;login:

THE MAGAZINE OF USENIX & SAGE

April 2002 • Volume 27 • Number 2

inside:

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

conference reports

This issue's reports focus on on the Conference on File and Storage Technologies (FAST 2002) held in Monterey, California, January 28-30, 2002.

Our Thanks to the Summarizers:

Ismail Ari
Scott Banachowski
Zachary Peterson

Conference on File and Storage Technologies

Monterey, California

January 28-30, 2002

KEYNOTE I

Storage: From Atoms to People

Robert Morris, IBM Almaden Research Center

Summarized by Zachary Peterson

Dr. Morris began by defining the importance and motivation of the FAST conference. Storage is getting faster and larger. In fact, it has increased by 14 orders of magnitude. However, these increases are only interesting when they aid computer scientists. Morris asserted that "storage determines the way we use computers" and, therefore, is a technology worthy of investigation, the most important existing technology being the hard disk drive.

Morris enumerated the challenges that face the disk drive and how IBM Research plans to addresses them. The greatest of these challenges is the hard, physical limit at which the magnetic properties used to store data no longer hold, called the superparamagnetic limit – a limit that has been passed and re-predicted a few times. IBM has pushed this limit out by various means of manipulating the physical organization of the magnetic media. Making the bits more square and smaller, combined with a layering of magnetic substrates, enables current production drives to achieve greater capacities with a higher signal-to-noise ratio. IBM hopes to continue this trend in their future production disks by reducing the size of bits to a single grain and by utilizing electron beam lithography to create very small and accurate components.

IBM also looks beyond the standard disk drive architecture, and the limitations inherent in such a design, for the future of storage. The disk arm is too confining: "We need disk fingers," said Morris. He went on to introduce microelectromechanical systems, or MEMS-based devices. One MEMS device would contain many read/write heads operating in parallel on a single media surface. IBM has produced a prototype of such a device, called "Millipede," that uses array-heated heads to make pits in a polymer media surface.

Morris concluded by charging the attending researchers of futuristic storage to consider an ideal case where storage devices will be self-organizing, self-optimizing, and self-protecting. He believes the IBM IceCube is the beginning of such devices. Many IceCubes are placed physically contiguous with each other in three dimensions, reducing the space needed to manage a large storage array. When an IceCube fails, it is simply left in the structure, letting the other devices recover around it. This is the first step IBM Research is making toward self-managing storage, and they hope to continue this trend through an ideology called "autonomic computing." This concept transcends storage and will affect all levels of context-based computing. In general, researchers need to move toward an environment where systems should be easy to use and easy to maintain for the end user, while still providing the performance and capacity gains seen in the past.

SESSION: SECURE STORAGE

Summarized by Zachary Peterson

Strong Security for Network-Attached Storage

Ethan Miller, Darrell Long, University of California, Santa Cruz; William Freeman, TRW; Benjamin Reed, IBM Research

Ethan Miller presented a set of security protocols to provide for an on-disk method of securing data in a network-attached storage system. Even someone who absconds with a disk using strong

security cannot gain access to the data. Additionally, the presence of an authentication scheme means that maliciously changed data can be detected.

Miller presented three schemes of security, each offering higher levels of protection with slightly decreased system performance. In scheme 1, each block is secured using public-key encryption and signed using a hash function. Scheme 2 extends this model to include an HMAC for added authentication and security but increases processing time at the client and the server. Scheme 3 avoids using the slow public-key encryption methods used in schemes 1 and 2, and replaces them with a secure keyed-hash approach. Results of these three schemes compared to a baseline system with no security showed that the public-key encryption schemes suffer significantly in sequential I/O operations. However, the last scheme shows only slightly degraded performance, about 1% to 20% degradation, compared to the baseline. This work demonstrates that on-disk security and authentication for network-attached storage can be achieved efficiently using a keyed-hash approach.

A Framework for Evaluating Storage System Security

Erik Riedel, Mahesh Kallahalla, and Ram Swaminathan, HP Labs

Erik Riedel asserted that there is a need for a quantitative evaluation of storage security. This is because storage has unique propreties that differentiate it from other security applications, such as networks. Propreties such as sharing, distribution, and persistence make applying network security ideas unsatisfactory. He went on to develop a framework of security variables, such as user operations, encryption methods, and attacks, that when permuted, expose the benefits and drawbacks for categories of existing storage security. This framework is especially useful for comparing aspects of security and performance for various methods of security. Riedel then showed some trace-driven simulator results that, when applied to the common framework, illustrate that encrypt-on-disk systems are a preferred method of security over encrypt-on-wire, providing the best security for the least effort. The framework and the analysis can be applied to answer questions beyond this particular result and to different environments.

Enabling the Archival Storage of Signed Documents

Petros Maniatis and Mary Baker, Stanford University

Consider a situation where two people agree to a contract, the contract is digitally signed by each person, and archived. Significantly later, one of the signers challenges the contract. What problems arise with the passage of time? Petros Maniatis addressed these issues, providing one possible solution that extends traditional archival storage to support archiving of long-term contracts.

As time passes, issues arise that make it difficult to ensure the long-term validity of signed data, the sensitivity of keys being the most outstanding issue. Keys are lost, names are changed, and digital certificates expire. This issue begs two questions: "Can one trust a 30-year-old signature key?" and "How does one verify such a signature?" Maniatis introduced KASTS, a key archival service that uses time stamping and timed storage of keys as an answer to these questions.

KASTS uses two main components, a Time-Stamping Server (TSS) and a Key Archival Service (KAS), to establish a time of signing and an effective method for verifying old signatures. KASTS uses a versioned and balanced tree for the public keys of signatures. Maniatis argued that this structure is a feasible and effective method of storing keys. For more information, refer to *http://identiscape.stanford.edu/*.

