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

PDL CONSORTIUM MEMBERS

EMC Corporation
Engenio Information Technologies, Inc.
Hewlett-Packard Labs
Hitachi, Ltd.
Hitachi Global Storage Technologies
IBM Corporation
Intel Corporation
Microsoft Corporation
Network Appliance
Oracle Corporation
Panasas, Inc.
Seagate Technology
Sun Microsystems
Veritas Software Corporation

THE
PDL Packet
THE NEWSLETTER ON PDL ACTIVITIES AND EVENTS • FALL 2004
http://www.pdl.cmu.edu/

Ursa Major Starting to Take Shape

Greg Ganger

Ursa Major will be the first full system constructed on PDL's journey towards self-* storage. Ursa Major will be a large-scale object store, in the NASD and OSD style, based on standard server technologies (e.g., rack-mounted servers and gigabit-per-second Ethernet). This article discusses the long-range project goals, some of the progress that has been made on Ursa Major, and near-term plans.

The Self-* Storage Vision

Human administration of storage systems is a large and growing issue in modern IT infrastructures. PDL's Self-* Storage project explores new storage architectures that integrate automated management functions and simplify the human administrative task. Self-* (pronounced "self-star"—a play on the unix shell wild-card character) storage systems should be self-configuring, self-tuning, self-organizing, self-healing, self-managing, etc. Ideally, human administrators should have to do nothing more than provide muscle for component additions, guidance on acceptable risk levels (e.g., reliability goals), and current levels of satisfaction.

We think in terms of systems composed of networked "intelligent" *storage bricks*, which are small-scale servers consisting of CPU(s), RAM, and a number of disks. Although special-purpose hardware may speed up network communication and data encode/decode operations, it is the software functionality and distribution of work that could make such systems easier to administer and competitive in performance and reliability. Designing self-*-ness in from the start allows construction of high-performance, high-reliability storage infrastructures from weaker, less-reliable base units; with a RAID-like argument at a much larger scale, this is the storage analogue of cluster computing's benefits relative to historical supercomputing.

We refer to self-* collections of storage bricks as *storage constellations*. In exploring self-* storage, we plan to develop and deploy a large-scale storage

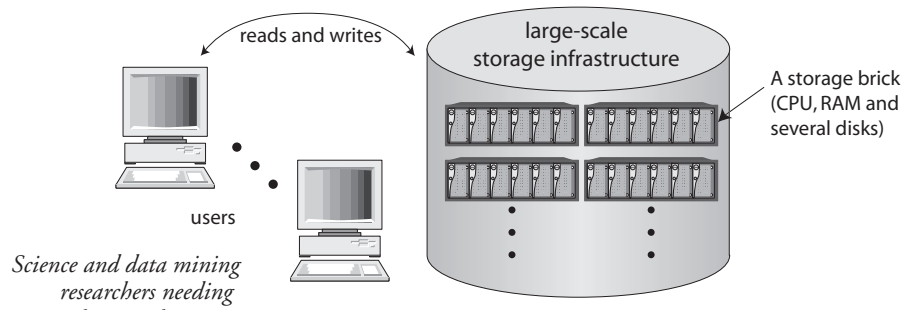


Figure 1. Brick Based infrastructure: a case study for understanding data center challenges and proving ground for new self-management approaches.

continued on page 12

FROM THE DIRECTOR'S CHAIR

Greg Ganger

Hello from fabulous Pittsburgh!

2004 has been a strong year of transition in the Parallel Data Lab, with several Ph.D. students graduating and the Lab's major new project (Self-* Storage) ramping up. Along the way, a faculty member, a new systems scientist, and several new students have joined the Lab, several students spent summers with PDL Consortium companies, and many papers have been published.

For me, the most exciting thing has been the graduations: Garth Goodson, John Linwood Griffin, Jiri Schindler, Steve Schlosser, and Ted Wong. The first four are the first Ph.D. students to complete their degrees with me as their primary advisor, ending long speculation that I never intended to allow any of my students to leave :). All five of these new Doctors are now working with PDL Consortium companies (Garth at Network Appliance, John and Ted at IBM, Jiri at EMC, and Steve at Intel); one or two of them are even expected back at the 2004 PDL Retreat as industry participants.

The PDL continues to pursue a broad array of storage systems research, and this past year brought completion of some recent projects and good progress on the exciting new projects launched last year. Let me highlight a few things.

Of course, first up is the big new project, Self-* Storage, which explores the design and implementation of self-organizing, self-configuring, self-tuning, self-healing, self-managing systems of storage bricks. For years, PDL Retreat attendees pushed us to attack "storage management of large installations," and this project is our response. With generous equipment donations from the PDL Consortium companies, we hope to put together and maintain 100s of terabytes of storage for use by ourselves and other CMU researchers (e.g., in data mining, astronomy, and scientific visualization). IBM, Intel, and Seagate have helped us with seed donations for early development and testing. The University administration is also excited about the project, and is showing their support by giving us some of the most precious resource on campus: space! Two thousand square feet have been allocated in the new Collaborative Innovation Center (CIC) building for a large machine room configured to support today's high-density computing and storage racks. Planning and engineering the room has been an eye-opening experience, inducing new collaborations in areas like dynamic thermal management. Some of the eye-popping specifications include 20KW per rack (average), with a total of two megawatts total, 1300 gallons of chilled water per minute, and 150,000 pounds of weight.

The Self-* Storage research challenge is to make the storage infrastructure as self-* as possible, so as to avoid the traditional costs of storage administration—deploying a real system will allow us to test our ideas in practice. Towards this end, we are designing Ursa Major (the first system) with a clean slate, integrating management functions throughout. In realizing the design, we are combining a number of recent and ongoing PDL projects (e.g., PASIS and self-securing storage) and ramping up efforts on new challenges, such as block-box device and workload modeling, automated decision making, and automated diagnosis.

PDL's Fates database storage project continues to make strides, and has con-

THE PDL PACKET

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

PUBLISHER
Greg Ganger

EDITOR
Joan Digney

The PDL Packet is published once per year and provided to members of the PDL 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.

CONTACT 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
ganger@ece.cmu.edu

Anastassia Ailamaki
natassa@cs.cmu.edu

Anthony Brockwell
abrock@stat.cmu.edu

Chuck Cranor
chuck@ece.cmu.edu

Christos Faloutsos
christos@cs.cmu.edu

Garth Gibson
garth@cs.cmu.edu

Seth Goldstein
seth@cs.cmu.edu

Mor Harchol-Balter
harchol@cs.cmu.edu

Chris Long
chrisl@cs.cmu.edu

Todd Mowry
tcm@cs.cmu.edu

David O'Hallaron
dros@cs.cmu.edu

Adrian Perrig
adrian@ece.cmu.edu

Mike Reiter
reiter@cmu.edu

Mahadev Satyanarayanan
satya@cs.cmu.edu

Srinivasan Seshan
srini@cmu.edu

Dawn Song
dawnsong@ece.cmu.edu

Chenxi Wang
chenxi@ece.cmu.edu

Hui Zhang
hui.zhang@cs.cmu.edu

STAFF MEMBERS

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

Stan Bielski
Mike Bigrigg
John Bucy
Joan Digney
Gregg Economou
Manish Prasad
Raja Sambasivan
Ken Tew
Linda Whipkey
Terrence Wong

GRADUATE STUDENTS

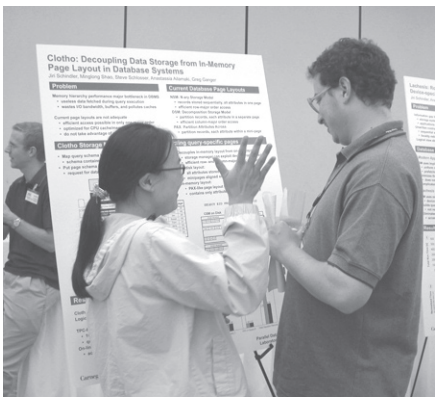
Michael Abd-El-Malek
Mukesh Agrawal
Kinman Au
Shimin Chen
Ivan Dobric
Stavros Harizopoulos
James Hendricks
Andrew Klosterman
Julio López
Chris Lumb
Amit Manjhi
Michael Mesnier
Jim Newsome
Spiros Papadimitriou
Stratos Papadomanolakis
Adam Pennington
Ginger Perng
David Petrou
Brandon Salmon
Minglong Shao
Vlad Shkapenyuk
Shafeeq Sinnamohideen
Craig Soules
John Strunk
Eno Thereska
Niraj Tolia
Tiankai Tu
Mengzhi Wang
Jay Wylie
Shuheng Zhou

FROM THE DIRECTOR'S CHAIR

nected with scientific computing application domains to begin exploring "computational databases." A Fates-based database storage manager transparently exploits select knowledge of the underlying storage infrastructure to automatically achieve robust, tuned performance. As in Greek mythology, there are three Fates: Atropos, Clotho, and Lachesis. The Atropos volume manager stripes data across disks based on track boundaries and exposes aspects of the resulting parallelism. The Lachesis storage manager utilizes track boundary and parallelism information to match database structures and access patterns to the underlying storage. The Clotho dynamic page layout allows retrieval of just the desired table attributes, eliminating unnecessary I/O and wasted main memory. Altogether, the three Fates components simplify database administration, increase performance, and avoid performance fluctuations due to query interference. Computational databases are a new approach to scientific computing in which observation data are loaded into general database systems, and then the scientific queries are optimized automatically by the DBMS rather than by the scientist producing special-purpose code. Fates and other new database systems ideas will be key to the automated optimization component of computational databases.

Other ongoing PDL projects are also producing cool results. For example, the MEMS-based storage work has culminated in an approach for evaluating whether a new storage technology needs a new interface protocol. The self-securing devices project continues to explore intrusion survival features of augmenting devices with security functionality. Among other things, we have extended the storage-based intrusion detection idea so that it can work in commodity disks, despite the lack of direct network attachment or file-based interfaces. The PASIS project has extended the scalable read/write protocols to more general read-modify-write operations. The context-enhanced semantic file systems work has produced some very promising early results. This newsletter and the PDL website offer more details and additional research highlights.

