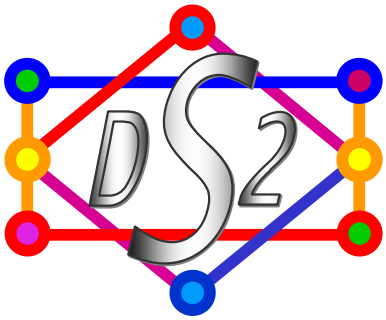
INFORMATION STORAGE INDUSTRY CONSORTIUM

Data Storage Devices and Systems (DS2) Roadmap



January 2005

© 2005 Information Storage Industry Consortium — All Rights Reserved
Reproduction Without Permission is Prohibited

DS2 Workshop Participant Group Photo at CMRR/UCSD – La Jolla, CA – April 28, 2004

Information Storage Industry Consortium
3655 Ruffin Road, Suite 335 - San Diego, CA 92123-1833

insic@insic.org 858-279-7230 www.insic.org

TABLE OF CONTENTS

LIST OF WORKSHOP PARTICIPANTS AND CONTRIBUTORS iii
Acknowledgements v

INTRODUCTION 1

EXECUTIVE SUMMARY 3

APPLICATION-AWARE STORAGE 10
Participants 10
Introduction 10
Application-aware storage and active disks 12
Autonomic storage 13
Archive, backup and disaster recovery 13
Replication and consistency protocols 14
Measurement, modeling, and characterization 15
On-disk layout techniques 15
Naming, organizing and searching 16
Prediction and grouping 17
I/O scheduling 18
Storage transports 19
Versioning systems 19
Volume management 20
Summary 20
Additional areas not discussed at the thrust's meeting 21
Reference 22

ACTIVE STORAGE DEVICES 23
Participants 23
Introduction 23
Active storage devices 23
Challenges 24
Research issues 26
Debates 27
Areas of implementation (additional details) 28
Concerns (additional details) 31
Summary 33
References 34

LONG-TERM STORAGE 36
Participants 36
Introduction 36
Issues in long-term preservation 37
Opportunities 38
Nomenclature 39
Tiers of preservation 40

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Cost and social constraints 43
Desirable technology scenarios 44
Research questions 45
Summary 46
References 47

PERVASIVE STORAGE 49
Participants 49
Introduction 49
Pervasive Storage defined 49
Terminology 50
Assumptions 51
System configuration scenarios 52
Usage scenarios 53
Research questions 56
Research questions related to the basic models 57
Research questions related to consumer usage scenarios 59
Research questions related to corporate citizen usage scenarios 59
Research questions related to sensor network usage scenarios 59
Research questions related to data center usage scenarios 60
Summary 61

SECURITY AND PRIVACY 62
Participants 62
Introduction 62
Research Topics 63
Summary 65

AUTONOMIC STORAGE 66
Participants 66
Introduction 66
Background 68
Self-configuring systems 69
Self-optimizing systems 69
Self-healing systems 70
Self-protecting systems 70
Issues 71
Research directions 74
Summary 77
References 79

LIST OF ACRONYMS 81

TRADEMARKS 82

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

LIST OF WORKSHOP PARTICIPANTS AND CONTRIBUTORS

Thrust	Name	Organization
Application-Aware Storage	**Michael Mesnier, Thrust Leader**	Intel & Carnegie Mellon University
	Dave Burton	Pillar Data Systems
	Dirk Grunwald	University of Colorado
	Steve Halladay	StorageTek
	Meena Kandaswamy	Intel
	Tom Lanzatella	VERITAS Software Corporation
	John Maroney	Quantum
	Richard New	Hitachi Global Storage Technologies
	Julian Satran	IBM
	Jiri Schindler	EMC Corporation
	Nisha Talagala	Sun Microsystems
	James Williams	Oracle Corporation
Active Storage Devices	**Erik Riedel, Thrust Leader**	Seagate Research
	Neil Bannister	Silicon Graphics, Inc.
	Deepak Ganesan	University of California, Los Angeles
	Gordon Hughes	University of California, San Diego
	Will Leonard	Xyratex
	James Nunez	Los Alamos National Laboratory
	Anand Sivasubramaniam	Pennsylvania State University
	David Thiel	Hewlett-Packard
Long-Term Storage	**Tom Ruwart, Thrust Co-Leader**	University of Minnesota
	Giora Tarnopolsky, Thrust Co-Leader	INSIC
	Christopher Bock	NASA
	Randal Burns	Johns Hopkins University
	James Graber	Library of Congress
	Donald Petravick	Fermi National Accelerator Laboratory
	Thomas Schwarz	Santa Clara University
	John Webster	Data Mobility Group
	Gray Zimmerman	BAE Systems
Security & Privacy	**James Hughes, Thrust Leader**	StorageTek
	Roger Cummings	VERITAS Software Corporation
	Howard Gobioff	Google
	Robert Thibadeau	Seagate Research
	Jeff Wisted	Agere Systems

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Thrust	Name	Organization
Pervasive Storage	**Craig Harmer, Thrust Co-Leader**	VERITAS Software Corporation
	Paul Massiglia, Thrust Co-Leader	VERITAS Software Corporation
	Larry Krantz	EMC Corporation
	Reagan Moore	San Diego Supercomputer Center
	Paul Siegel	University of California, San Diego
	Boleslaw Szymanski	Rensselaer Polytechnic Institute
	Hans van Rietschote	VERITAS Software Corporation
	Hu Yoshida	Hitachi Data Systems
Autonomic Storage	**Remzi Arpaci-Dusseau, Thrust Leader**	University of Wisconsin
	David Du	University of Minnesota
	Robert Gray	International Data Corporation
	Les Hodge	StorageTek
	Arnold Jones	Storage Networking Industry Association
	Dilip Kandlur	IBM
	Ethan Miller	University of California, Santa Cruz
	Wayne Rickard	Seagate Technology
	Ahmed Tewfik	University of Minnesota
	Curt Tilmes	NASA
	Geoff Voelker	University of California, San Diego
	Qian Wang	Pennsylvania State University
	John Wilkes	Hewlett-Packard Laboratories
Additional Participants		
	Roger Bohn	Information Storage Industry Center/UCSD
	Andrew Chien	University of California, San Diego
	Matthew Clark	Qualcomm
	Ron Durbin	Information Storage Industry Center/UCSD
	Paul Frank	INSIC
	Sharon Rotter	INSIC
	Amin Vahdat	University of California, San Diego