SESSION: PERFORMANCE AND MODELING

Summarized by Scott Banachowski

WOLF – A Novel Reordering Write Buffer to Boost the Performance of Log-Structured File Systems

Jun Wang and Yiming Hu, University of Cincinnati

Log-structured file systems make good use of disk bandwidth by combining several writes into a single sequential disk access. However, one shortcoming of log-structured file systems is the overhead incurred from cleaning. Cleaning is the process of reclaiming space in a segment occupied by obsolete blocks; by rewriting the segment's live blocks to the log, the entire segment is freed.

Jun Wang presented a method (called WOLF) for reducing the cleaning overhead of log-structured file systems. The key idea comes from the observation that file accesses form a bimodal distribution: some files are repeatedly rewritten while others rarely change. If the bimodal distribution of data is classified when written to disk, each type of data can be stored in separate segments. Over time, segments of rewritten data will have almost all their blocks quickly invalidated, and segments of infrequently modified data will accumulate few holes.

WOLF uses an adaptive grouping algorithm to identify active and inactive data, and assigns the data into separate log segments. Using this method, rewritten data may be reordered into a bimodal distribution of segments, leaving little work for the cleaner. The algorithm tracks segment buffer block accesses with reference counters for a time-window of initialized 10 minutes

to determine which kind of segment the data belongs to.

Wang described the performance of a WOLF implementation adapted from the Sprite LFS source. The metric used in measurements was overall write cost, a value that incorporates garbage collection overhead by including the expense of reading and rewriting cleaned blocks. WOLF performed with a 25–35% overall write performance improvement over LFS, and a 53% reduction of cleaning overhead.

Storage-Aware Caching: Revisiting Caching for Heterogeneous Storage Systems

Brian C. Forney, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau, University of Wisconsin

Brian Forney spoke about a storage-aware caching algorithm. The research addresses cache buffer policies that do not work well in heterogeneous storage systems. An example problem is a client with a single cache that presents a uniform workload to both a slow and a fast disk. Data from the fast disk pushes data from the slow disk out of the cache, causing the client to repeatedly access the slow disk for cache misses. The presented solution is to make caching policies aware of device performance.

A cache is partitioned into variable-sized buffers, each with its own policy. This allows the cache to adapt to workload changes. The challenge is deciding how to partition the cache: a static allocation is simple yet wasteful, so a desired approach is to dynamically adjust the partition size according to access patterns. The dynamic algorithm records delays during a window of disk requests to measure the device behavior, and balances the allocation of partitions based on the relative performance.

The caching algorithm was evaluated using a simulation combining a disk model with a network model. The simulation, configured 16 disk RAIDs, was fed a synthetic workload and a Web server trace. Forney found that their implementation performed similarly to LANDLORD, a performance-comparison algorithm rather than an implementation comparison. The simulation showed that their policy alleviated dramatic performance drops due to naïve caching policies when a slow disk is in the system.

An interesting comment posed to Forney was that the assumption of uniform workload may not hold, because systems usually try to migrate infrequently accessed data to slower disks. Forney conceded that this will reduce the effect of the algorithm but added that layout is more of a long-term decision whereas storage-aware caching makes short-term decisions. Both need to be made and made cooperatively. Additionally, there may be situations, such as accessing remote data beyond administrative control of the user of the data, where changing layout may not be possible.

Timing-Accurate Storage Emulation

John Linwood Griffin, Jiri Schindler, Steven W. Schlosser, John C. Bucy, and Gregory R. Ganger, Carnegie Mellon University

John Linwood Griffin talked about the Memulator system developed at CMU. The Memulator is an emulator for MEMS-based storage devices that uses timing-accurate storage emulation (TASE). TASE combines a device simulator with a timing manager so that the emulator responds to input with accurate response times. This tool is helpful for testing MEMS storage devices, because although we can simulate the devices we don't yet have them available to test in real systems with real applications.

The TASE system consists of three components: the communication manager is responsible for simulating the device on a bus and translating bus signals to simulator requests; the data manager uses a RAM-based cache to hold the data stored on the device; and the timing manager keeps the system state, timing info, and the simulation engine. Obvious limitations of a TASE system is that it must be capable of responding to requests faster than the device it emulates, and the memory must be large enough to cache data required by the application using the device.

The TASE was validated by comparing the performance of an emulated disk drive to the real disk. The emulator response time errors were within 0.1% of the real device. To show that nonexistent devices may be tested using real benchmarks, Linwood presented performance results of the MEMS Memulator from the PostMark benchmarks.

SESSION: HANDLING DISASTER SESSION

Summarized by Zachary Peterson

Venti: A New Approach to Archival Data Storage

Sean Quinlan and Sean Dorward, Bell Labs, Lucent Technologies

Best Paper Award

In the absence of his co-workers, Rob Pike spoke on behalf of the authors of the Venti file system. Venti's key contribution is its ability to manage an on-disk archive of data efficiently and quickly. The motivation for this work is that tape is slow and difficult to manage. Additionally, secondary magnetic storage has become cheap and plentiful, meaning that tape is no longer a cost-effective solution. Venti accomplishes a new architecture for tapeless archiving by providing a write-once block interface to all files. In essence, all files are copy-on-written, creating a new, authenticated version of a file for every write performed.