On the education front, in Spring 2004, we again offered our storage systems course to undergraduates and masters students at Carnegie Mellon. Topics span the design, implementation, and use of storage systems, from the characteristics and operation of individual storage devices to the OS, database, and networking techniques involved in tying them together and making them useful. The base lectures were complemented by real-world expertise generously shared by guest speakers from industry. We continue to work on the book, and several other schools have picked up this trend and started teaching similar storage systems courses. Perhaps the most exciting news is that CMU has given me time off from teaching and committee work to focus on completing the book ("Storage Systems") on which this class should be based. We view providing storage systems education as critical to the field's future; stay tuned.



Minglong describes her work on Clotho, an aspect of the Fates Database Storage System, to Vlad.

I'm always overwhelmed by the accomplishments of the PDL students and staff, and it's a pleasure to work with them. As always, their accomplishments point at great things to come.

YEAR IN REVIEW

September 2004

- Natassa gave an invited tutorial on "Database Architectures for New Hardware" at VLDB04 in Toronto, Canada.
- John Griffin successfully defended his Ph.D. dissertation titled "Timing Accurate Storage Emulation: Evaluating Hypothetical Storage Components in Real Computer Systems."
- James Hendricks presented "Secure Bootstrap is Not Enough: Shoring up the Trusted Computing Base" at the 11th SIGOPS European Workshop in Leuven, Belgium.
- Mengzhi Wang presented "Storage Device Performance Prediction with CART Models" at MASCOTS04 in Volendam, The Netherlands. A poster version was presented at ACM SIGMETRICS 2004.
- 12th Annual PDL Retreat and Workshop.

August 2004

- Garth Goodson successfully defended his Ph.D. dissertation titled "Efficient, Scalable Consistency for Highly Fault-tolerant Storage."
- Minglong Shao presented "Clotho: Decoupling Memory Page Layout from Storage Organization" at VLDB04 in Toronto, Canada.
- Christos attended KDD2004 in Seattle, WA where he had two papers being presented: "Fast Discovery of Connection Subgraphs" and "Recovering Latent Time-Series from their Observed Sums: Network Tomography with Particle Filters."

July 2004

- Chris Long presented "Chameleon: Towards Usable RBAC" at the DIMACS Workshop on Usable Privacy and Security Software at Rutgers Univ., NJ.

June 2004

- David Petrou presented "Cluster Scheduling for Explicitly-Speculative Tasks" at the Internat'l Conference on Supercomputing (ICS) 2004 in St.-Malo, France.
- Jay Wylie presented "A Protocol Family Approach to Survivable Storage Infrastructures" at FuDiCo II in Bertinoro, Italy.
- Chenxi Wang spoke on "Dynamic Quarantine of Internet Worms" at DSN 04 in Florence, Italy.Jay Wylie also spoke at this conference, presenting "Efficient Byzantine-tolerant Erasure-coded Storage."
- Christos gave a tutorial on "Indexing and Mining Streams" at SIGMOD 2004 in Paris, France.

May 2004

- 6th annual PDL Industry Visit Day.
- Jiri Schindler and Steve Schlosser, Greg's first Ph.D. students to finish, were awarded their degrees in ECE. Ted Wong received his Ph.D. from SCS.
- Mike Mesnier presented "File Classification in Self-* Storage Systems" at ICAC-04 in New York. He also helped lead a storage workshop on intelligent storage devices and worked with others in the industry to put together a research roadmap for the next 5-10 years.
- Chenxi Wang presented "Providing Content-based Services on top of Peer-to-peer Systems" at DEBS'04 (Distributed Event-Based Systems), in Edinburgh, Scotland.
- Mengzhi Wang successfully proposed her Ph.D research titled "Black-Box Storage Device Models with Learning."
- James Newsome and Amit Manjhi spent the summer interning at Intel's Pittsburgh office, and Eno Thereska interned with Microsoft Research in Cambridge, UK. Niraj Tolia was also in Cambridge, UK, at the Intel Research Lab.

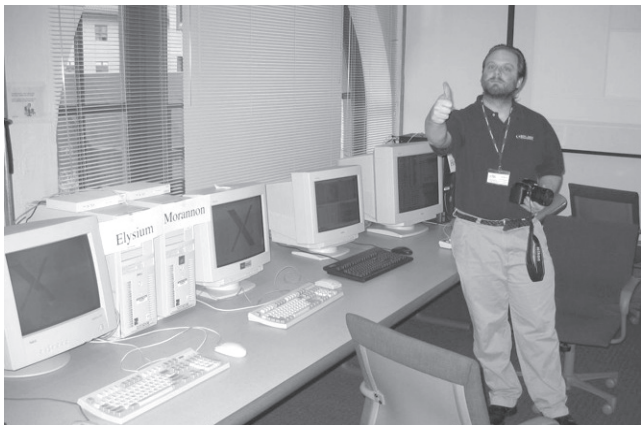
April 2004

- Steve Schlosser successfully defended his Ph.D. dissertation, "Using MEMS-based Storage Devices in Computer Systems."
- 10 PDL faculty and students attended FAST 04 in San Francisco, CA.
- Eno Thereska presented "A Framework for Building Unobtrusive Disk Maintenance Applications" at FAST 04. He and his co-authors Jiri Schindler, John Bucy, Brandon Salmon, Christopher R. Lumb, Gregory R. Ganger were awarded Best Student Paper.
- Also presenting at FAST 04 were Jiri Schindler ("Atropos: A Disk Array Volume Manager for Orchestrated Use of Disks") and Steve Schlosser ("MEMS-based Storage Devices and Standard Disk Interfaces: A Square Peg in a Round Hole?")
- Over the past year, several industry visitors have contributed to our Storage Systems course including David Black, EMC; Erik Riedel, Seagate; and Mike Kazar, Network Appliance.

March 2004

- SDI Speaker: Larry Huston,

continued on page 19



Adam gives the go-ahead for a PDL Industry Visit Day demo.

D-SPTF: Decentralized Request Distribution in Brick-based Storage Systems

Lumb, Golding & Ganger

Proceedings of ASPLOS'04, Boston, Massachusetts, October 7–13 ,2004.

Distributed Shortest-Positioning Time First (D-SPTF) is a request distribution protocol for decentralized systems of storage servers. D-SPTF exploits high-speed interconnects to dynamically select which server, among those with a replica, should service each read request. In doing so, it simultaneously balances load, exploits the aggregate cache capacity, and reduces positioning times for cache misses. For network latencies expected in storage clusters (e.g., 10-200µs), D-SPTF performs as well as would a hypothetical centralized system with the same collection of CPU, cache, and disk resources. Compared to popular decentralized approaches, D-SPTF achieves up to 65% higher throughput and adapts more cleanly to heterogeneous server capabilities.

Cluster Scheduling for Explicitly Speculative Tasks

Petrou, Ganger & Gibson

Proceedings of the 18th Annual ACM International Conference on Supercomputing (ICS'04), Malo, France, June 26–July 1, 2004.

Large-scale computing often consists of many speculative tasks to test hypotheses, search for insights, and review potentially finished products. E.g., speculative tasks are issued by bioinformaticists comparing DNA sequences and computer graphics artists adjusting scene properties. We promote a way of working that exploits the inherent speculation in application-level search made more common by the cost-effectiveness of grid and cluster computing. Researchers and end-users disclose sets of speculative tasks that search an application space, request specific results as needed, and cancel unfinished tasks if early results suggest no need to continue. Doing so matches natural usage patterns, making users more effective, and also enables a new class of schedulers.

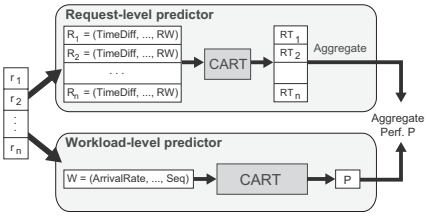
In simulation, we show how batch-active schedulers reduce user-observed response times relative to conventional models in which tasks are requested one at a time (interactively) or in batches without specifying which tasks are speculative. Over a range of situations, user-observed response time is about 50% better on average and at least two times better for 20% of our simulations. Moreover, we show how user costs can be reduced under an incentive cost model of charging only for tasks whose results are requested.

Storage Device Performance Prediction with CART Models

Wang, Au, Ailamaki, Brockwell, Faloutsos & Ganger

Proceedings of the 12th Annual Meeting of the IEEE / ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS Volendam, The Netherlands. October 5–7, 2004.

Storage device performance prediction is a key element of self-managed storage systems and application planning tasks, such as data assignment. This work explores the application of a machine learning tool, CART models, to storage device modeling. Our approach predicts a device's performance as a function of input workloads, requiring no knowledge of the device internals. We propose two uses of CART models: one that predicts per-request response times (and then derives aggregate values) and one that predicts aggregate values directly from workload characteristics. After being trained on the device in question, both provide accurate black-box models across a range of test traces from real environments. Experiments show that these models predict the



Per-request and feature-based predictors. r_i is the i-th request in the workload; RT_i is the response time of r_i; R_i is the set of per-request characteristics for r_i; W is the set of workload-level characteristics.

average and 90th percentile response time with an relative error as low as 19% when the training workloads are similar to the testing workloads and a good interpolation across different workloads.

Toward Automatic Context-based Attribute Assignment for Semantic File Systems

Soules & Ganger

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-04-105. June 2004.

Semantic file systems enable users to search for files based on attributes rather than just pre-assigned names. This paper develops and evaluates several new approaches to automatically generating file attributes based on context, complementing existing approaches based on content analysis. Context captures broader system state that can be used to provide new attributes for files, and to propagate attributes among related files; context is also how humans often remember previous items [2], and so should fit the primary role of semantic file systems well. Based on our study of ten systems over four months, the addition of context-based mechanisms, on average, reduces the number of files with zero attributes by 73%. This increases the total number of classifiable files by over 25% in most cases, as is shown in Figure 1. Also, on average, 71% of the content-analyzable files also gain additional valuable attributes.

continued on page 6

continued from page 5

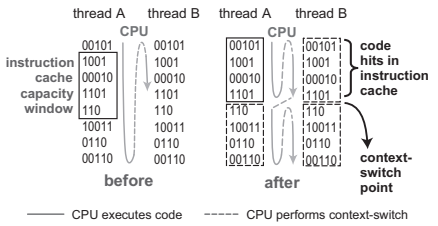
STEPS Towards Cache-Resident Transaction Processing

Harizopoulos & Ailamaki

Proceedings of the 30th VLDB Conference, Toronto, Canada, 29 August –3 September 2004.