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Acknowledgements

INSIC gratefully acknowledges the contributions to the DS2 Workshop from the co-sponsoring organizations and, in particular, the following individuals from these organizations, whose efforts made the workshop possible:

The Center for Magnetic Recording Research (CMRR) at the University of California San Diego:

Paul Siegel, Director
Gordon Hughes, Associate Director

The Information Storage Industry Center (ISIC) at UCSD:

Roger Bohn, Director
Ron Durbin, Director of Industry Relations

The California Institute for Telecommunications and Information Technology [Cal-(IT)^2] at UCSD:

Ramesh Rao, Director

The Digital Technology Center (DTC) at the University of Minnesota:

David Du, Intelligent Storage Consortium
Tom Ruwart, Intelligent Storage Consortium Technical Manager

In addition, the following individuals composed the DS2 Technical Committee that organized the 2004 workshop, and their contributions are gratefully acknowledged:

Remzi H. Arpaci-Dusseau, University of Wisconsin, DS2 Thrust Leader
Paul D. Frank, Information Storage Industry Consortium (INSIC)
Craig Harmer, VERITAS Software Corporation, DS2 Thrust Co-leader
Gordon F. Hughes, Center for Magnetic Recording Research, UCSD
James P. Hughes, StorageTek, DS2 Thrust Leader
Thomas W. Lanzatella, VERITAS Software Corporation
Paul Massiglia, VERITAS Software Corporation, DS2 Thrust Co-leader
Michael P. Mesnier, Intel Corporation and Carnegie Mellon University, DS2 Thrust Leader
Erik Riedel, Seagate Technology, DS2 Thrust Leader
Thomas M. Ruwart, Digital Technology Center, University of Minnesota, DS2 Thrust Co-leader
Paul H. Siegel, Center for Magnetic Recording Research, UCSD
Giora J. Tarnopolsky, Information Storage Industry Consortium, DS2 Thrust Co-leader

The hospitality of the Center for Magnetic Recording Research, UCSD, where the DS2 Workshop was held, and the contributions of the CMRR staff for local arrangements are gratefully acknowledged. In addition, we thank Sharon Rotter and Barbara Brittain of INSIC for their substantial organizational contributions to the DS2 Workshop, and similarly thank Judith C. Davis of INSIC for her considerable efforts in formatting and editing the DS2 Roadmap document.

Last, but certainly not least, INSIC acknowledges the outstanding leadership of Giora Tarnopolsky, in the organization of the DS2 Workshop, in the editing of the DS2 Roadmap document, and in the management of INSIC's DS2 Research Program initiative.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

INTRODUCTION

The Information Storage Industry Consortium (INSIC) organized the 2004 Workshop on the Future of Data Storage Devices and Systems (DS2), which was held on April 27-29, 2004, at the University of California San Diego (UCSD). The workshop was co-sponsored by UCSD's Center for Magnetic Recording Research – CMRR, by the California Institute for Telecommunications and Information Technology – Cal-$(IT)^2$, by UCSD's Information Storage Industry Center – ISIC, and by the University of Minnesota's Digital Technology Center - DTC. The purpose of the workshop was to discuss new research opportunities and unresolved technical issues in the discipline of data storage systems and data storage devices, and to formulate a research roadmap that identifies high-reward subjects of inquiry. The conclusions of the workshop's labors are contained in this document.

Data storage systems have become the backbone of a new era, whereby economic output and creative endeavors are dependent on knowledge-based activities and on organizations, governments, and individuals possessing powerful tools to create, access, and manage information. In spite of the profound changes that information technologies have already exerted on society, the data storage systems and devices that enabled those changes are in an early stage of their development. There continues to be rapid evolution and metamorphosis of data storage systems and devices, which are traits of the early stages of new phenomena. Research in data storage systems and devices offers opportunities for managing higher orders of complexity and abstraction than has been possible so far, and thereby offers the potential to create new products and new applications in wide fields of human endeavor.

The Technical Committee of the DS2 Workshop identified six subjects of discussion, or "thrusts", as a useful way of structuring the work. These thrusts are:

Application-aware Storage refers to the introduction of application context within a storage device and in aggregates of storage devices. Example topics include attribute-managed storage, attribute-based learning environments, and semantically smart devices.

Active Storage Devices refers to issues of distributed data processing at the storage device or assemblies of devices. Example topics include programmable storage devices and device-based data mining.

Long-term Storage includes the issues associated with the perpetual preservation of digital content, archiving, and intelligent backup. Example topics include long-term semantic continuity, hardware migration models, information lifecycle management, and curatorial data.

Pervasive Storage includes the storage research issues in a pervasive computing infrastructure, such as ubiquitous digital content creation, storage for disconnected environments, personal servers, and data distribution techniques for roaming users.

Security and Privacy includes the security issues associated with storage devices and storage networks, beyond mechanisms now extant. Example topics include data dispersal and retrieval algorithms, and rightful data access or denial to claimants.

Autonomic Storage includes self-managed storage, models for self-configuring storage, self-securing, self-healing, self-tuning and, in general, self-perpetuating, internally-consistent storage systems.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

The thrusts appear listed along a loosely defined scale of locality of implementation, from device-level to system-level, but not ordered by their complexity: even a single application-aware storage device is algorithmically and logically complex. There are significant areas of overlap among the thrusts' realms of applicability, and this is further discussed in the thrust reports.