Authentication and manageability are satisfied by using the SHA1 hash function that distills all data blocks into a 20-byte digest. Pike quipped that "Venti can compress any amount of data to 20 bytes." This SHA1 digest can further be used to identify redundant data blocks for reclamation to free disk space. However, Pike believed this idea of reclamation is transient. With average disk sizes increasing so rapidly, it's time to "let go" of issues of capacity management.

In implementation, the SHA1 performed reasonably fast (60MBps at 700MHz) and provided an efficient random access facility to any blocks in the file system. Results from a Venti prototype show that after four years of file system activity, Venti experienced only a 10% increase in file system overhead, and could effectively reduce the size of data stored on a file system by up to 76%.

Myraid: Cost-Effective Disaster Tolerance

Fay Chang, Minwen Ji, Shun-Tak Leung, John MacCormick, Sharon Perl, and Li Zhang, Compaq SRC

John MacCormick spoke about achieving fault tolerance in a distributed RAID environment. The research presented introduces a distributed storage system called Myriad, which claims to achieve similar levels of fault tolerance and performance as a local RAID with mirroring. He began with an example of how Myraid operates. Assume a system that has many sites connected by a WAN. To achieve high fault tolerance, each site's data blocks exist in a "redundancy group." This group protects data from disaster by using cross-site checksums and erasure codes. MacCormick argues that this distributed method of checksums is more cost effective than mirroring. Essentially, this architecture can use less disk, hence decreasing the total cost of ownership, while not sacrificing reliability.

Implementing cross-site checksums presented some challenges, which the speaker addressed. First, checksum updates are not idempotent and must use version numbers to enforce consistency. Additionally, the overwriting of data blocks can "unprotect" other blocks in the same redundancy group. Therefore, overwrites are not done in place. MacCormick showed results of this architecture that illustrated a high level of reliability, similar to a double-mirrored RAID. Myraid was also shown to be able to reduce the total cost of ownership by up to 25%.

SnapMirror: File-System-Based Asynchronous Mirroring for Disaster Recovery

R. Hugo Patterson, Stephen Manley, Mike Federwisch, Dave Hitz, Steve Kleiman, and Shane Owara, Network Appliance

Computer data has become so important to its owners that data loss and downtime is not only an inconvenience but, in many cases, can cost an enterprise millions of dollars in revenue. Backing up data has two coarse-grain solutions. Tape can be used to perform daily snapshots, which is cheap but still faces the possibility of losing an entire day's work. A more reliable approach is to use an online, synchronized mirroring of data; however, this can be very expensive for large data sets and can be bandwidth intensive. Hugo Patterson presented work that finds a medium between these two poles, called SnapMirror. SnapMirror is an asynchronous, periodic mirroring tool that uses batches updates to maintain data integrity. The frequency and size of batches can be tuned to increase reliability, but this also increases the need for network bandwidth. The key idea of this work is that by lagging behind, and not performing synchronous updates, that a SnapMirror system can reduce cost, and bound potential data loss.

SnapMirror uses the existing WAFL file system metadata to find block-level differences in the updates, thus avoiding full scans of the data. New data is written to new disk blocks, so that by copying newly allocated disk blocks one has copied all newly written disk blocks. The results presented showed that, on average, using asynchronous 15-minute updates can reduce data transfers by 50%, hourly updates by 58%, and daily updates by over 75%. The results indicate that by adjusting the "frequency knob" on updates, SnapMirror can fill the cost and performance void between tape-based and synchronous mirroring-based backup systems.

WORK-IN-PROGRESS REPORTS

Summarized by Ismail Ari

Chair Scott Brandt gathered the speakers and explained the rules: "14 short talks, 8.57 minutes per talk, just enough to introduce your idea." Scott also promised not to tackle them off the podium as long as they obeyed the rules. The players were excited, the whistle was blown, and the game began.

These papers can be found at: *http://www.usenix.org/events/fast02/wips.html.*

StorageAgent: An Agent-Based Approach for Dynamic Resource Sharing in a Storage Service Provider (SSP) Infrastructure

Sandeep Uttamchandani, IBM Almaden Research Center

The resources to be shared in a storage server are cache, memory, and CPU. This paper presents an agent-based architecture to improve the throughput and latency of data access by leasing resources to agents and reclaiming these resources when required. This way the client-sharing is not ad hoc but is controlled and efficiently utilized by a resource manager. Beyond the data allocation and performance monitoring, the

architecture also envisions agent monitoring and data security via access tickets.

NFS over RDMA

Brent Callaghan, Sun Microsystems

NFS traffic over gigabit networks takes 90% of the CPU while resulting in low throughput. Using Remote Direct Memory Access (RDMA) protocols, they expect NFS to make full and efficient use of gigabit networks. NFS over RDMA over transport (GigE, IWARP, FC, InfiniBand, etc.) is claimed to be much more efficient than NFS over TCP/IP or UDP/IP over transport. The benefits of this system were compared to DAFS and TCP Offload Engines (TOE) by the audience.

The Case for Massive Arrays of Idle Disks (MAID)

Dennis Colarelli, Dirk Grunwald, and Michael Neufeld, University of Colorado, Boulder

The talk focused on the power usage of disk arrays and how the power cost could be reduced by file migration and disk spin-down techniques to make the RAIDs (or MAIDs) comparable in price to the tape libraries. Even if we assume the disk and tape unit prices to be the same, the big difference in power usage of these two archival storage systems renders disk arrays a costlier choice. At 7.25 cents per KW/h assuming a 24x7 data center operation, it would cost $9,400 to power the tape library system vs. $91,500 to power the disks in the disk array.