Online transaction processing (OLTP) is a multibillion dollar industry with high-end database servers employing state-of-the-art processors to maximize performance. Unfortunately, recent studies show that CPUs are far from realizing their maximum intended throughput because of delays in the processor caches. When running OLTP, instruction-related delays in the memory subsystem account for 25 to 40% of the total execution time. In contrast to data, instruction misses cannot be overlapped with out-of-order execution, and instruction caches cannot grow as the slower access time directly affects the processor speed. The challenge is to alleviate the instruction-related delays without increasing the cache size.

We propose Steps, a technique that minimizes instruction cache misses in OLTP workloads by multiplexing concurrent transactions and exploiting common code paths. One transaction paves the cache with instructions, while close followers enjoy a nearly miss-free execution. Steps yields up to 96.7% reduction in instruction cache misses for each additional concurrent transaction, and at the same time eliminates up to 64% of mispredicted branches by loading a repeating execution pattern into the CPU. This paper (a) describes the design and implementation of Steps, (b) analyzes Steps using microbenchmarks, and (c) shows Steps performance when running TPC-C on top of the Shore storage manager.



Before: As the instruction cache cannot fit the entire code, when the CPU switches (dotted line) to thread B it will incur the same number of misses. After: If we "break" the code into several pieces that fit in the cache, and switch execution back and forth between the two threads, thread B will find all instructions in the cache.

Clotho: Decoupling Page Layout from Storage Organization

Shao, Schindler, Schlosser & Ailamaki & Ganger

Proceedings of the 30th VLDB Conference. Toronto, Canada, 29 August –3 September 2004.

As database application performance depends on the utilization of the disk and memory hierarchy, and the speed gap between the processor and memory components widens, smart data placement plays a central role in increasing locality and in improving memory utilization. Existing techniques, however, do not optimize accesses to all levels of memory hierarchy and for all the different workloads, because each storage level uses different technology (cache, memory, disks) and each application accesses data using different (often conflicting) patterns. This paper introduces Clotho, a new buffer pool and storage management architecture. Clotho decouples in-memory page layout from data organization on non-volatile storage devices, enabling independent data layout design at each level of the storage hierarchy. Using Clotho, a DBMS can maximize cache and memory utilization by (a) transparently using appropriate data layouts on memory and non-volatile storage, and (b) dynamically synthesizing data pages to follow application access patterns at each level as needed. Clotho enables (a) independently-tailored page layouts for dynamically changing as well as compound workloads, and (b) use of alternative technologies at each level (e.g., disk arrays or MEMS-based storage devices). We describe the Clotho design and implementation using disk array logical volumes and simulated MEMS-based storage devices, and we evaluate performance under a variety of workloads.

AutoPart: Automating Schema Design for Large Scientific Databases Using Data Partitioning

Papadomanolakis & Ailamaki

16th International Conference on Scientific and Statistical Database Management (SSDBM). Santorini Island, Greece. June 21–23, 2004.

Database applications that use multi-terabyte datasets are becoming increasingly important for scientific fields such as astronomy and biology. Scientific databases are particularly suited for the application of automated physical design techniques, because of their data volume and the complexity of the scientific workloads. Current automated physical design tools focus on the selection of indexes and materialized views. In large-scale scientific databases, however, the data volume and the continuous insertion of new data allows for only limited indexes and materialized views. By contrast, data partitioning does not replicate data, thereby reducing space requirements and minimizing update overhead. In this paper we present AutoPart, an algorithm that automatically partitions database tables to optimize sequential access assuming prior knowledge of a representative workload. The resulting schema is indexed using a fraction of the space required for indexing the original schema. To evaluate AutoPart we built an automated schema design tool that interfaces to commercial

continued on page 7

continued from page 6

database systems. We experiment with AutoPart in the context of the Sloan Digital Sky Survey database, a real-world astronomical database, running on SQL Server 2000. Our experiments demonstrate the benefits of partitioning for large-scale systems: Partitioning alone improves query execution performance by a factor of two on average. Combined with indexes, the new schema also outperforms the indexed original schema by 20% (for queries) and a factor of five (for updates), while using only half the original index space.

Dynamic Quarantine of Internet Worms

Wong, Wang, Song, Bielski & Ganger

Proceedings of the International Conference on Dependable Systems and Networks (DSN-2004). Palazzo dei Congressi, Florence, Italy. June 28th –July 1, 2004.

If we limit the contact rate of worm traffic, can we alleviate and ultimately contain Internet worms? This paper sets out to answer this question. Specifically, we are interested in analyzing different deployment strategies of rate control mechanisms and the effect thereof on suppressing the spread of worm code. We use both analytical models and simulation experiments. We find that rate control at individual hosts or edge routers yields a slowdown that is linear in the number of hosts (or routers) with the rate limiting filters. Limiting contact rate at the backbone routers, however, is substantially more effective -- it renders a slowdown comparable to deploying rate limiting filters at every individual host that is covered. This result holds true even when susceptible and infected hosts are patched and immunized dynamically. To provide context for our analysis, we examine real traffic traces obtained from a campus computing network. We observe that rate throttling could be enforced with minimal impact on legitimate communications. Two worms observed in the traces, however, would be significantly slowed down.

The Safety and Liveness Properties of a Protocol Family for Versatile Survivable Storage Infrastructures

Goodson, Wylie, Ganger & Reiter

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-03-105. March 2004.

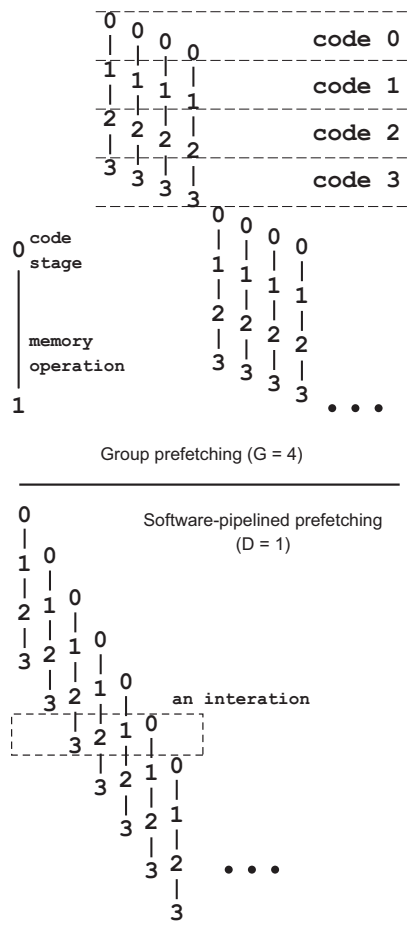
Survivable storage systems mask faults. A protocol family shifts the decision of which types of faults from implementation time to data-item creation time. If desired, each data-item can be protected from different types and numbers of faults with changes only to client-side logic. This paper presents proofs of the safety and liveness properties for a family of storage access protocols that exploit data versioning to efficiently provide consistency for erasure-coded data. Members of the protocol family may assume either a synchronous or asynchronous model, can tolerate hybrid crash-recovery and Byzantine failures of storage-nodes, may tolerate either crash or Byzantine clients, and may or may not allow clients to perform repair. Additional protocol family members for synchronous systems under omission and fail-stop failure models of storage-nodes are developed.

Improving Hash Join Performance through Prefetching

Chen, Ailamaki, Gibbons & Mowry

Proceedings of the 20th International Conference on Data Engineering (ICDE 2004). Boston, MA. March 30 –April 2, 2004.

Hash join algorithms suffer from extensive CPU cache stalls. This paper shows that the standard hash join algorithm for disk-oriented databases (i.e. GRACE) spends over 73% of its user time stalled on CPU cache misses, and explores the use of prefetching to improve its cache performance. Applying prefetching to hash joins is complicated by the data dependencies, multiple code paths, and inherent randomness of hashing. We present two techniques, group prefetching and software-pipelined prefetching, that overcome these complications. These schemes achieve 2.0–2.9X speedups for the join phase and 1.4–2.6X speedups for the partition phase over GRACE and simple prefetching approaches. Compared with previous cache-aware approaches (i.e. cache partitioning), the schemes are at least 50% faster on large relations and do not require exclusive use of the CPU cache to be effective.



Intuitive pictures of the prefetching schemes.

continued on page 16

AWARDS & OTHER PDL NEWS

September 2004

Sanjay Seshan Welcomes a New Brother!

Asha, Sanjay and Srini Seshan are proud to announce the arrival of a new baby boy to their family. Arvind Seshan arrived at 6:16 p.m. on September 9th, weighing 6 lbs. 14 oz. and measuring 21 inches. Our congratulations to the family!

September 2004

Garth Goodson, John Griffin join PDL Consortium Companies

Garth Goodson successfully defended his thesis on "Efficient, Scalable Consistency for Highly Fault-tolerant Storage" on August 28, and has a start date of September 13 for his new job at Network Appliance in Sunnyvale, CA. John Griffin also successfully defended his research on "Timing Accurate Storage Emulation: Evaluating Hypothetical Storage Components in Real Computer Systems" on September 1. He is set to move to the New York City area to begin work at IBM's T.J Watson Research Center in mid-September. Dissertation abstracts begin on page 12.

June 2004

Srinivasan Seshan awarded Finmeccanica Chair

Srinivasan Seshan, associate professor in the CS Department, has been awarded the school's Finmeccanica Chair. Endowed in 1989, the Italian Finmeccanica Fellowship "acknowledges promising young faculty members in the field of computer science," and is designed as a three year appointment.

The Finmeccanica Group ranks among the largest international firms in its operating sectors of aerospace, defense, energy, transportation and information technology. They are active in the design and manufacture of aircraft, satellites, power generation components, information and technology services—to name a few—and the realization of these systems via engineering, electronics, information technology and innovative materials.