INSIC has, since 1991, successfully led "pre-competitive" research programs in technology for magnetic hard disk drives, magnetic data storage tape, holographic storage and optical storage devices. During 1997-1999, INSIC also sponsored the seminal Network-Attached Storage Devices (NASD) pre-competitive research program, whose purpose was to develop, explore, validate and document the technologies required to enable the deployment and adoption of object-oriented, network-attached storage devices and systems. In all of INSIC's programs, pre-competitive research refers to R&D on subjects of undisputed relevance whose solutions are sufficiently removed from marketplace implementation that industrial and institutional cooperative efforts are both possible and desirable. Typically, at the time of inception of a pre-competitive research program, the program's goals are perceived to be so ambitious and distant from practical implementation that their pre-competitive nature is assured. Yet, the technology goals of many of INSIC's research programs have, by and large, been accomplished, and have eventually become integrated into mass-produced devices. An important goal of the INSIC DS2 Workshop was to analyze whether in the arena of storage *systems*, with its heavy reliance on computer science, software and standards, there indeed are such pre-competitive research subjects. The participants in the DS2 Workshop discussed the existence of pre-competitive systems research topics. Their conclusions are contained in the thrust reports that follow in this document. In addition, pre-competitiveness itself was intensely scrutinized during the three-day workshop's informal discussions and wrap-up session; the contents of these exchanges, while not directly stated in the roadmap, were nevertheless very positive as to the suitability of data storage systems subjects as pre-competitive research programs.

The answer that the DS2 Workshop participants gave as to the feasibility of pre-competitive research in storage systems was definitively positive. The primary result that emerges from this roadmap is that there are significant opportunities for expanding the applicability and enhancing the efficiency of data storage systems. The accomplishment of these goals requires solving problems of vast complexity, both in their core technologies and in their interdependencies, and these problems appear eminently well-suited for pre-competitive research programs.

INSIC looks upon the spectrum of systems research described in the DS2 Roadmap as the cornerstone on which, with the support of participating companies and institutions, high return-on-investment pre-competitive research programs in data storage systems and devices will be established.

Giora J. Tarnopolsky
INSIC DS2 Program Leader
Palo Alto, California

Paul D. Frank
INSIC Executive Director
San Diego, California

In the following document, enumerations of pre-competitive research topics and/or issues are denoted thusly:

- Research topic discussions are highlighted in blue, with red bulletization as shown here.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

EXECUTIVE SUMMARY

Data storage systems have grown rapidly in functionality and complexity. All facets of human activity are now affected by the ability to efficiently access and store digital data. Storage systems hardware is relatively inexpensive as compared to the economic value of the data stored in those systems. The data storage capacity of the building blocks of storage systems (tape cartridges and libraries, disk drives, and optical discs) has increased rapidly while, simultaneously, storage device and non-volatile solid state memory costs have decreased. The availability of digital storage in a variety of formats with high data content is affecting all tiers of the storage infrastructure. The realm of applicability and functionality of data storage systems could expand significantly by advances in the science and technology of digital data storage. INSIC and the sponsoring organizations perceive avenues for enhancing the effectiveness of data storage systems research through the development of an industrial, academic and governmental partnership in the pre-competitive research arena.

Opportunities for these advances are classified in the six thrust areas: Application-aware Storage, Active Storage Devices, Long-term Storage, Pervasive Storage, Security and Privacy, and Autonomic Storage. These disciplines have significant areas of overlap, and are loosely classified along a spectrum from closest to the discrete device to the highest abstraction in the system. This DS2 Roadmap identifies areas of pre-competitive research, meaning subjects of undisputed relevance whose solutions are sufficiently removed from marketplace implementation that industrial and institutional cooperative efforts are both possible and desirable.

1.1 Application-aware storage

Application-aware storage devices acquire and use knowledge about applications, usage patterns, and the environment in which the device is working. This knowledge is then used to enhance system performance. For instance, an application-aware device would enhance the efficiency of data layout, caching/pre-fetching strategies, intrusion detection, and Quality of Service (QoS). Application-aware devices do not necessarily run applications code locally, although application-awareness and active storage device features could co-exist in the same device.

An application-aware device could acquire knowledge on, for instance: spatial and temporal access patterns of the data – for better caching and pre-fetching; relationships among data, users and applications – for finding files; data replication factors of data being stored – helpful in data reconstruction and in making efficient use of the storage resources; aggregation information (e.g., part of a RAID5 stripe) – also helpful in data reconstruction; access control lists and what I/O is "normal" – for device-resident anomaly detection; network topologies and host connectivity – for better caching and pre-fetching; power consumption requirements – for reducing costs/heat in the data center; caching hierarchies – to minimize redundant caching among devices; application goals (e.g., performance, availability, reliability, security) – for ensuring QoS; and device-resident de-mapping functions – for mapping file data to specific blocks.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

The "application-awareness" features would be beneficial for a broad range of storage implementations: autonomic/self-* storage[1]; archive, backup and disaster recovery; replication and consistency protocols; measurement, modeling, and characterization; on-disk layout techniques; prediction and grouping; naming, organizing and finding files; I/O scheduling; storage transports; versioning systems; and volume management.

Among an extensive menu of possible performance enhancements, the development of application-aware storage devices encompasses the following research issues:

- The proper languages and mechanisms for transferring knowledge to devices. This includes identifying the right set of application hints for specifying spatial/temporal characteristics of stored data, and establishing machine-learning techniques leading to improving the organization of data within or across devices.

- The manner in which applications specify their goals and requirements and the manner in which storage systems and devices specify their capabilities for meeting applications' demands.

- Mechanisms for standardizing bulk metadata exchange between applications and the storage systems in order to make an effective use of such metadata in backup/restore applications.

- Mechanisms for standardizing information lifecycle management (ILM) related metadata, and research into the effective use of such metadata in the storage device.

- Enhancements to consistency protocols, elimination of redundant caches, data reconstruction, and application-failure recovery.

- Techniques for guaranteeing consistency in the face of application concurrency and Byzantine faults in storage systems that maintain multiple versions of users' data.

- Techniques to automatically discover relationships among data objects within the storage system, and for associating a data object with the context in which it was created and stored, so that subsequent queries for a file are context-sensitive.

- Issues in storage transports, including techniques for matching application-level security needs to capabilities of the transport, and trade-offs between reliability, availability, security and performance.

1.2 Active storage devices

Active storage devices, also called intelligent storage, run application-specific code to perform application-specific functions. Active storage devices provide advanced functions that operate on non-volatile data either through a fixed function or through the use of general-purpose programming within the device. An active storage device must contain non-volatile storage, and may contain other storage devices, as well.