Federated File Systems for Clusters with Remote Memory Communication

Suresh Gopalakrishnan, DiscoLab, Rutgers University

Federated file system (FedFS) provides global namespace for distributed applications and is built as a layer on top of local file systems. While file access, permissions, and consistency are taken care of by the local file system, load balancing, file migration and global naming are handled by FedFS. Virtual directory entries are cached at each node and updates are exchanged periodically. File lookups go through the virtual directory managers.

An Iterative Technique for Distilling a Workload's Important Performance Information

Zachary Kurmas, Georgia Tech; Kimberly Keeton, HP Labs.

The idea is to extract information from a workload trace to synthetically generate workloads with similar performance. They try to obtain useful attribute sets (mean request size, inter-arrival times, run counts) by subtracting from or adding to initially chosen attributes and measuring if the last action has changed the response-time distribution. If not, then this attribute is not included. The goal is to maximize the potential benefit of all attributes in a certain attribute group. The usefulness and viability of the technique was questioned for systems not trained with that specific workload.

Larger Disk Blocks or Not?

Steve McCarthy, Mike Leis, and Steve Byan, Maxtor Corporation

To continue with the 100% doubling of disk capacity, obstacles to increased bits per inch should be removed. In his presentation Steve McCarthy proposed an increase in sector size from 512 bytes to 4096 bytes. The additional capacity gain will be 10–12%. However, the question was whether the gain from larger sector size would be lost back due to internal fragmentation.

Lazy Parity Update: A Technique to Improve Write I/O Performance of Disk Array Tolerating Double Disk Failures

Young Jin Nam, Dae-Woong Kim, Tae-Young Choe, and Chanik Park, Pohang University of Science and Engineering, Kyungbuk, Republic of Korea

RAID6 can tolerate double disk failures, however the write I/O performance is 66% of RAID5. Young Jin Nam presented a lazy parity update (LPU) technique to improve the write I/O performance in RAID6. LPU separates parity groups into a forward parity group (FPG) and backward parity group (BPG), and updates to BPG are deferred until the RAID is idle.

The Armada Framework for Parallel I/O on Computational Grids

Ron Oldfield and David Kotz, Dartmouth College

Ron Oldfield presented the Armada framework for building I/O-access paths for data-intensive grid applications. Filtering the tremendous amounts of raw data is the bottleneck for these applications. They propose a distributed filter usage. They are trying to reduce the amount of data transferred through the network and provide a mechanism for arranging access to distributed and replicated data sets. Fault tolerance issues are not resolved yet.

IBM Storage Tank[tm]: A Distributed Storage System

D.A. Pease et al., IBM Almaden Research Center; R.C. Burns, Johns Hopkins University; Darrell D.E. Long, University of California, Santa Cruz

David Pease introduced IBM's Storage Tank, a distributed object file system that allows heterogeneous file sharing in SAN. It has load balancing, fail-over processing, and integrated backup and restore capabilities. More information can be found at: *http://www.almaden.ibm.com/cs/storagesystems/stortank*.

DATA PLACEMENT BASED ON THE SEEK TIME ANALYSIS OF A MEMS-BASED STORAGE DEVICE

Zachary N.J. Peterson, Scott A. Brandt, Darrell D.E. Long, University of California, Santa Cruz

Zachary Peterson presented his simulation results for access times in MEMS-based storage devices. He pointed out the similarities and differences between disk and MEMS and explained how these differences should affect data layout in MEMS. He identified equivalence regions, or regions of media that share the same seek time, on the device in which data could be placed efficiently. Someone noted that "CMU says their disk layout works well with MEMS"; Zachary replied, "This work is an alternative approach, and further comparisons must be done," which summarizes his presentation.

LOGISTICAL NETWORKING RESEARCH AND THE NETWORK STORAGE STACK

James S. Plank, Micah Beck, and Terry Moore, University of Tennessee

Micah Beck started by saying, "Everything you know about network storage is wrong" to invite people to view the end-to-end picture in remote storage access rather than the networked storage device alone. He presented the proposed network storage stack that had a Logistical File System, L-Bone, and Internet Backplane Protocol (IBP), an IP equivalent for storage. For details, see the Logistical Computing and Internetworking (LoCI) project at the University of Tennessee: *http://loci.cs.utk.edu*.

ENHANCING NFS CROSS-ADMINISTRATIVE DOMAIN ACCESS

Joseph Spadavecchia and Erez Zadok, Stony Brook University

Erez Zadok claims NFS actually stands for "No File Security." The NFS access model is weak. The server depends on the client to specify the user credentials to use and has no flexible mechanism to map or restrict the credentials given by the client. Second, there is no mechanism to hide data from users who do not have privileges to access it. They address these problems by a combination of (1) range-mapping, which allows the NFS server to restrict and flexibly map the credentials set by the client and (2) file-cloaking, which allows the server to control the data a client is able to view or access beyond normal UNIX semantics.

CONQUEST: BETTER PERFORMANCE THROUGH A DISK/PERSISTENT-RAM HYBRID FILE SYSTEM

An-I A. Wang, Peter Reiher, Gerald J. Popek, UCLA; Geoffrey H. Kuenning, Harvey Mudd College

Andy Wang presented Conquest, which shows what to do with tons of battery-backed RAM or MRAM. Metadata and small files are in memory while only the contents of large files go to disk. Most other file systems are designed for disk.

COOPERATIVE BACKUP SYSTEM

Sameh Elnikety and Willy Zwaenepoel, Rice University; Mark Lillibridge, Compaq SRC; Mike Burrows, Microsoft Research

This paper presents the design of a novel backup system built on top of a peer-to-peer architecture with minimal supporting infrastructure. The system can be deployed for both large-scale and small-scale peer-to-peer overlay networks. It allows computers connected to the Internet to back up their data cooperatively.