—with info from CMU 8.5x11 News, Jun 3, 2004.

May 2004

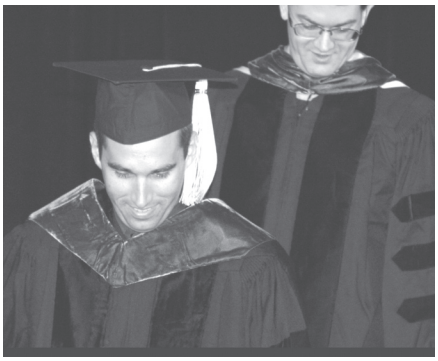
PDL Student Graduates: Congratulations Ted!

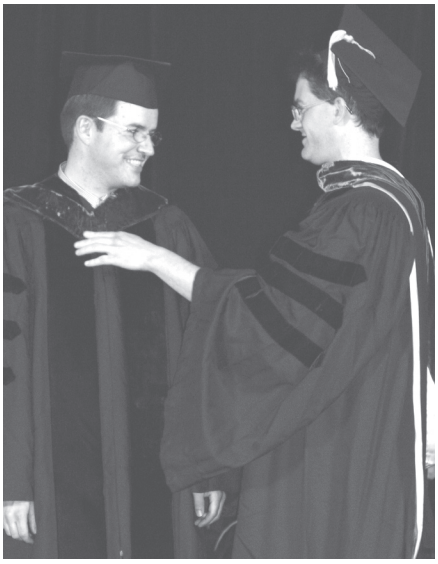
Ted Wong convocated with a Ph.D. from the School of Computer Science on May 16. His thesis work was on "Decentralized Recovery for Survivable Storage Systems" (abstract is available on page 13). Ted has been with IBM Research since December 2003, working on self-managing brick-based storage systems. In the last twelve months, Ted has also gotten married, moved across the US to start his new job at Almaden, and bought a house.

May 2004

Greg's First Ph.D. Students to Graduate: Congrats to Jiri and Steve!

Greg's first students to complete their Ph.D.s, Steve Schlosser and Jiri Schindler, were awarded their degrees in Electrical and Computer Engineering this spring. The hooding ceremony took place on May 15 and they attended the University's convocation ceremonies on May 16. Jiri has been working at EMC in Boston since last fall, and Steve has taken a position with Intel in Pittsburgh. Abstracts of their dissertations are available starting on page 12.





May 2004

Best Student Paper Award at PAKDD '04

One of Christos Faloutsos' students, Jia-Yu (Tim) Pan, won the 'best student paper' award in PAKDD 2004 for the paper "AutoSplit: Fast and Scalable Discovery of Hidden Variables in Stream and Multimedia Databases," co-authored with Christos Faloutsos, Masafumi Hamamoto and Hiroyuki Kitagawa. The paper proposed a new, generic method for pattern detection;

continued on page 9

continued from page 8

experiments on motion capture data, images and financial time series, showed that it consistently outperforms the traditional SVD method. Congratulations, Tim!

April 2004

PDL Researchers Receive Best Student Paper Award at FAST '04

Congratulations to PDL researchers Eno Thereska, Jiri Schindler, John Bucy, Brandon Salmon and Gregory R. Ganger, who have been awarded Best Student Paper by the program committee of the USENIX Conference on File and Storage technologies (FAST '04) for their paper "A Framework for Building Unobtrusive Disk Maintenance Applications."

The FAST Best Student Paper award was also given to PDL researchers in 2002 when Jiri Schindler, John Linwood Griffin, Christopher R. Lumb, and Gregory R. Ganger were recognized for their paper "Track-Aligned Extents: Matching Access Patterns to Disk Drive Characteristics."

April 2004

Best Paper Award at ICDE 2004

Shimin Chen, Anastassia Ailamaki, Phillip Gibbons, and Todd Mowry received the best paper award for "Improving Hash Join Performance through Prefetching" at the International Conference on Data Engineering (ICDE) 2004. The conference took place in Boston, MA from March 30 through April 2. ICDE is one of the top database conferences, with hundreds of submitted papers, and extremely selective acceptance ratio (typically, 1-of-5 to 1-of-7). The paper focuses on the most expensive database operation ('join'), and proposes novel methods to accelerate it.

April 2004

HGST Joins the PDL Industrial Research Consortium

The PDL would like to welcome Hitachi Global Storage Technologies (HGST) to the group of Consortium member companies who support our research. From www.hitachigst.com: Hitachi Global Storage Technologies was founded in 2003 and was formed as a result of the strategic combination of Hitachi and IBM's storage technology businesses and have their head office in San Jose, CA. Other major development and manufacturing locations are found worldwide - in Japan, Thailand, Singapore and Mexico. The company's vision is to enable users to fully engage in the digital lifestyle by providing access to large amounts of storage capacity in formats suitable for the office, on the road and in the home.

March 2004

CMU Sensor Detects Computer Hard Drive Failures

The Critter Temperature Sensor, a new heat-sensitive sensor to detect computer hard drive failures, which attaches to a user's desktop computer, is being deployed across campus to monitor the working environment of university computers, according to Michael Bigrigg, a project scientist for the PDL.

"We are trying save the life of the computer hard drive. Hard drives get hot and the sensor is designed to pick up the slightest temperature variation." Bigrigg added that the sensor will help researchers understand wasted energy with the hope of extending the lifespan of a computer hard drive by sensing how much daily heat a hard drive endures. So far, the new sensor, the size of a dime, has been deployed in offices and labs throughout Carnegie Mellon's Hamburg Hall.

–with info from ece news & events, Mar. 1, 2004

February 2004

Network Appliance Donates Filer for PDL Storage Needs

Network Appliance has donated a FAS900 series filer with 2 Terabytes of raw capacity and all of the software bells and whistles, with a retail value of $170K, in all. This filer will be used for critical PDL storage needs, including a software development repository, the PDL web server, and Lab member home directories.

February 2004

Mowry to Head Intel Lab

Todd Mowry, associate professor of Computer Science, will succeed Mahadev Satyanarayanan as head of Intel Research Pittsburgh, effective this May. Mowry will bring a new research thrust to the lablet at the intersection of databases, architecture, compilers and operating systems. According to Satyanarayanan, in the two short years of its existence, Intel Research Pittsburgh is already making a big impact on a number of areas of research, including personal computing mobility (Internet Suspend/ Resume project), wide-area sensing (IrisNet project), and interactive search of complex data (Diamond project). "We are clearly past the startup phase, and can look forward to continued growth and many more accomplishments in 2004 and beyond," Satyanarayanan said.

–cmu.misc.news, Feb. 3, 2004

January 2004

Spiros Papadimitriou wins a Best Paper Award at VLDB03

Computer Science Ph.D. candidate Spiros Papadimitriou has received a Best Paper Award from the Very Large Data Bases (VLDB) 2003 Conference for his paper "Adaptive, Hands-Off Stream Mining," which was co-au-

continued on page 11

continued from page 1

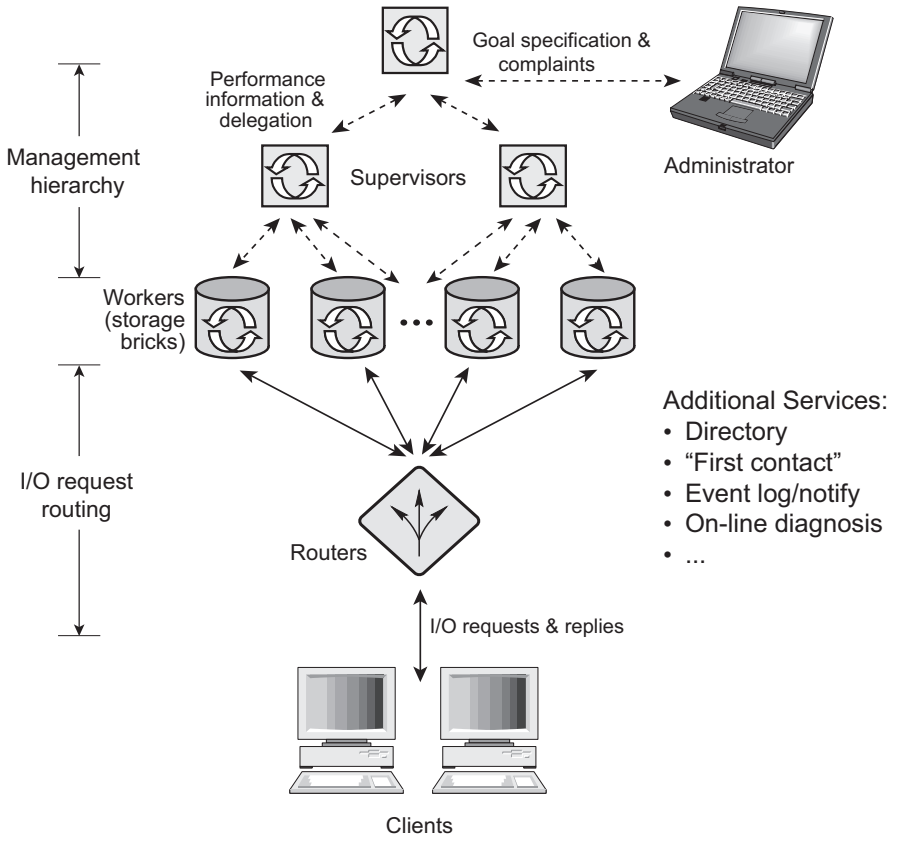


Figure 2. Self-* Storage Architecture

constellation, with capacity provided to research groups (e.g., in data mining and scientific visualization) around Carnegie Mellon who rely on large quantities of storage for their work. We are convinced that such deployment and maintenance is necessary to evaluate the ability of new architectures/technologies to simplify administration for the system scales and workload mixes that traditionally present difficulties. As well, our use of low-cost hardware and immature software will push the frontiers of fault-tolerance and automated recovery mechanisms, which are critical for storage infrastructures.