---

[1] As in computing languages, the asterisk refers to a substitution variable. In the construct "self-*" the asterisk could acquire any of the values "healing", "configuring", "optimizing", etc.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Active storage devices could improve performance in many application areas, among them: indexing and searching, including database and data-aware processing; QoS and performance management, traffic analysis, and adaptive computing; scientific computing, mesh computation, cluster computing and grid computing; sensor networks, in-situ processes for data reduction before transmission to upstream nodes; and wise drives, reliability, long-term storage and diagnostics.

There are many unresolved issues for the implementation of active storage devices. Among these issues one finds: the need for an open, standardized application programming interface (API) for a general-purpose programming environment; mechanisms to establish "ownership" of blocks to ensure that only "authorized" blocks are written or overwritten; the management of concurrency between competing active functions in the system; and the assessment of correctness of outcomes in storage systems that, by their ability to perform general-purpose data processing, present a much larger catalogue of possible behaviors than present-day systems.

The research charter for active storage devices includes:

- A model of distributed computation. There is need for a theory of how to distribute the functionality in a system around a computing environment to support reasoning about whole-system characteristics and behavior.

- Resource management at the devices. Such management is needed for handling competing active functions vying for the same resources. The solution to this issue would probably be an event-based, soft real-time system.

- Internal device API for active functions. Establish a specification of how active functions interact with the local hardware environment in a way that is both flexible and powerful.

- Correctness/reliability/stability. Presently, in a disk drive or disk array, most of the corner cases (where parameters of operation acquire values in the far-flung boundaries of their ranges) are tested and the interface command structure is rather limited. In the new active storage system, that may not be true – there are now many more dimensions to the problem.

- Specialized hardware for fixed functions. This is not a critical path issue, but offers a rich field of exploration – hardware-optimized functions may prove valuable in some settings.

1.3 Long-term storage

The adoption of computers and their application to all fields of human endeavor, together with the availability of efficient data transport networks, has caused a high rate of digital asset generation and has stimulated demands for the availability of these assets over wide geographical areas and long periods. As the first half-century of the information revolution ends, there is widespread recognition that digital assets will not be available to future generations unless their permanency is purposefully engineered into their data storage and retrieval systems. We are now, as a community, only beginning to wrestle with the subtleties of preserving digital records for much longer than the duration of a single individual's life.

Long-term storage encompasses both the conceptual aspects of storage and its physical implementation. The preservation of digital information needs to address, among other issues:

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

the balance between the rate of asset generation and the available volumetric storage capacity and storage bandwidth; the limited useful lifetime of storage media, devices and systems, presently measured on the order of 10 – 20 years; the rapid evolution of the logical structures used to codify and retrieve the data; the emergence of hitherto unknown intellectual content: for instance, dynamically linked documents from various provenances or digitally animated movies; the ability to discard useless records while perpetuating the desired ones; and the economic and societal resources needed for the long-term preservation of assets.

Complex issues need to be addressed to accomplish long-term digital preservation. It is expected that society will invest heavily in long-term storage, as the alternative would be to compromise business and government processes or to abandon cultural heritages. Among the research topics identified:

- The longevity of the physical bit streams on various archival media and storage mechanisms.
- Integrity of data bit streams in large arrays of storage devices or libraries.
- Diagnostic means for assessing data integrity early enough to avoid failure. Self-diagnosis and self-repairing data systems.
- Hardware and software strategies for static, or only slowly increasing, volumetric densities. How feasible does migration become if the volumetric storage capacity ceases to grow exponentially?
- What is the optimal storage architecture for long-term preservation? Is the current spectrum of off-line tape to near-line to on-line storage a valid scheme for indefinite preservation? What should be the distribution of storage tiers?
- Is migration a reliable preservation technique, when the original bit streams and applications cease to exist in the migration-target system? What are the risks inherent in multiple successive migrations?
- Preservation strategies. Is emulation technologically and economically feasible? How would a Universal Virtual Computer (UVC) be developed, and what changes in procedures for creating digital objects would this approach demand?
- How should operating systems be documented to enable emulation? Is emulation of the hardware a better approach? Will emulation lead to an indefinitely recursive long-term preservation cycle, with the newer generation machines being required to emulate the emulators running in older-generation systems?
- Curatorial strategies. What techniques will ensure that specific records can be accessed at some indefinite time in the future? What metadata should be automatically created with each new digital object to enable this process? How is the authenticity and provenance of digital records to be assured in the future?
- What are the costs of long-term preservation? Can a long-term preservation utility be created and run as a profitable enterprise?

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- How is the preservation of non-traditional documents (active documents, dynamic simulation results, computer animations, algorithm-dependent records) going to be accomplished?

- What are the limits of long-term preservation? What is the volume of data that can be preserved for any finite investment in the preservation infrastructure?

1.4 Pervasive storage

Research in pervasive storage would address untapped opportunities that have been created by the proliferation of devices that possess significant data capacity – of either previously stored data or newly created data. The focus is not at all on the devices, but whether the (possibly hundreds of millions of) devices with multiple gigabytes of distributed storage (data centers, workstations, notebook computers, storage appliances like Apple Corporation's iPod® and digital cameras, all capable of distant data transport) can be integrated into federated or otherwise unified systems that could provide services not available today. Pervasive storage is about pervasive access to information which must support the "disconnected" as well as connected operation of devices that create or otherwise possess digital data. The underlying assumptions about technology scenarios are that storage will be inexpensive and readily available, and even low-cost devices will either be equipped with multiples of tens of gigabytes of storage or could create such data volume even if not locally storing it within the physical boundary of the device. A second assumption addresses connectivity, meaning that sustained, broad bandwidth will be available for the transmission of data among storage devices. Different configurations emerge, with a dichotomy between storage cells and interconnected storage farms. Some of the beneficiaries of pervasive storage would be consumers, individual "corporate citizens", sensor networks, and data centers.

A broad spectrum of high return-on-investment research topics emerges, among them:

- Modeling the relative merits and operational characteristics of storage cells vs. pure storage farms.

- Name-space management, the creation of universal naming schemes that ensure that information objects are uniquely identified regardless of "home" location. Name space consistency and the implementation of name-space "federation", which would enable the use of name spaces that are managed privately but can be aggregated with the global name space so that the information objects the federated name spaces represent can participate in integrated pervasive storage

- Security – the development of strong security models accessible to and understood by broad populations. Access security and disaster recovery policies. Security for applications executing in storage cells.