KEYNOTE II

AVAILABILITY AND MAINTAINABILITY ➔ PERFORMANCE: NEW FOCUS FOR A NEW CENTURY

David Patterson, University of California, Berkeley

Summarized by Scott Banachowski

David Patterson started by listing the three most important aspects in building systems over the past 15 years: performance, performance, and cost performance. However, with so much emphasis on performance, we are forgetting to make systems that people want to maintain. In the future, the metric for comparing computer servers will shift emphasis from performance to availability.

The price of technology continues to decrease (Moore's law), but salaries increase over time, so the total cost of ownership of systems becomes dominated by the salaries of those maintaining them. Patterson showed recent data revealing that a third to a half of the price of systems goes into keeping them running (i.e., paying people to keep the system up). Patterson assured the audience that the world was behind him in his sentiment that availability and maintainability are serious issues by citing the ideas of such luminaries as Jim Gray, Butler Lampson, John Hennessy, and Bill Gates.

Patterson listed new goals for the research community to investigate in the next century: availability, changeability, maintainability, and evolutionary growth (ACME). Systems being used today are failing to meet desired standards in all four areas. Fault tolerance is not solving availability problems, difficult upgrade procedures hinder changeability, systems are unforgiving in their maintainability, and the back end of systems fall short of providing evolutionary growth.

As systems become more automated, the possibility of error now lies with designers and operators. However, automation typically addresses easy tasks and hides the implementation, leaving mistake-prone humans to mess with the harder tasks, operating systems of increased complexity and reduced visibility. Should computer science build margins of safety into their systems, the same way civil engineers beef up a bridge by

adding fudge factors to the design parameters? The challenges that stand in the way of ACME are twofold: hardware and software failures plague us, and human error plagues us. Patterson quoted Shimon Peres (Peres's law): "If a problem has no solution, it may not be a problem, but a fact, not to be solved, but to be coped with over time."

The path Patterson outlined to begin addressing the problem of ACME is to collect data on failures, create ACME benchmarks, start applying margins of safety in designs, and create and evaluate techniques for ACME. For example, a new benchmark might be "time to recover." We can inject failures into systems and measure the QoS during the recovery period. We also need to focus on making designs palatable to operators, not just end users.

Patterson took several questions and comments from the audience; he responded by reiterating the themes of the talk. We need to characterize failures before we can make failure recovery benchmarks; the goal is to build systems that are forgiving of mistakes. An interesting observation is that people in the field are embarrassed by the state of computers. We are proud of the performance, but know that the ACME goals are lacking. When we come up with benchmarks that measure ACME, academics will be happy to do the research because they will have the ability to quantify their results and progress.

SESSION: WIDE-AREA STORAGE

Summarized by Ismail Ari

Safety, Visibility, and Performance in a Wide-Area File System

Minkyong Kim, Landon Cox, and Brian Noble, University of Michigan

The goal is to help mobile clients reach their home file systems without giving up consistency and sharing to avoid WAN overhead. Client updates are held in nearby WayStations for safety and are periodically exchanged between WayStations and servers for update visibility through reconciliation. Due to the danger of out-of-order updates happening between reconciliations, WayStations keep file versions in "escrow" (cache) in the event that they are referenced at other replicas. A WayStation (replica site) that makes an update visible to another via reconciliation must retain a copy of the update for as long as the other replica may refer to it.

The pessimistic bilateral reconciliation that locks the server until a two-phase commit succeeds is compared to an optimistic unilateral reconciliation that assumes reconciliation messages will be received. A prototype consisting of a cache manager, a server, and a WayStation is implemented in Java.

Their trace analysis results show that commits by different users are separated by 1.9–2.9 hours, which gives enough time for WayStations to propagate shared data back to the server before it is needed. Only 0.01% of all operations caused sharing within 15 seconds. Fluid replication's update performance does not depend on wide-area connectivity. Ten megabytes of escrow space was enough, even for the worst cases. Mobile data was both safe and visible. Everybody was happy.

Obtaining High Performance for Storage Outsourcing

Wee Teck Ng, Hao Sun, Bruce Hillyer, Elizabeth Shriver, Eran Gabber, and Banu Ozden, Bell Labs

Since storage outsourcing to Storage Service Providers (SSP) is becoming a big market, it is important to do a performance and viability analysis of remote storage access over various network conditions.

This research implements a real testbed with two routers, hosts, disk, disk arrays, and storage gateways. A storage over IP (iSCSI) prototype has been implemented in FreeBSD. A FreeBSD dummynet package was used to introduce variable network delays, bandwidth limitations, and packet losses. A SmartBits program was used to introduce a fixed background traffic. Using this setup, a number of application benchmarks (SSH, Postmark, TPC-C) have been tested with remote-block-level access. The results show that network delay can adversely affect application performance but can be alleviated by caching and application prefetching. For fast networks disks are the bottleneck. More information can be found at: *http://www.bell-labs.com/project/iSCSI*.

PersonalRAID: Mobile Storage for Distributed and Disconnected Computers

Sumeet Sobti, Nitin Garg, Chi Zhang, Xiang Yu, and Randolph Wang, Princeton University; Arvind Krishnamurthy, Yale University

PersonalRAID is designed for a single user to control a number of distributed and disconnected personal storage devices. This device addresses the challenges mobile users are facing: lack of a single transparent storage space ubiquitously available with a certain degree of reliability assurance; inconvenience of manual data movement; and poor performance due to synchronization during disconnects and connects. With PersonalRAID the user never has to perform manual hoarding or manual propagation of data.