Our self-* storage white paper [Ganger 03] categorizes storage administration tasks and overviews the self-* storage project in more detail.

Early Foci: Fault-tolerance, Versatility & Instrumentation

The initial stages of the self-* storage project have consisted of large amounts of design and infrastructure building. The foci during this process has been on enabling high levels of fault-tolerance, versatility, and instrumentation. Together, these can provide a foundation for the automated decision making, feedback control, diagnosis, and repair mechanisms needed to achieve self-*-ness.

Fault tolerance is achieved by spreading data redundantly across a set of workers (i.e., storage bricks) so as to ensure availability despite failures. PDL's PASIS project has developed novel data access protocols that provide efficient fault tolerance by exploiting local data versioning within each storage-node (as provided, in our case, by PDL's S4 object store). The protocols gain efficiency from several features. First, they allow use of m-of-n erasure codes in addition to standard replication, which can substantially reduce communication overheads. Second, most read operations complete in just a single round-trip; only concurrent sharing or failures require additional rounds. Third, most protocol processing is shifted to clients, allowing greater scalability. Fourth, the protocols do not require digital signatures; cryptographic hashes are instead used to tolerate Byzantine failures. These features, combined, will allow Ursa Major to protect important data from multiple failures without excessive overhead. Versatility is achieved by allowing each data object—actually, each range of data in each object—to be encoded differently, tolerating different numbers and types of failures, and to be stored on different workers. The data access protocol family, by design, allows the fault model and encoding scheme to be selected on a per-data-item basis. Additional versatility is achieved by allowing the encoding and/or distribution to be changed dynamically.

Ursa Major's versatility helps most clearly with tuning, offering fine-grained control over key performance and reliability determiners of distributed storage systems. Rather than forcing a single pre-configured data distribution policy to be applied for all storage, or even each entire dataset, Ursa Major allows this policy choice to be specialized for each object and to be changed dynamically. This versatility eliminates three key problems. First, finding a single encoding choice (e.g., replication vs. erasure codes) that is right for all files requires compromising on performance and estimating which types of files are likely to be most popular. Second, choosing a single degree of fault tolerance for all data ignores the differing reliability goals of different datasets, making all data pay the capacity and performance costs associated with the maximum fault tolerance desired. Third, disallowing dynamic change prevents observation-based setting of the policies. Ursa Major's versatility

continued on page 11

continued from page 9

thored with Anthony Brockwell and Christos Faloutsos. The conference took place this past September in Berlin and is one of the most prestigious and selective database conferences.

– with info from CMU 8.5x11 News, Jan. 8, 2004

January 2004

Seagate supports the Self-* Storage Project with Equipment Donation

Greg Ganger (Assoc. Professor, ECE and CS) and the Parallel Data Lab (PDL) have received a $25K equipment grant of high-end SCSI disks from Seagate. The grant significantly increases the capacity of the testbed for PDL's new Self-* Storage project, which seeks to create large-scale self-managing, self-organizing, self-tuning storage systems from generic servers.

January 2004

LSI Logic (now Engenio) Joins PDL Research Consortium

The PDL is pleased to announce that LSI Logic has joined the PDL Consortium of companies that support and participate in PDL research. From the LSI Logic page: Founded in 1981, LSI Logic pioneered the ASIC (Application Specific Integrated Circuit) industry. LSI Logic is a leading designer and manufacturer of communications, consumer and storage semiconductors for applications that access, interconnect and store data, voice and video. Today, the company focuses on providing highly complex ASICs, ASSPs (Application Specific Standard Products), RapidChip™, host bus adapters, software and storage systems.

November 2003

Steve and Rachel Married!

Congratulations to Steve Schlosser and Rachel Fielder, who were married on November 8, 2003, in Pittsburgh at the Schenley Park Visitor Center. Steve recently graduated from CMU with his Ph.D. in Electrical and Computer Engineering and is now at Intel Research in Pittsburgh.

URSA MAJOR

continued from page 10

allows reasonable default choices to be used initially and then modified, object by object, to match each object's observed workload and any load imbalances in the system.

Instrumentation pervades Ursa Major's design, providing information about the system required for it to diagnose, repair, and even predict problems. Two primary forms of information are collected: events and traces. An "event" is an observation made by some component of the system that it believes may be of interest to another component or to a diagnosis agent. Examples include a worker reporting corrupted blocks, a metadata server reporting an attempted quota violation, and a client reporting an unresponsive worker. The event service collects these events, which are reported to it, and allows components of the system to query for events of interest. (The event service is thus a publish-subscribe database system.) For example, system-healing services may subscribe for events indicating failures or lost data and take corrective action. Diagnosis services may mine substantial portions of the event history to identify root causes of recurring problems.

A "trace" is a recorded sequence of requests and response timings. Many components will retain such traces of their activity and responsiveness, including workers, metadata servers, and clients wishing to report performance problems. Traces allow post-hoc analysis of workloads and efficiency and also allow the system to explore "what if?" questions about addition of datasets and workload redistribution. Cross-component activity tracking is used to correlate requests observed at different system components, so that performance debugging services can deduce which component(s) are responsible for performance problems observed by clients.

Setting Up the Physical Infrastructure

A large-scale computing infrastructure project involves logistics challenges beyond building and maintaining software. To deploy Ursa Major at interesting scales, we need large amounts of equipment and properly-conditioned machine room(s) in which to operate it. For today's high-density computing equipment, machine room design involves power, cooling, and structural challenges that were new to us and Carnegie Mellon. And, of course, acquiring the equipment and funds for renovations are always challenging.

Carnegie Mellon's administration is excited about this project, and has demonstrated this excitement in the most tangible ways: space allocations and renovations. Over the last year, PDL's existing machine room has

continued on page 16

DISSERTATION ABSTRACTS

PH.D. DISSERTATION

Timing-Accurate Storage Emulation: Evaluating Hypothetical Storage Components in Real Computer Systems

John Linwood Griffin, ECE
September 1, 2004

Timing-accurate storage emulation offers the opportunity to investigate novel uses of storage in computer systems, permitting forays into the space of hypothetical device functionalities without the difficulties of developing and supporting extensively nonstandard or novel interface actions in prototype or production systems. This dissertation demonstrates that there is a current and pressing need for a new storage evaluation technique, and that it is feasible to design and construct a timing-accurate storage emulator and to use an emulator for interesting systems-level experimentation.

Timing-accurate storage emulation offers a unique performance evaluation capability: the flexibility of simulation and the reality of experimental measurements. This 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 (or mathematical model) of that component. This allows simulated storage components to be plugged into real systems, which can then be used for complete, application-based experiments. To accomplish this, the emulator must synchronize the simulator's internal time with the real-world clock, inserting requests into the simulator when they arrive and reporting completions when the simulator determines they are done. If the simulator's model represents a real component, the system-observed performance will be of that component. Thus, the results from application benchmarking will represent the end-to-end performance effect of using that component in a real system.

We built a functional timing-accurate storage emulator and demonstrated its use in experiments involving currently-existing storage products, experiments evaluating the potential of nonexistent storage components, and experiments evaluating interactions between modified external system architectures and modified or hypothetical storage device functionality. To explore standalone hypothetical storage components in computer systems, we configured our emulator with three device models representing a currently-available production disk drive, a hypothetical 50,000 RPM disk drive, and a hypothetical MEMS-based storage device, and executed three distinct application-level workloads against these three emulator configurations. To explore new system architectures with expanded device functionality, we applied the principles of timing-accurate storage emulation in an investigation into storage-based intrusion detection systems. This experimentation demonstrates the feasibility of including intrusion detection capabilities into a standalone processing-enhanced disk drive, and also demonstrates how existing communications paths may be used by an operating system to transmit and receive information regarding the configuration and operational status of such an intrusion detection-enhanced device.

PH.D. DISSERTATION

Efficient, Scalable Consistency for Highly Fault-tolerant Storage

Garth Goodson, ECE
August 28, 2004

Fault-tolerant storage systems spread data redundantly across a set of storage-nodes in an effort to preserve and provide access to data despite failures. One difficulty created by this architecture is the need for a consistent view, across storage-nodes, of the most recent update. Such consistency is made difficult by concurrent updates, partial updates made by clients that fail, and failures of storage-nodes.

This thesis demonstrates a novel approach to achieving scalable, highly fault-tolerant storage systems by leveraging a set of efficient and scalable, strong consistency protocols enabled by storage-node versioning. Versions maintained by storage-nodes can be used to provide consistency, without the need for central serialization, and despite concurrency. Since versions are maintained for every update, even if a client fails part way through an update, concurrency exists during an update, the latest complete version of the data-item being accessed still exists in the system—it does not get destroyed by subsequent updates. Additionally, versioning enables the use of optimistic protocols.

This thesis develops a set of consistency protocols appropriate for constructing blockbased storage and metadata services. The block-based

continued on page 13

John looks to his future.

continued from page 12

storage protocol is made space efficient through the use of erasure codes and made scalable by off-loading work from the storage-nodes to the clients. The metadata service is made scalable by avoiding the high costs associated with agreement algorithms and by utilizing threshold voting quorums. Fault-tolerance is achieved by developing each protocol in a hybrid storage-node fault model (a mix of Byzantine and crash storage-nodes can be tolerated), capable of tolerating crash or Byzantine clients, and utilizing asynchronous communication.

PH.D. DISSERTATION

Using MEMS-based Storage Devices in Computer Systems

Steven W. Schlosser, ECE
May 2004

MEMS-based storage is an interesting new technology that promises to bring fast, non-volatile, mass data storage to computer systems. MEMS-based storage devices (MEMStores) themselves consist of several thousand read/write tips, analogous to the read/write heads of a disk drive, which read and write data in a recording medium. This medium is coated on a moving rectangular surface that is positioned by a set of MEMS actuators. Access times are expected to be less than a millisecond with energy consumption 10-100X less than a low-power disk drive, while streaming bandwidth and volumetric density are expected to be around that of disk drives.

This dissertation explores the use of MEMStores in computer systems, with a focus on whether systems can use existing abstractions and interfaces to incorporate MEMStores effectively, or if they will have to change the way they access storage to benefit from MEMStores. If systems can use MEMStores in the same way that they use disk drives, it will be more likely that MEMStores will be adopted when they do become available.