- Metadata architecture for pervasive storage.

- Data consistency assurance protocols for distributed pervasive architectures.

- Physical data management: cache management, replication algorithms, access paths, reliability models.

- Data filtering and the interplay of pervasive storage with active storage devices.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Connectivity and the architecture of pervasive storage with limited connected duty cycle.
- Economics: infrastructure, deployment strategies, users' acceptance of pervasive storage.

1.5 Security and privacy

Security and privacy for storage systems are part of the research charter of all the thrusts, and are discussed in their individual contexts. Security refers both to the physical integrity of the data and to mechanisms that prevent accidental or malicious threats to the data. Privacy refers to ensuring selective access while maintaining ready access to all rightful agents. Security and privacy issues arise in all tiers of storage system management. The concerns include intrusion detection at all layers, data protection in transit or in repositories, and safeguards on the capability of inside users to compromise storage assets. The following research issues emerge for this thrust area:

- Storage system management without access to the stored data.
- Data encryption at the device, system, and transport levels, and trade-offs between privacy and performance.
- Key management and security of keys. Means to ensure that encrypted data will be retrievable and secure over long periods of time.
- Data destruction. Is the destruction of keys of encrypted data sufficient "erasure"? What is the traceability mechanism that may ensure the destruction of all cached copies and all backup copies of files?
- The development of comprehensive, system-level notion of data integrity and data integrity management to prevent undetectable changes to the data.
- Issues in storage transports, including techniques for matching application-level security needs to capabilities of the transport, and trade-offs between reliability, availability, security and performance.
- Security models accessible to and understood by broad populations. Access security and disaster recovery policies. Security for applications executing in storage cells.

1.6 Autonomic storage

Autonomic storage addresses the complexity of storage systems management by providing techniques for storage systems to manage their own operation. The four elements of autonomic techniques, namely self-configuring, self-optimizing, self-healing and self-protecting storage systems, may yield storage system performance better than that achievable by human system administrators. Self-configuring systems will automate the installation, configuration, and integration of storage systems. A self-optimizing system will continually adjust its operational parameters in order to improve efficiency – without human involvement. Self-healing storage systems will automatically detect, diagnose and repair software and hardware problems. Finally, a self-protecting system will automatically detect and defend against malicious or accidental threats to the data. The thrust work considers the applicability of autonomic ideas to storage systems, and identifies the following research opportunities:

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Techniques to expose and describe the decisions made by autonomic systems must be developed. If computer systems are to make important decisions, those systems must be able to explain their decision-making process to human administrators.

- Devirtualization: Virtualization interferes with performance tuning and problem diagnosis. Tools that can discover the underlying physical nature of devices and larger-scale systems are required.

- Reliability: Research is needed into methods for building robust and reliable systems in a complex and heterogeneous systems environment.

- Evaluation and metrics: For a marketplace of autonomic technologies, standard benchmarks must be developed in order to enable competing products and approaches to be rigorously compared.

- Study of processes and practices: To understand how to build autonomic technology, the actual processes that exist in today's systems must be studied.

- Management policies: As storage automation systems become commonplace, there must be research not only into the systems that enable policy-based automation, but into the actual policies themselves.

- Specialized storage systems: As storage uses become more diverse and specialized, a different approach to building low-cost manageable systems is to specialize the storage systems to the needs of important application classes. Hence, further research into the design and implementation of such specialized systems is recommended.

The detailed reports from each of the six thrust areas comprise the chapters of the remainder of this document, and they include further definition and in-depth exploration of this seemingly vast array of potential data storage systems-oriented research opportunities.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

APPLICATION-AWARE STORAGE

Participants

Michael Mesnier, Intel & Carnegie Mellon University, (Thrust Leader)
Dave Burton, Pillar Data Systems
Dirk Grunwald, University of Colorado
Steve Halladay, StorageTek
Meena Kandaswamy, Intel
Tom Lanzatella, VERITAS Software Corporation
John Maroney, Quantum
Richard New, Hitachi GST
Julian Satran, IBM
Jiri Schindler, EMC
Nisha Talagala, Sun Microsystems
James Williams, Oracle Corporation

Introduction

Application-aware storage devices are those which possess knowledge about the environments in which they operate. Unlike today's block-based devices, which literally cannot distinguish between a free and used block, application-aware devices will possess a much deeper understanding of the application data they store. For example, an application-aware disk drive may understand which blocks composed a certain file, or perhaps have knowledge of the directory structure of a given file system. Similarly, larger storage devices (e.g., RAID arrays, storage servers, and appliances) could have even more application-awareness, given their increased processing power and storage capability.

Modern day file servers have a level of application-awareness that would benefit storage devices. So, in effect, we are proposing that storage devices become more similar to file servers (e.g., by understanding files, directories, users, etc.). However, we can go further. In addition to storage devices evolving to the capability of file servers, additional information as to user and application behavior can be harnessed to provide even higher degrees of service. For example, a storage device that learned user behavior could better help a user search for his files.

In general, application-awareness includes information about the operating systems, applications, users, and data being stored. This knowledge comes in many forms, ranging from anticipated file access patterns to security and QoS requirements, on perhaps a per-user or per-application basis. Specific knowledge includes, but is not limited to, the following:

- Spatial and temporal access patterns of the data – for better caching and pre-fetching
- Relationships among data, users and applications – for finding files
- Data replication factors of data being stored – helpful in data reconstruction
- Aggregation information (e.g., part of a RAID5 stripe) – also helpful in data reconstruction

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Access control lists and what I/O is "normal" – for device-resident anomaly detection
- Network topologies and host connectivity – for better caching and pre-fetching
- Power consumption requirements – for reducing costs/heat in the data center
- Caching hierarchies – to minimize redundant caching amongst devices
- Application goals (e.g., performance, availability, reliability, security) – for ensuring QoS
- Device-resident de-mapping functions – for mapping file data to specific blocks

If a storage device could acquire this knowledge, it could be used to improve storage features such as performance, availability, reliability and security. The challenge is coming up with an established methodology for transferring such knowledge into the storage system. Knowledge can be either statically configured through application hints and protocol extensions (e.g., extensions to the SCSI architecture) or automatically discovered by a running system (e.g., inference-based systems or semantically smart systems). Determining what knowledge is needed for specific optimizations, where in the storage stack this knowledge is best utilized, and the methods for statically or dynamically obtaining such knowledge are all active areas of research and development. In the sections that follow, we summarize what we feel to be the relevant issues of application-aware storage in each of the above focus areas.