PersonalRAID has a log-structured file system (LFS) design, where disconnection is analogous to a graceful LFS shutdown, and connection (and replay) is analogous to LFS recovery. LFS was preferred because of the fast replaying and low recording overheads. The key component between the fixed hosts is the mobile storage device (such as the IBM microdrive) that is called Virtual-A, or VA (movable storage on Windows PCs is

drive "A"). The fixed host is unusable without the VA plugged in, but this is a personal system, and it is assumed that the user will carry the VA with her.

Removing and adding hosts (reconfiguration) is easy. The system can also recover from both the fixed host and VA device losses, the latter being trickier. Several benchmark analyses show that transparency and reliability are achieved without any serious performance penalty.

SESSION: SELF-ORGANIZING STORAGE SYSTEMS

Summarized by Ismail Ari

The three presentations in this session were all made by groups from HP Labs in Palo Alto. Details can be found at: *http://www.hpl.hp.com/SSP*.

Hippodrome: Running Circles Around Storage Administration

Eric Anderson, Michael Hobbs, Kimberly Keeton, Susan Spence, Mustafa Uysal, and Alistair Veitch, HP Labs

Kimberly Keeton talked about the storage system configuration challenges and difficulty of understanding modern workloads. She stated that in today's world, human experts use rules of thumb and trial and error to iteratively configure storage systems, a process that often takes too long and results in incorrectly provisioned systems.

Hippodrome automates the iterative storage system configuration process. In each loop iteration, Hippodrome takes capacity and performance requirements for the workload as input, and efficiently searches a large space of possible storage designs to find a minimal cost design that meets those requirements. Once a satisfactory design has been chosen, that design is implemented by adding or subtracting storage resources and potentially migrating data from an existing configuration to the target configuration. Finally, the running workload is analyzed to learn more about the performance required by the application, which can be used as input to the next iteration of the loop.

Hippodrome starts with as little as capacity information and converges to a valid storage design without further human intervention, often in only a few loop iterations. Convergence time can be decreased through the use of initial performance hints, even if they are inaccurate. Using near-minimal resources, the storage systems designed by Hippodrome provide performance within 15% of solutions determined by human experts. A prototype implementation constitutes a proof of concept for synthetic and email server workloads.

Selecting RAID Levels for Disk Arrays

Eric Anderson, Ram Swaminathan, Alistair Veitch, Guillermo A. Alvarez, and John Wilkes, HP Labs

The manual rule of thumb in RAID-level selection is to "tag" the data with a RAID level before seeing the array configuration. In the automated design a "solver" takes as input the workload description, the target array types and their configuration schemes. The output of the solver is a storage system design capable of supporting that workload.

The initial tests showed that solvers utilizing a pre-tagged workload resulted in expensive solutions, so they switched to using partially adaptive (deciding RAID level associated with an individual store when first assigning the store to an LU) and fully adaptive (changing RAID level of an LU even after stores had been assigned) solvers. Using a set of real workloads, they tested the tagger-based (using "rules of thumb" or analytic performance models) and the adaptive-solver schemes. Best results were obtained when the solver was allowed to revise its RAID-level selection decision at any time (fully adaptive). The benefits of the fully adaptive scheme outweighed its computational costs. This approach produces solutions with average cost/performance 14–17% better than the best results for the tagging solutions and 150–200% better than their worst solutions.

Appia: Automatic Storage Area Network Fabric Design

Julie Ward, Troy Shahoumian, and John Wilkes, HP Labs; Michael O'Sullivan, Stanford University

SAN fabric design consists of connecting hosts to storage devices via hubs and switches and is an NP-hard problem. Manual designs usually over-provision just to be safe. However, SAN fabric is extremely costly. Appia saved millions of dollars by efficient allocation, and it found results in a few minutes that would have taken days otherwise. An even more important property of Appia is that the resulting designs can be proven to be correct, which reduces the chance of human error.

Two efficient algorithms for automatic SAN design are demonstrated in this paper: FlowMerge and QuickBuilder. FlowMerge tries to eliminate port violations (each flow should be assigned to a single port and port number on switches and storage devices are limited) by merging flowsets together. Flowset merges are also performed to eliminate fabric nodes to reduce cost where possible. QuickBuilder assigns flows to ports and then recursively builds fabric modules for each independent port group. In general, FlowMerge works better with sparse connectivity requirements and QuickBuilder with dense connectivity requirements.

SESSION: THE FUTURE OF STORAGE TECHNOLOGY

Summarized by Zachary Peterson

Future Magnetic Recording Technologies

Mark Kryder, Seagate Research

This presentation examined the details of how magnetic recording works today and how physicists are using various technologies to push out the superparamagnetic limit to increase disk density. The superparamagnetic effect causes bits to destabilize and disappear when the media grains become too small. By using a demagnetizing field, researchers have been able to lengthen the mean time to destabilization, an effect they call "thermal relaxing." Using this technology Seagate has been able to create a 101Gb/in^2 density disk, almost three times a dense as the originally predicted limit.

Other methods for increasing density were presented. By making bits more square, a higher signal-to-noise ratio can be achieved while increasing aerial density. Similarly, by recording bits orthogonally to the media, instead of longitudinally, disks can reach greater density with less noise. However, complications in writing at such high densities cause problems on the current magnetic media. Kryder suggests that changing the media to a more self-ordered magnetic array (SOMA) could provide a more viable solution. By using a heat-assisted magnetic recording (HAMR) and a bow-tie antenna in combination with SOMA media, a more focused and exact media head can be developed, promising densities of 50Tb/in^2.