Since real MEMStores do not yet exist, I present a detailed software model that allows their use to be explored under a variety of workloads. To answer the question of whether a new type of device requires changes to systems, I present a methodology that includes two objective tests for determining whether the benefit from a device is due to a specific difference in how that device accesses data or is just due to the fact that the device is faster, smaller, or uses less energy than current devices. I present a range of potential uses of MEMStores in computer systems, examining each under a number of user workloads, using the two objective tests to evaluate their efficacy.

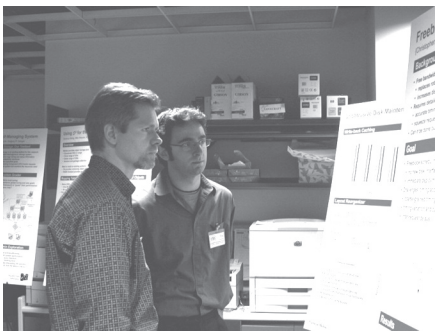
Using the evidence presented and the two objective tests, I show that systems can incorporate MEMStores easily and employ the same standard abstractions and interfaces used with disk systems. At a high level, the intuition is that MEMStores are mechanical storage devices, just like disk drives, only faster, smaller, and requiring less energy to operate. Accessing data requires an initial seek time that is distance-dependent, and, once access has begun, sequential access is the most efficient. This intuition is described in more detail, and the result is shown to hold for the range of uses presented.

PH.D. DISSERTATION

Decentralized Recovery for Survivable Storage Systems

Theodore Ming-Tao Wong, SCS
December 1, 2003

Modern society has produced a wealth of data to preserve for the long term. Some data we keep for cultural benefit, in order to make it available to future generations, while other data we keep because of legal imperatives. One way to preserve such data is to store it using *survivable* storage systems. Survivable storage is distinct from reliable storage in that it tolerates confidentiality failures in which unauthorized users compromise component storage servers, as well as crash failures of servers. Thus, a survivable storage system can guarantee both the availability and the confidentiality of stored data.



Eno discusses Freeblock Scheduling with Robin Huber of Engenio at the PDL Spring Visit Day.

Research into survivable storage systems investigates the use of *m*-of-*n* threshold sharing schemes to distribute data to servers, in which each server receives a share of the data. Any *m* shares can be used to reconstruct the data, but any *m - 1* shares reveal no information about the data. The central thesis of this dissertation is that to truly preserve data for the long term, a system that uses threshold schemes must incorporate recovery protocols able to overcome server failures, adapt to changing availability or confidentiality requirements, and operate in a decentralized manner.

To support the thesis, I present the design and experimental performance analysis of a *verifiable secret redistribution* protocol for threshold sharing schemes. The protocol redistributes shares of data from old to new, possibly disjoint, sets of servers, such that new shares generated by redistribution cannot be combined with old shares to reconstruct the original data. The protocol is decentralized, and does not require intermediate reconstruction of the data; thus, one does not create a central point of failure or risk the exposure of the data during protocol execution. The protocol incorporates a verification capability that enables new servers to confirm that their

continued on page 13

continued from page 7

Efficient Byzantine-tolerant Erasure-coded Storage

Goodson, Wylie, Ganger & Reiter

Proceedings of the International Conference on Dependable Systems and Networks (DSN-2004). Palazzo dei Congressi, Florence, Italy. June 28th –July 1, 2004.

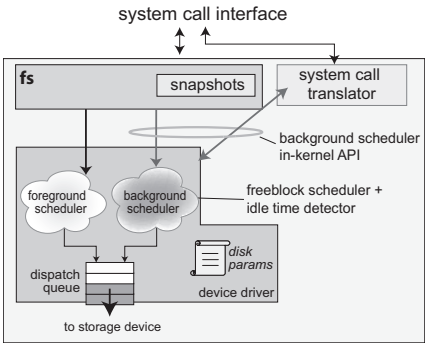
This paper describes a decentralized consistency protocol for survivable storage that exploits local data versioning within each storage-node. Such versioning enables the protocol to efficiently provide linearizability and wait-freedom of read and write operations to erasure-coded data in asynchronous environments with Byzantine failures of clients and servers. By exploiting versioning storage-nodes, the protocol shifts most work to clients and allows highly optimistic operation: reads occur in a single round-trip unless clients observe concurrency or write failures. Measurements of a storage system prototype using this protocol show that it scales well with the number of failures tolerated, and its single request response time compares favorably with an efficient implementation of Byzantine-tolerant state machine replication.

A Framework for Building Unobtrusive Disk Maintenance Applications

Thereska, Schindler, Bucy, Salmon, Lumb & Ganger

Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST '04). San Francisco, CA. March 31, 2004.

This paper describes a programming model and system support for clean construction of disk maintenance applications. Such applications expose the disk activity to be done, and then process completed requests as they are reported. The system ensures that these applications make steady forward progress without competing for disk access with a system's primary applications. It opportunistically completes maintenance requests by using disk idle time and free-block scheduling. In this paper, three disk maintenance applications (backup, write-back cache destaging, and disk layout reorganization) are adapted to the system support and evaluated on a FreeBSD implementation. All are shown to successfully execute in busy systems with minimal (e.g., <2%) impact on foreground disk performance. In fact, by modifying FreeBSD's cache to write dirty blocks for free, the average read cache miss response time is decreased by 15–30%.



Freeblock system components.

MEMS-based storage devices and standard disk interfaces: A square peg in a round hole?

Schlosser & Ganger

Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST '04). San Francisco, CA. March 31, 2004.

MEMS-based storage devices are a new technology that is significantly different from both disk drives and semiconductor memories. These differences motivate the question of whether they need new abstractions to be utilized by systems, or if existing abstractions will work well. This paper addresses this question by examining the fundamental reasons that the abstraction works for existing systems, and by showing that these reasons hold for MEMS-based storage. This result is borne out through several case studies of proposed roles MEMS-based storage devices may take in future systems, and potential policies that may be used to tailor systems' access to MEMS-based storage. We argue that when considering the use of MEMS-based storage in systems, their performance should be compared to that of a hypothetical disk drive that matches the speed of a MEMS-based storage device. We discuss exceptional workloads that can use specific features of MEMS-based storage devices and that may require extensions to current abstractions. Also, we consider the ramifications of the assumptions that are made in today's models of MEMS-based storage devices.

Diamond: A Storage Architecture for Early Discard in Interactive Search

Huston, Sukthankar, Wickremesinghe, Satyanarayanan, Ganger, Riedel & Ailamaki

Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST '04). San Francisco, CA. March 31, 2004.

This paper explores the concept of early discard for interactive search of unindexed data. Processing data inside storage devices using downloaded searchlet code enables Diamond to perform efficient, application-specific filtering of large data collections. Early discard helps users who are looking for "needles in a haystack" by eliminating the bulk of the irrelevant items as early as possible. A searchlet consists of a set of application-generated filters that Diamond uses to determine whether an object may be of interest to the user. The system optimizes the evaluation order of the filters based on run-time measurements of each filter's selectivity and computational cost. Diamond can also dynamically

continued on page 15

continued from page 14

partition computation between the storage devices and the host computer to adjust for changes in hardware and network conditions. Performance numbers show that Diamond dynamically adapts to a query and to run-time system state. An informal user study of an image retrieval application supports our belief that early discard significantly improves the quality of interactive searches.

Atropos: A Disk Array Volume Manager for Orchestrated Use of Disks

Schindler, Schlosser, Shao, Ailamaki & Ganger

Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST '04). San Francisco, CA. March 31, 2004.

The Atropos logical volume manager allows applications to exploit characteristics of its underlying collection of disks. It stripes data in track-sized units and explicitly exposes the boundaries, allowing applications to maximize efficiency for sequential access patterns even when they share the array. Further, it supports efficient diagonal access to blocks on adjacent tracks, allowing applications to orchestrate the layout and access of two-dimensional data structures, such as relational database tables, to maximize performance for both row-based and column-based accesses.

File Classification in Self-* Storage Systems

Mesnier, Thereska, Ellard, Ganger & Seltzer

Proceedings of the First International Conference on Autonomic Computing (ICAC-04). New York, NY. May 2004.

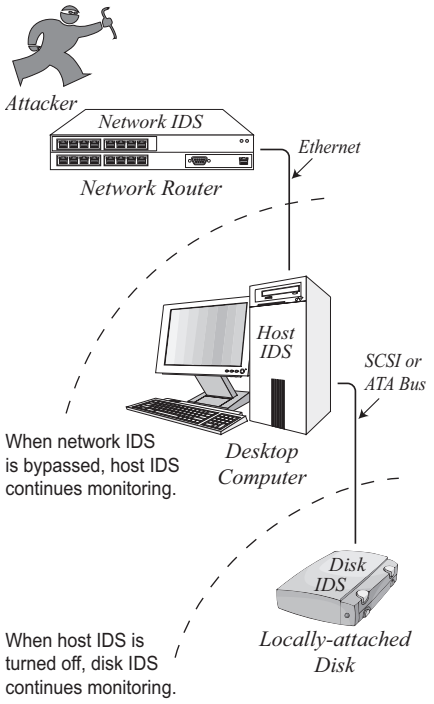
To tune and manage themselves, file and storage systems must understand key properties (e.g., access pattern, lifetime, size) of their various files. This paper describes how systems can automatically learn to classify the properties of files (e.g., read-only access pattern, short-lived, small in size) and predict the properties of new files, as they are created, by exploiting the strong associations between a file's properties and the names and attributes assigned to it. These associations exist, strongly but differently, in each of four real NFS environments studied. Decision tree classifiers can automatically identify and model such associations, providing prediction accuracies that often exceed 90%. Such predictions can be used to select storage policies (e.g., disk allocation schemes and replication factors) for individual files. Further, changes in associations can expose information about applications, helping autonomic system components distinguish growth from fundamental change.

On the Feasibility of Intrusion Detection Inside Workstation Disks

Griffin, Pennington, Bucy, Choundappan, Muralidharan & Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-03-106. December, 2003.