The goal of the application-aware storage thrust area is to outline the potential benefits and future research directions for this emerging field. Of particular interest is the potential for application-aware storage in the following focus areas:

- Application-aware storage
- Autonomic/self-* storage[1]
- Archive, backup and disaster recovery
- Replication and consistency protocols
- Measurement, modeling, and characterization
- On-disk layout techniques
- Prediction and grouping
- Naming, organizing and finding files
- I/O scheduling
- Storage transports
- Versioning systems
- Volume management

[1] "Human administration of storage systems is a large and growing issue in modern IT infrastructures. We are exploring new storage architectures that integrate automated management functions and simplify the human administrative task. ***Self-* storage systems*** are self-configuring, self-organizing, self-tuning, self-healing, self-measuring systems of storage bricks. Borrowing organizational ideas from corporate structure and technologies from AI and control systems, self-* storage should simplify storage administration, reduce system cost, increase system robustness, and simplify system construction." [PDL])

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

We begin with a discussion of application-aware storage in the context of active storage devices. At the DS2 workshop, both application-aware storage and active devices were thrust areas. Given their similarity, the following section serves to differentiate these two topics.

Application-aware storage and active disks

Application-aware storage and active disks share the same thesis: increased application knowledge can be used to improve storage system performance. The key distinction between the two is that application-aware storage involves moving *characteristics* of the application closer to the storage whereas the utilization of active disks moves the application itself closer to the storage. These two are not mutually exclusive. The benefit of application-aware storage is being able to customize the storage for the application, whereas active disks eliminate the network delays between clients and storage devices, preserve network bandwidth through intelligent data filtering, and give applications a better view of the internals of a storage device (for optimization purposes).

Consider a simple example in the context of database systems. If the application were to make available to the storage device a mapping of the database records to the blocks being stored, along with a list of the anticipated records to be accessed, then the storage device could more easily cache and pre-fetch the appropriate data blocks. This would be an example of application-awareness. If, in addition, the application were to physically run part of the database (the query engine) on the storage device, then only the filtered contents would be returned to the user. This would be an example of an active disk.

The exact definitions we arrived at for these two thrust areas are as follows:

- Active storage devices run application-specific code to perform application-specific functions. These functions, built-in or field-programmed, exploit the processing capabilities of the storage devices to improve application performance. Examples include functions for executing database, data mining, and multimedia operations at the storage device.

- Application-aware storage devices acquire knowledge about applications, users, and other environment factors. This knowledge can then be used to improve the overall functioning of the device (e.g., more effective data layout, caching/pre-fetching strategies, intrusion detection, and QoS). Application-aware storage devices can gain knowledge through a combination of user hints, statistical accounting, and inductive logic.

Given these definitions, active storage can still benefit from additional application-awareness. Additional knowledge as to, say, the types of queries the application expects to submit (e.g., in the form of hints) would enable optimization within the device (e.g., data pre-fetching). In a sense, an active storage device is simply *closer* to the application but knows no more about the behavior of the application than does a traditional storage device. As such, any additional information with respect to the spatial/temporal access patterns of the application, or the desired goals (e.g., performance, availability, and reliability) can be exploited within an active disk.

As storage devices continue to evolve, we anticipate active disks playing a major role. In the context of application-aware storage, we see the following research opportunity:

- Programmable environments for storage devices would allow applications to transfer specific knowledge into the device. This knowledge can then be used by an active disk to optimize execution of a particular applet. The largest research questions relate to

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

establishing the proper language and mechanisms for transferring such knowledge. At the very least, knowledge will include spatial/temporal characteristics of the application, but additional information such as related files or contextual information may benefit more search-based applets.

Autonomic storage

Autonomic storage strives to automate frequent administrative tasks associated with running a storage system. Tasks include common chores such as resource provisioning and capacity planning, performance tuning and load balancing, problem diagnosis and healing, and data protection and lifecycle management. By offloading this work from IT administrators and centralized management applications, systems may become more scalable and reliable. The interface into an autonomic system from the perspective of an administrator or application will be one of the goals. Rather than manually configuring the storage for an application (the way things are done today), the storage system will self-configure for an application, and do so in accordance with the goals specified by the application (e.g., no more than 5 minutes of downtime, and all transactions must complete within 5 seconds).

Application-awareness is therefore a fundamental building block of autonomic storage, as goals are application specific. Specific research areas for application-awareness in autonomic storage systems include:

- Self-tuning devices that need to understand application access patterns.
- Self-organizing devices that need to know if they are, say, part of a RAID5 array.
- Self-securing devices that need to know what the application authorization criteria are or what accesses are considered "normal" (for anomaly detection).

There are also opportunities for categorizing both storage capabilities and application requirements by type, for performance, availability, reliability and security. Policy-based management could then map the requirements to the capabilities in an autonomic fashion. In addition to the large amount of industry and academic research already underway for the development of the autonomic algorithms necessary to accomplish this mapping, there are additional opportunities for standardization. In particular:

- The manner in which applications specify their goals/requirements and storage systems/devices specify their capabilities for performance, availability, reliability and security could be standardized, as could be the out-of-band communication mechanisms by which components of a self-* system communicate.

Archive, backup and disaster recovery

Modern day backup and disaster recovery (i.e., restoring a system's data from a backup) techniques are limited by at least two factors. First, there is the large amount of block metadata used to distinguish between free and used blocks, as well as the metadata used to describe which data blocks have changed relative to a given backup (i.e., snapshot maps). Access to such bulk metadata is typically gated by the storage application (e.g., file system) and not readily available to a device-level (i.e., physical) backup application. In other words, in order to backup a file system, a backup application must first ask the file system which blocks it is using. Typically, this is done by having the backup application simply read files from the file system as though it were a normal client, referred to as a logical backup. However, the desire is to have a more efficient physical backup that reads data directly from the devices, but this requires:

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Standardizing bulk metadata exchange between the applications and the storage system and researching the mechanisms for effective use of such metadata in backup/restore applications.

