowship from IBM. The fellowship, eligible for renewal, covers Garth's tuition for the year and includes a stipend of $15,000. Garth plans to spend some time with the storage research group at IBM Almaden in the next year.

Garth is a Ph.D. student in ECE and has recently been working on Self-Securing Storage Systems (S4 for short). Self-securing storage prevents intruders from undetectably tampering with or permanently deleting stored data by internally auditing all requests and keeping all versions of all data for a window of time, regardless of commands received from potentially-compromised host operating systems. Within this window, valuable information exists for intrusion diagnosis and recovery. Garth is also a teaching assistant for 18-546: Introduction to Storage Systems.

February 2001

3 Professors Receive Research Grants from Dept. of Defense**

The U.S. Department of Defense has awarded grants to three Carnegie Mellon faculty members, including Greg Ganger, Assistant Professor of Electrical and Computer Engineering, for their national defense research efforts. The grants were three of 20 awards totaling $9.3 million. The average award is $875,000 per year for three years.

Ganger and co-PI David Nagle have planned a project for the Air Force Office of Sponsored Research focusing on "Enabling Dynamic Security Management of Networked Systems via Device-Embedded Security." Please see the article on page 8 for further information.

* SCS Today
** CMU 8.5 x 11 News
† ECE News

Conference goers at the PDS Workshop & Retreat, November 2000