Storage-based intrusion detection systems (IDSes) can be valuable tools in monitoring for and notifying administrators of malicious software executing on a host computer, including many common intrusion toolkits. This paper makes a case for implementing IDS functionality in the firmware of workstations' locally attached disks, on which the bulk of important system files typically reside. To evaluate the feasibility of this approach, we built a prototype disk-based IDS into a SCSI disk emulator. Experimental results from this prototype indicate that it would indeed be feasible, in terms of CPU and memory costs, to include IDS functionality in low-cost desktop disk drives.



The role of a disk-based intrusion detection system (IDS). A disk-based IDS watches over all data and executable files that are persistently written to local storage, monitoring for suspicious activity that might indicate an intrusion on the host computer.

Integrating Portable and Distributed Storage

Tolia, Harkes, Kozuch & Satyanarayanan

Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST '04). San Francisco, CA. March 31, 2004.

We describe a technique called lookaside caching that combines the strengths of distributed file systems and portable storage devices, while negating their weaknesses. In spite of its simplicity, this technique proves to be powerful and versatile. By unifying distributed storage and portable storage into a single abstraction, lookaside caching allows users to treat devices they carry as merely performance and availability assists for distant file serv-

continued on page 18

DISSERTATION ABSTRACTS

continued from page 13

shares can be used to reconstruct the original data.

PH.D. DISSERTATION

Matching Application Access Patterns to Storage Device Characteristics

Jiri Schindler, ECE
August 22, 2003

Conventional computer systems have insufficient information about storage device performance characteristics. As a consequence, they utilize the available device resources inefficiently, which, in turn, results in poor application performance. This dissertation demonstrates that a few high-level, device-independent hints encapsulating unique storage device characteristics can achieve significant I/O performance gains without breaking the established abstraction of a storage device as a linear address space of fixed-size blocks. A piece of system software (here referred to as storage manager), which translates application requests into individual I/Os, can automatically match application access patterns to the provided characteristics. This results in more efficient utilization of storage devices and thus improved application performance.

This dissertation (i) identifies specific features of disk drives, disk arrays, and MEMS-based storage devices not exploited by conventional systems, (ii) quantifies the potential performance gains these features offer, and (iii) demonstrates on three different implementations (FFS file system, database storage manager, and disk array logical volume manager) the benefits to the applications using these storage managers. It describes two specific attributes: the access delay boundaries attribute delineates efficient accesses to storage devices and the parallelism attribute exploits the parallelism inherent to a storage device. The two described performance attributes mesh well with existing storage manager data structures, requiring minimal changes to their code. Most importantly, they simplify the error-prone task of performance tuning.

Exposing performance characteristics has the biggest impact on systems with regular access patterns. For example in database systems, when decision support (DSS) and on-line transaction processing (OLTP) workloads run concurrently, DSS experiences a speed up of up to 3X, while OLTP exhibits a 7% speedup. With a single layout taking advantage of access parallelism, a database table can be scanned efficiently in both dimensions. Additionally, scan operations run in time proportional to the amount of query payload; unwanted portions of a table are not touched while scanning at full bandwidth.

URSA MAJOR

continued from page 11

been renovated to have more power, cooling, and communication capabilities; it can now house approximately 8 racks of storage bricks, assuming that each rack's equipment draws 8-10 kilowatts (KW) of power, and connect them to the rest of campus with multiple gigabit Ethernet links. In addition, we have been allocated 2000 square feet in the new building for an additional, larger machine room, and this room is being designed to support high-power computation infrastructure in addition to our large-scale storage infrastructure. Overall, the room provides space for 50-60 racks of equipment plus the associated power distribution and cooling equipment, assuming a forward-looking average of 20 KW per rack.

The power and cooling challenges are substantial and all new to CMU's physical facilities staff; HP and APC experts, among others, having been helping advise us on how to get things right, and we are planning long-term collaborations on dynamic thermal management and power-savings research.

On the equipment front, we still have a ways to go. We recently submitted two sizeable infrastructure grants (one to NSF and one to DoD), which would provide approximately half of our deployment target (25 racks of storage bricks). We are hoping for our industry partners to provide the other half, enabling the large-scale experiences needed to truly explore the self-* storage vision. Good initial progress has been made. Generous donations from Intel, IBM, and Seagate have provided two racks for our current development and testing of the Ursa Major software. NSF and DoD funds have roughly matched these donations so far. In addition, Cisco has donated networking equipment to establish multi-gigabit paths between racks and from the machine room to the campus backbone.

Status & Plans

We have made significant progress in developing the software for Ursa Major. Several experimental versions of the fault-tolerance protocols and server storage mechanisms have been built in previous projects, and we have been refining both their designs and their implementations. A client library for providing a clean object API has been created, and an NFS server has been ported atop it; this provides an interface that unmodified clients can use to access constellation storage. First instances of metadata and event services have been created

continued on page 20

Chuck Cranor

We are pleased to welcome Dr. Chuck Cranor to the PDL. He joined Carnegie Mellon as a system scientist last December and is working with us to make the Self-* Storage project real. Prior to moving to Pittsburgh, Chuck worked at AT&T Labs-Research in Florham Park, New Jersey. Chuck received his B.S. in Electrical Engineering from the University of Delaware, and M.S. and D.Sc. in Computer Science from Washington University in St. Louis, Missouri. As part of his graduate research he wrote the UVM Virtual Memory system, which is in worldwide use as part of the kernel of the NetBSD and OpenBSD open-source operating systems projects.

Chuck is interested in systems research in the areas of computer operating systems and networking. In particular, he is interested in the design of low-level systems software that is used to control a computer's resources and build higher-level applications and services. He feels that understanding a wide range of hardware and kernel-level software structures enables one to design kernel and application-level interfaces that are well optimized, clean, safe, and portable.

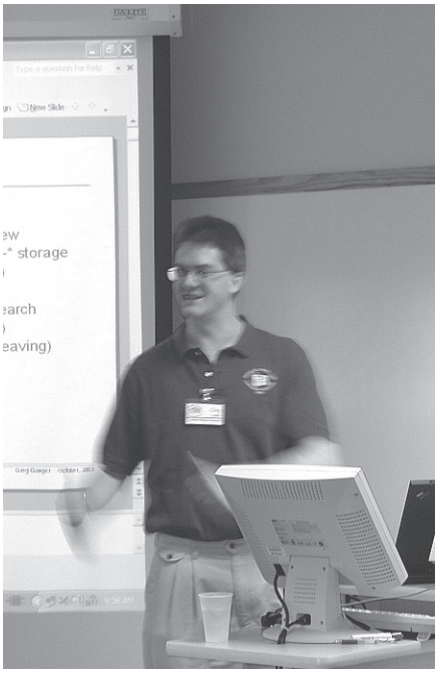
Other areas Chuck has worked in include high speed network monitoring, packet telephony, and system on a chip embedded devices. His past experience maintaining and remodeling his research group's computer lab at AT&T is proving invaluable in the development of Ursa Major and other complete systems.

David O'Hallaron

David O'Hallaron is an Associate Professor in the departments of Computer Science and Electrical and Computer Engineering at Carnegie Mellon University. He received his Ph.D. from University of Virginia. After a stint at GE, he joined the Carnegie Mellon faculty in 1989.

Prof. O'Hallaron works on parallel and distributed computer systems. He is currently leading (with Jacobo Bielak) the Carnegie Mellon Quake project, which is developing the capability to predict the motion of the ground during strong earthquakes, and the Euclid project, which is developing computational database systems that represent massive scientific datasets as database structures and that perform the scientific computing process by creating, querying, and updating these databases. He is also leading (with Satya, Casey Helfrich, and Mike Kozuch) the pilot deployment on the Carnegie Mellon campus of Internet Suspend/Resume (ISR), which combines virtual machines and distributed storage systems to allow people to access their personal computers from any other computer. The new PDL petabyte data storage system, Ursa Major, would be an ideal repository for storing and analyzing both the massive Quake datasets and the ISR virtual machine state.

In 1998 the CMU School of Computer Science awarded Prof. O'Hallaron and the other members of the CMU Quake Project the Allen Newell Medal for Research Excellence. In 2000, a benchmark he developed for the Quake project was selected by SPEC for inclusion in the influential CPU2000 and CPU2000omp (Open MP) benchmark suites. In November, 2003, Prof O'Hallaron and the other members of the Quake team won the 2003 Gordon Bell Prize, the top international prize in high performance computing. In Spring 2004, he was awarded the Herbert Simon Award for Teaching Excellence by the CMU School of Computer Science. With Randy Bryant, he recently published a new core computer systems text (Computer Systems: A Programmer's Perspective, Prentice Hall, 2003) that has been adopted by numerous schools worldwide.



Greg welcomes our PDC members to the PDL Spring Visit Day.

continued from page 15

ers. Careless use of portable storage has no catastrophic consequences. Experimental results show that significant performance improvements are possible even in the presence of stale data on the portable device.

Design and Implementation of a Freeblock Subsystem

Thereska, Schindler, Lumb, Bucy, Salmon & Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-03-107, December, 2003.

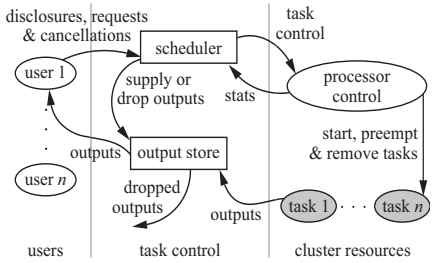
Freeblock scheduling allows background applications to access the disk without affecting primary system activities. This paper describes a complete freeblock subsystem, implemented in FreeBSD. It details new space- and time-efficient algorithms that make freeblock scheduling useful in practice. It also describes algorithm extensions for using idle time, dealing with multi-zone disks, reducing fragmentation, and avoiding starvation of the inner- and outer-most tracks. The result is an infrastructure that efficiently provides steady disk access rates to background applications, across a range of foreground usage patterns.

Scheduling Explicitly-speculative Tasks

Petrou, Ganger & Gibson

Carnegie Mellon University Technical Report CMU-CS-03-204, November 2003.