The second limiting factor in backup/restore is the notion of application consistency in the data blocks that are stored. When the storage application is in a consistent state, no outstanding transactions are in flight. For example, a database application with a transaction that was only half-completed on the storage device would not be interested in a backup of the device at that particular moment in time. Backup/restore applications should therefore only restore from backups representing a consistent view of an application. The metadata exchange alone is not sufficient for determining whether or not a collection of blocks or objects are in a consistent state relative to the storage application. Additional awareness as to higher-level application metadata structures (e.g., file system directories) would be necessary for this, and this requires:

- Researching and standardizing the mechanisms by which applications specify consistency requirements on the application data being stored.

Archival systems would also benefit from application-level knowledge, but in a different way. With respect to information lifecycle management (ILM), an archival system could readily employ application-level knowledge as to retention policies and frequency of access, thereby automating the process of moving user files between disk and tape and automatically purging expired data. In particular, we have an opportunity for:

- Standardizing the exchange of ILM-related metadata between the applications and the storage system and researching the mechanisms for effective use of such metadata in information lifecycle management.

It is our belief that the recent trends toward object-based storage devices (OSD) will greatly improve best practices in terms of archive and backup/restore. OSD addresses the metadata problem by giving the storage devices control over low-level block metadata. In addition, it provides user-settable attributes that can hold information with respect to ILM.

Replication and consistency protocols

Data replication is a powerful technique for improving the performance and/or availability of a system. However, with replication comes the cost of consistency protocols for ensuring the replicas are up-to-date. Application-specific replication policies and consistency semantics could be exploited within a storage system to improve the efficiency of replication. In particular, these areas are especially suited for research in application-aware extensions:

- Exclusive caching: application information about higher-level caching structures could be used to eliminate the effects of redundant caches.
- Data reconstruction: information about replication factors could be used in peer-to-peer storage environments to reconstruct data between storage nodes.
- Application failure: by matching application failure boundaries to that of the underlying data layout, unavailable replicas won't necessarily result in application down-time as long as enough of the right replicas are still online.
- Consistency protocols: application-specified consistency semantics could make replica management more efficient (e.g., if all replicas don't need to be immediately up-to-date, write performance could be improved).

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Measurement, modeling, and characterization

Configuring a storage system involves selecting the right kind and number of devices for a given workload, deciding how data gets distributed across the devices, and then deciding how data is organized within each device. However, even a single disk drive is a complex storage device and storage systems are even more so. Connecting together multiple devices and efficiently configuring these devices for a given workload can be a challenging task. Even more, devices can be configured for varying levels of performance, availability and reliability. Exploring this trade-off space requires more than a brute-force technique that simply tests all possible device configurations and picks the best one – the space is much to large for such an approach.

Therefore, current approaches for exploring storage configurations are either trace-based or analytical. In trace-based approaches, the I/O workload is traced (recorded) and replayed against a candidate storage configuration to see how well it would perform. The candidate configuration could either be a real collection of devices or a simulator. In analytical approaches, a workload is characterized by a few key spatial/temporal parameters (e.g., average request size, inter-arrival rate) and then input into a model of the candidate storage system; the model then outputs a value such as the expected latency of throughput for the workload. In both cases, the goal is to determine which configuration results in the desired level of performance, availability and reliability for a given workload. In effect, this allows a system to help evaluate what-if scenarios (e.g., what if another disk was added to a workload, or what if the workload was moved to a differently configured RAID array?).

Whether trace-based or analytical, there are opportunities for more effective device configuration if application goals and workload characteristics can be made known to the storage system at configuration time. As such, intelligent systems can dynamically configure devices as per the workload and desired goals of an application. Unfortunately, a big gap exists between the workload characteristics known by an application and the resulting I/O seen in the storage system. This is in part due to device sharing among applications (concurrency), but is also due to the behavior of the I/O subsystem of an application's operating system (the entity that issues I/O requests on behalf of an application).

Application-aware storage in the context of measurement, modeling and characterization therefore presents the following research opportunities:

- Determining what application-level workload characteristics are essential for effective device configuration.
- Determining how application-level workload characteristics, in the face of concurrency and operating system interactions, translate into storage/device-level workloads.
- Determining what mechanisms/heuristics are useful when translating application-level goals to requirements within the storage system (e.g., if desired performance is > x and capacity is < y, then use mirroring instead of RAID5).

On-disk layout techniques

File systems and databases employ a variety of techniques for mapping file system data and metadata structures to disk blocks. Techniques range from clustered block allocation and track-aligned extents to immediate files and embedded index-nodes. The objective is simple: organize data so as to minimize request time and maximize throughput. How to best organize data depends on the spatial/temporal access patterns of the data in question. As such, the

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

more information a storage system has on the anticipated access patterns of the data, the more effectively the data can be stored. For example, files that are write-mostly could be stored in a log-structured file system whereas read-mostly files could be aggressively replicated within the same device or across devices (to minimize read latency).

In general, file systems and other storage applications have a variety of policies to choose from when selecting an on-disk layout for a particular data object. Selecting the right policy (e.g., aggressively replicate) depends on the classification of the file (e.g., is read-mostly). Moreover, the same policy decisions made by present-day block-based storage applications will be shared by object-based storage devices of the future, as an object storage device will be responsible for selecting a policy for the on-disk layout of a data object.

Application-aware storage devices (block-based or object-based) that possess knowledge about the spatial/temporal access patterns will be able to more effectively manage the data they store. To this end, research will continue to focus on the best ways to obtain information about the workload characteristics of data objects. There are at least three research areas in terms of application-aware storage:

- Establishing the right set of application hints for specifying spatial/temporal characteristics of stored data and developing mechanisms to use this knowledge in improving the organization of data within or across devices.
- Establishing black-box techniques (e.g., semantically smart disks) for inducing I/O traffic in the storage device in order to determine the spatial/temporal characteristics of the storage data and developing mechanisms to use this knowledge in improving the organization of data within or across devices.
- Establishing machine learning techniques for inferring the spatial-temporal characteristics of the storage data and developing mechanisms to use this knowledge in improving the organization of data within or across devices.