Non-Magnetic Data Storage: Principles, Potentials, and Problems

Hans Coufal, IBM Almaden Research Center

Hans Coufal began his talk by showing the rapidly increasing capacity-and-speed growth rates, along with the exponentially decreasing costs, of disk drives. However, this trend will not continue, or else "we will have a disk with infinite capacity, infinite bandwidth, no latency, and we will give it away for free," Coufal joked. He went on to introduce a number of non-magnetic storage devices, or advanced data storage devices that, he feels, will play a significant role in the future of storage.

MRAM is an emerging variation of non-volatile RAM that uses magnetics to provide a low-power, non-rotating media, produced at the cost of DRAM, with the speed of SRAM. Coufal went on to show that the ultimate in capacity can be achieved; individual bits represented by xenon atoms have been demonstrated in a laboratory environment. A practical device using this technology, however, is in the distant future. The next device shown was the IBM Millipede, a MEMS-based device that uses 32x32 read/write heads in parallel on polymer media, achieving a 40Gb/in^2 density. IBM has already produced a working prototype of this device. Holographic storage also promises to be an useful and interesting medium. Shining a focused laser beam into a recording medium produces an entire page of memory. IBM has built a holographic device with a 150Gb/in^2 density with a throughput of 1Gb/sec. What is really exciting is that this storage can be organized associatively, so that one would send a filtered reference request beam for "airplanes" and be returned an entire data page of airplane photographs.

Coufal warned that although these new storage devices are technologically feasible, their integration into actual products may take time. It is just a question of economics.

Storage Bricks Have Arrived

Jim Gray, Microsoft Research

When disk drives first came into existence in the 1950s, every drive came equipped with a software interface that enabled people to use it. Drives have changed dramatically since then, but Jim Gray of Microsoft Research concludes that we will return to the built-in software interface paradigm. Every disk will be its own computer, with processors, memory, and interfacing software; something printers already do today. Consider a disk that runs Oracle or DB2, or both! This level of abstraction provides a pleasant layer to a user who wants performance and ease of use. Gray claimed that the central processing unit is not busy and should be thrown away, or more precisely, given a more specific task to do.

An on-disk processor could, for instance, optimize disk arm placement, control network connections, or perform parallel I/O operations to many disks controlled by one CPU. In fact, some companies, such as Network Appliance, are already designing disks using this technology. By making disks into computers, a reduction in maintenance and per-byte cost can be achieved. These cost-benefit advantages make storage bricks a more attractive option than tape for large capacity systems. Additionally, by adding an application layer to the disk, disks become user friendly, essentially a plug-and-play type of device.

SESSION: PARALLEL I/O

Summarized by Scott Banachowski

GPFS: A Shared-Disk File System for Large Computing Clusters

Frank Schmuck and Roger Haskin, IBM Almaden Research Center

IBM's GPFS, a shared-disk file system for parallel clusters, can be configured in a SAN, a symmetric cluster, or as a cluster of I/O nodes. There is an implementation of GPFS running on the ASCI White supercomputer, an LLNL, that has a throughput of 7GB/sec. The architecture combines large disks and RAIDs

into a single file system. Features include wide-striping, large blocks, multiple parallel nodes, byte-range locking, log recovery, RAID replication, and online management.

Schmuck chose to focus his talk on distributed locking mechanisms. GPFS uses token-based locks to avoid excess messages passing to a central server. There is one token server which grants permissions to modify ranges of bytes within a file. When a node wants a currently held token, the holder must flush its data to disk and relinquish the token. For sequential write sharing, this mechanism has very low overhead. To increase performance for concurrent write sharing, a token request lists both currently required data and data desired in the future. This allows flexibility in the amount of data that nodes must relinquish when a token is requested, and leads to efficient concurrent write sharing that requires only a single revocation per node. In GPFS, there is an efficient way to handle metadata updates for shared writes. To prevent lock conflicts for metadata updates, one node is dynamically elected "metanode," and is responsible for collecting and merging the size and mtime updates from the other nodes that modify the file. The GPFS disk allocation map is interleaved so that a region of the map includes space on all disks; a node accessing a region of the map is able to allocate for striping patterns without contention of other nodes.

Some data on the write-sharing throughput of the system was presented; they found performance to scale linearly with number of participating nodes, up to the point where the throughput exceeded the capability of the network switch (about 18 nodes).

Exploiting Inter-File Access Patterns Using Multi-Collective I/O

Gokhan Memik, UCLA; Mahmut Kandemir, Pennsylvania State University; Alok Choudhary, Northwestern University

One problem facing scientific parallel applications is that several processing nodes may try to access multiple data structures simultaneously. Currently, applications either access multiple data structures stored in multiple files with poor performance or access multiple data structures from a single file, an approach that doesn't scale well. Gokhan Memik presented a technique called Multi-Collective I/O (MCIO) which is optimized for these data access patterns. Collective I/O (CIO) aims to help multiple nodes access a single file, but its access patterns don't match the case where the nodes access different data structures in those files.

In CIO, multiple nodes communicate with each other to consider the best way to access files by determining the storage pattern of all nodes' access requests, and then dividing the combined requests so that disks are accessed in an efficient way. Once fetched, the data is then communicated to the nodes that requested it. MCIO expands CIO by considering the interaction of multiple files and assigning the files to different nodes, and then letting each subset of nodes do a collective I/O per file. The problem is shown to be NP-hard for arbitrary-sized files, so it requires heuristic approaches.

Two such approaches are presented: a greedy algorithm that assigns a node to the file it reads the most, and a graph algorithm that solves a maximal matching problem using the Netflow package. Memik described an MCIO experiment where the performance was compared to a naive CIO using some synthetic access patterns. The MCIO algorithm improved response time by 80%. One audience member questioned whether the authors assumed that processor-to-processor bandwidth was greater than the processor-to-I/O bandwidth, which turned out to be the case.