Large-scale computing often consists of many speculative tasks to test hypotheses, search for insights, and review potentially finished products. For example, speculative tasks are issued by bioinformaticists comparing DNA sequences and computer graphics artists adjusting scene properties. This paper promotes a new computing model for shared clusters and grids in which researchers and end-users



Interaction between users, the scheduler, and the computing center's resources.

exploring search spaces disclose sets of speculative tasks, request results as needed, and cancel unfinished tasks if early results suggest no need to continue. Doing so matches natural usage patterns, making users more effective, and also enables a new class of schedulers. In simulation, we demonstrate how batchactive schedulers significantly reduce user-observed response times relative to conventional models in which tasks are requested one at a time (interactively) or requested in batches without specifying which are speculative. Over a range of simulated user behavior, for 20% of our simulations, user-observed response time is at least two times better under a batchactive scheduler, and about 50% better on average. Batchactive schedulers achieve such improvements by segregating tasks into two queues based on whether a task is speculative and scheduling these queues separately. Moreover, we show how user costs can be reduced under an incentive cost model of charging only for tasks whose results are requested.

A Protocol Family for Versatile Survivable Storage Infrastructures

Goodson, Wylie, Ganger & Reiter

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-03-103, December 2003.

Survivable storage systems mask faults. A protocol family shifts the decision of which types of faults from implementation time to data-item creation time. If desired, each data-item can be protected from different types and numbers of faults. This paper describes and evaluates a family of storage access protocols that exploit data versioning to efficiently provide consistency for erasure-coded data. This protocol family supports a wide range of fault models with no changes to the client-server interface or server implementations. Its members also shift overheads to clients. Readers only pay these overheads when they actually observe concurrency or failures. Measurements of a prototype block-store show the efficiency and scalability of protocol family members.

Balancing Locality and Randomness in DHTs

Zhou, Ganger & Steenkiste

Carnegie Mellon University Technical Report CMU-CS-03-203, November 2003.

Embedding locations in DHT node IDs makes locality explicit and, thereby, enables engineering of the trade-off between careful placement and randomized load balancing. This paper discusses hierarchical, topology-exposed DHTs and their benefits for content locality, and administrative control and routing locality.

A Prototype User Interface for Coarse-Grained Desktop Access Control

Long, Moskowitz & Ganger

Carnegie Mellon University Technical Report CMU-CS-03-200, November 2003.

Viruses, trojan horses, and other malware are a growing problem for computer users, but current tools and research do not adequately aid users

continued on page 19

continued from page 18

in fighting these threats. One approach to increasing security is to partition all applications and data based on general task types, or "roles," such as "Personal," "Work," and "Communications." This can limit the effects of malware to a single role rather than allowing it to affect the entire computer. We are developing a prototype to investigate the usability of this security model. Our initial investigation uses cognitive walkthrough and think-aloud user studies of paper prototypes to look at this model in the context of realistic tasks, and to compare different user interface mechanisms for managing data and applications in a role-based system. For most participants, our interface was simple to understand and use. In addition to a security model that is intrinsically useful, we believe development of this system will inform issues in the design and implementation of usable security interfaces, such as refinement of design guidelines.

Secure Bootstrap is Not Enough: Shoring up the Trusted Computing Base

Hendricks & van Doorn

Proceedings of the Eleventh SIGOPS European Workshop, ACM SIGOPS, Leuven, Belgium, September 2004.

We propose augmenting secure boot with a mechanism to protect against compromises to field-upgradeable devices. In particular, secure boot standards should verify the firmware of all devices in the computer, not just devices that are accessible by the host CPU. Modern computers contain many autonomous processing elements, such as disk controllers, disks, network adapters, and coprocessors, that all have field-upgradeable firmware and are an essential component of the computer system's trust model. Ignoring these devices opens the system to attacks similar to those secure boot was engineered to defeat.

YEAR IN REVIEW

continued from page 4

of Intel Pittsburgh, presented "Dimond: A Storage Architecture for Early Discard in Interactive Search."

- Christos gave a tutorial on "Stream and Sensor Data Mining" at EDBT 2004 in Heraklion Greece.

February 2004

- Chenxi Wang presented "On Timing Attacks Against Low-latency Mix-based Systems" at Financial Cryptography 04 in Florida.

January 2004

- Christos Faloutsos began his sabbatical at IBM Almaden, researching database privacy and data mining on large graphs.

December 2003

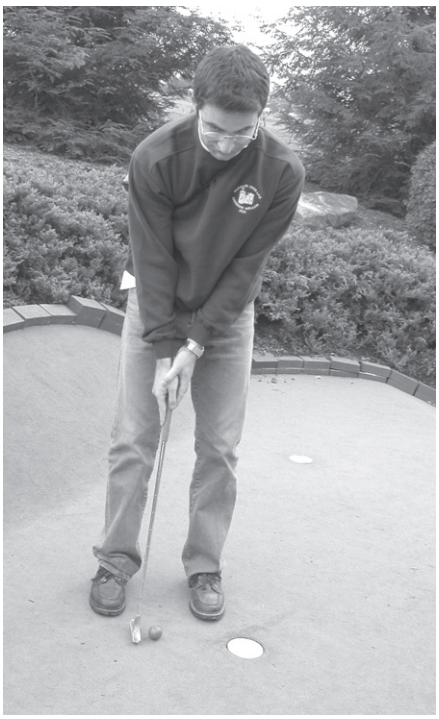
- Ted Wong successfully defended his Ph.D. dissertation "Decentralized Recovery for Survivable Storage Systems."

November 2003

- Greg spoke at Supercomputing 2003 in Phoenix, AZ on "Storage Systems Research at CMU's Parallel Data Lab.
- SDI Speaker: Garth Gibson, of Panasas, Inc., spoke on "Scalable Object-based Storage."
- James Newsome presented "GEM: Graph EMbedding for Routing and Data-centric Storage in Wireless Sensor Networks" at ACM SenSys in Los Angeles, CA.

October 2003

- 11th Annual PDL Retreat and Workshop.
- 8 PDL faculty and students attended SOSP 03 in Bolton Landing, NY.
- SDI Speaker: Craig Harmer, of Veritas Software Corporation, spoke on "Current Considerations and Future Features in a Real World File System."
- SDI Speaker: Howard Gobioff, of Google and also a PDL alumni, presented "The Google File System."
- SDI Speaker: Marc Shapiro, Microsoft Research, spoke on "Modeling Replication Protocols with Actions and Constraints."



Eno works on his putting game at the PDL 2003 Retreat.

Continued from page 18

and are being iteratively extended. New constellation infrastructure software is being created for starting the various components, monitoring their liveness, providing for migration and messaging, etc.

Our plan is to have a complete first instance of Ursa Major running by the end of 2004. This first instance will have the fault-tolerance, versatility, and instrumentation mechanisms described earlier. It will allow us to begin exploring performance tradeoffs and optimization, healing, and diagnosis algorithms in the context of the real system. (Initial explorations will occur in simulation.)

By the end of the academic year, we hope to deploy a refined instance of Ursa Major, with early optimization and healing agents, in support of our first real customer. Of course, initial customers will be carefully selected to have the appropriate expectations, since we expect to have problems in the early stages, possibly resulting in unexpected downtime and even data loss. The earthquake modeling research pursued by Prof. David O'Hallaron's group is an ideal first project, as the large datasets on which they work are created from much smaller observation datasets that are replicated elsewhere. Of course, we will also be exercising the deployed infrastructure with benchmark workloads to provide additional loads.

From customers' perspectives, the storage infrastructure will look like a large file server (initially NFS version 3). This design decision was made to reduce software version complexities and user-visible changes—client machines can be unmodified, and all bug fixes and experiments can happen transparently behind a standard file server interface. The resulting architecture is illustrated in Figure 3. Each user community will mount a single large file system, as though they were the only users, and communicate with a specific (virtual) *head-end system*.

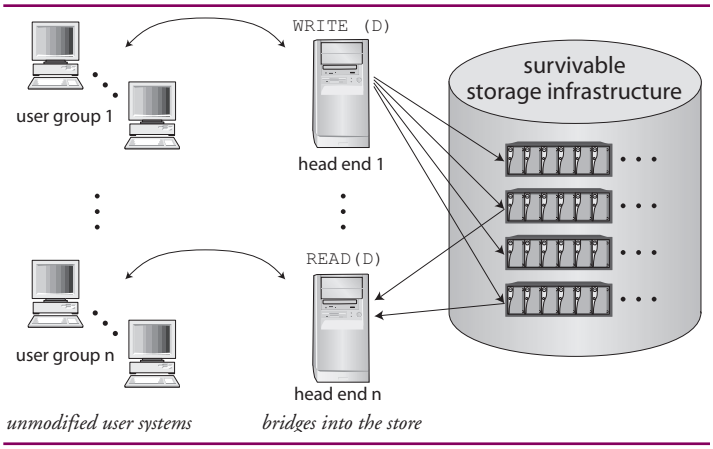


Figure 3. Head-end systems act as bridges into the storage infrastructure, translating NFS commands into the infrastructure's internal protocols.

As the system grows more solid, and our equipment infrastructure expands, we will add additional storage customers. As indicated earlier, the target is a petabyte of usable storage with a wide variety of real customers. Although the final goal is true self-*-ness, all early stages will require sysadmin staff for the deployed infrastructure and also time spent by graduate students. We plan to carefully account for and categorize every man-hour, allowing us to understand and report where their time goes and to quantify improvements made in iterations of the system's self-management features. The system instrumentation will also allow us to capture invaluable long-term information about workloads, data change rates, and hardware and software component failure rates.

Summary

PDL's Self-* Storage project is a big, exciting, scary expedition towards dependable, automated storage infrastructures. Things are off to a good start, and Ursa Major is starting to take shape. Stay tuned, over the next few years, as the project unfolds -- it promises to be very interesting.

Reference

Ganger, G.R. et al. Self-* Storage: Brick-based Storage with Automated Administration. Carnegie Mellon University Technical Report, CMU-CS-03-178, August 2003.

PDL Retreat 2003 attendee group photo.