Aqueduct: Online Data Migration with Performance Guarantees

Chenyang Lu, University of Virginia; Guillermo A. Alvarez and John Wilkes, HP Labs

Chenyang Lu presented the Aqueduct system developed for online data migration. Online data migration is used when data must be continuously accessible. The challenges include keeping data consistent and bounding the impact of migration on the performance of other activities. Aqueduct uses feedback control to enforce the QoS latency contracts of foreground operations.

During a migration, the system monitors the average I/O latency over a window of time, and, like a classical control circuit, uses the computed error to adjust the rate of migration. The controller uses an integral feedback loop, so that the rate is proportional to the sum of the worst errors (i.e., contract violations) measured in the migration history. Using the tuning parameters of victim latency (highest latency in the sampling period) and process gain (sensitivity of latency to changes of migration rate), control analysis was used to determine a stable but fast-tracking constant for the feedback loop.

In Lu's experiment, latency was bounded between 0.8 and 1 of the latency contract, meaning that the migration was not too conservative yet met the terms of the contract. He looked at performance experiments on a real enterprise-scale storage system by playing traces generated from an OpenMail workload. In this experiment Aqueduct reduced foreground I/O latency by 76% while reducing the latency contract violation ratio by 78%.

SESSION: LOW-LEVEL STORAGE OPTIMIZATION

Summarized by Scott Banachowski

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

Jiri Schindler, John Linwood Griffin, Christopher R. Lumb, and Gregory R. Ganger, Carnegie Mellon University

Best Student Paper Award

It is well known that random disk access isn't as efficient as sustained streaming. However, when accesses are track-aligned, efficiency increases even for small request sizes. Jiri Schindler presented track-aligned extents (traxtents) to allocate related data within track boundaries. If data is allocated on track-sized extents, all I/O may be aligned within track boundaries, meaning that single access will never incur rotational latency due to crossing a track.

To support traxtents, the file system must have very detailed layout information for each disk. Schindler et al. used an algorithm to determine track boundaries by looking for rotational latencies in disk operations. The operation is time-intensive, but the SCSI command set supports procedures that allow inference of track layout through queries, so the process is simplified for SCSI disks.

Schindler presented the performance results of a traxtents-supporting FFS vs. vanilla FFS. A 1GB file copy resulted in a 20% improvement in runtime, while a diff of two 512MB files resulted in 19% runtime reduction. A video server providing concurrent data streams was able to support 56% more streams due to the improvements in disk throughput and startup latency. Track-aligned extents also reduced the buffer space requirements of the file system. At the end of the talk Schindler was asked how it performed for small accesses, and he replied it performed well in the PostMark benchmarks.

Freeblock Scheduling Outside of Disk Firmware

Christopher R. Lumb, Jiri Schindler, and Gregory R. Ganger, Carnegie Mellon University

Christopher Lumb explored taking advantage of the free bandwidth offered by disks. When a disk head seeks a new position, there are two latency sources: the track seek time and the rotational latency after the head is positioned over the target track. If movement of the disk head is delayed so that it arrives at the target track just-in-time there is no rotational latency, so the time before the head movement may be exploited to do other useful tasks, such as continuing to fetch sequential data after the current read. Lumb found that this "free" bandwidth is available about a third of the time.

Freeblock scheduling is a scheme to take advantage of extra disk bandwidth. A difficulty is that its implementation requires very accurate disk models, especially if scheduling happens outside of the disk firmware (in the driver); a conservative fudge factor must be added to estimates, reducing available free bandwidth. However, they found that they could reduce fudge factors by supplying queued disk commands to the disk such that it never idles.

Freeblock scheduling was tested with a random small I/O workload. Lumb found 3.3MB/sec. of free disk bandwidth with little impact on foreground operation. However, this was lower than the expected 5.3MB/sec. predicted by their model. Lumb attributes this to disk model inaccuracies and confusion in the disk controllers prefetching system. Once the model was altered to reflect new findings, the performance was a closer match to predictions. In concluding remarks, Lumb stated that Freeblock scheduling provides 15% of the disk bandwidth for free.

Configuring and Scheduling an Eager-Writing Disk Array for a Transaction Processing Workload

Chi Zhang, Xiang Yu, and Randolph Y. Wang, Princeton University; Arvind Krishnamurthy, Yale University

Chi Zhang presented the concluding talk of the conference. The subject was scheduling I/Os for transaction processing applications. Transaction processing typically provides random access workloads that exhibit little locality or sequentiality. The research question is "How do we throw away disk space to improve performance?" Eager write is a policy where writes are placed at the free block closest to the current disk-head position, and mirroring allows a read to be serviced by the copy closest to the request. Zhang examines combining these techniques for improving performance in TPC-C (transaction processing) benchmarks, and calls it an eager-writing disk array (EW-Array).

To support an EW-Array, Zhang was faced with the problem of properly configuring the system so that it has enough disk space for eager writes, enough replicas to provided mirroring benefits, and the ability to handle striping. Determining the right configuration is a trade-off depending on the workload. On the other hand, Zhang shows how to design a disk scheduler to take advantage of the write-anywhere nature of eager writing.

A prototype of an EW-Array was developed as a logical disk driver for Windows, and the performance was compared to other disk array configurations by playing TCP-C traces at both original and accelerated rates. Zhang showed results that indicate that EW-Array achieves better response times and higher I/O rates than other approaches given the same extra storage space.