Naming, organizing and searching

The most pressing storage problem today is one of managing the storage and reducing total cost of ownership (TCO). However, if autonomic computing comes to pass, the problem will shift from managing devices to finding useful information within a large collection of autonomic devices. The basic problem is one of information retrieval (IR), and we believe application-aware storage can and will play an important role within future IR environments for large storage systems.

Information retrieval wrestles with the fundamental trade-off between precision and recall. Queries with high precision rates yield few false positives where those with high recall yield few false negatives. Which of these options is more desirable depends on the needs of the application, and an IR-enabled storage system must be prepared to deal with both. For a storage device to play an active role in helping find useful information, it must increase the number and types of relationships between data objects. Notice that a relational database management system (RDBMS) is not sufficient because it fails to capture more complex, often "fuzzy" relationships between data objects. Users want to specify degrees of importance, and may also want to specify higher-level contexts in which a relationship holds. For example, when searching for particular images, specifying the time of year may make sense in certain cases (e.g., searching for all images related to the month of November may return images of footballs

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

and turkeys). Simple database systems will have difficulty modeling these relationships, unless the relationship was explicitly provisioned for when the database was built.

Relationships can be as simple as attribute/value pairs (i.e., RDBMS), but higher level concepts will need object-oriented techniques, perhaps modeled with a unified modeling language (UML), or constructs such as knowledge/semantic networks that can model arbitrarily complex relationships among data objects. We feel that storage systems will evolve to cover this spectrum of search capability. Bridging the gap between state-of-the-art information retrieval technology and storage systems is, in part, a simple matter of the storage system designers catching up with the latest developments in IR.

The fundamental research questions that need to be addressed include at least the following:

- How to best name and organize data objects at the user level and how to map this organization into an internal organization that facilitates searching.
- How to allow users to represent multiple, complex relationships among data objects.
- How to automatically discover complex relationships within the storage system (e.g., that a certain PowerPoint file is related to a set of files for a given project).
- How to save the context in which a file was created so that subsequent queries for a file are context-sensitive.

Each of these research areas must determine how much user intervention is sufficient for making subsequent retrieval of the information effective/possible. At the very least, the user must convey relationships of importance; these relationships can then be maintained automatically by the storage system. The challenge is in finding the primitive/sufficient set of information that must be provided by the user, and demanding no more.

As to enabling technologies, the OSD standard is an important step taken by the storage industry to raise the level of abstraction at the storage level. Modeling complex relationships between data objects, at the device level, will help distribute search capability to the devices. Object-based storage devices will serve as a useful interface for conducting IR research in next-generation storage systems.

Prediction and grouping

Predictive models for file caching and on-disk layout have been heavily researched. *Intra*-file access pattern classification learns the access pattern of blocks within a file for on-disk layout and pre-fetching. *Inter*-file classification identifies relationships between objects for whole-file caching and pre-fetching. The exploitation of any additional application knowledge (e.g., hints) that could aid in making such predictions is an active area of research and one we expect to continue. For example, file names and attributes are themselves significant hints as to the manner in which a file is going to be used. The extent to which attributes, or other application-specific knowledge, can be used to predict characteristics of an application's files is a key research question. File characteristics include at least size, lifetime, access patterns and relationships to other data files/objects. Attributes include information such as the filename. As an obvious example, files ending in ".mpg" are likely to be read sequentially. Exploiting file attributes (names, user ids, access permissions, MIME types, etc.) for the prediction of file characteristics will enable a device to predict the characteristics of and therefore tune access to its data.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

In addition, for the self-managing and self-tuning capabilities of future storage systems, the ability to automatically group related files or data objects is critical. For example, learning to group all "short-lived" files and managing them as a collection could both improve the efficiency of a storage system and also enable additional predictions about *which* new files may or may not be short-lived.

In particular, there are these specific research opportunities:

- Determining what groups matter for self-managed storage systems, where each group is managed separately according to a specified goal or policy. Groups may be defined by size (e.g., small, medium, large), lifetime (e.g., short-lived vs. long-lived) and access pattern (e.g., sequential vs. random).
- Determining which application characteristics enable a self-managed storage system to predict a file's group when the file/object is first created. As such, the self-managed storage device can then automatically bind the appropriate policy to the file.

I/O scheduling

I/O scheduling (at the host, storage system and device levels) can significantly improve the efficiency and performance of a storage system, but is also useful for distinguishing between different qualities of service or service-level agreements among competing applications. There are at least three components of I/O scheduling that will benefit from application-awareness: demand-fetched I/O, pre-fetched I/O, and caching.

Demand-fetched I/Os are the requests being synchronously requested by an application. As such, the application will often block until the request has been satisfied. However, the order in which requests are sent to the storage devices does not have to be the same as the order in which they were issued by the applications. Particularly, in the face of concurrency from multiple applications, a storage system may benefit from request ordering. As such, issuing an I/O out-of-order may either improve the efficiency of the storage device (e.g., disk scheduling) or enable the storage device to honor a specified priority or quality of service for the I/O of a particular application.

Pre-fetched I/O, on the other hand, is asynchronous in nature and will not cause an application to block. The motivation is to pre-fetch that data that would ultimately require a demand-fetch by the application. Such pre-fetching can be driven by application-level hints about future accesses or predictions made within the storage system.

As for caching, once data has been fetched, selecting a cache eviction policy can also exploit hints from the application or predictions as to which data blocks are least likely to be used.

The key opportunities in this space are at least the following:

- Researching and standardizing the mechanisms by which applications specify priorities on entire workloads or on a request-by-request basis. The challenges are to determine the right priority levels, along with the mechanisms for honoring the specified priorities throughout the entire storage system.
- Researching the algorithms by which application-level I/O priorities can be used to improve the caching and pre-fetching of data.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Storage transports

Storage transports are networks specially built for storage traffic. In the SCSI Architecture Model (SAM), transports include technologies such as parallel SCSI, Fibre Channel, InfiniBand, and Internet SCSI. Outside of SAM, TCP/IP networks used for carrying NFS traffic could also be regarded as a storage transport. In addition, networking stacks designed with storage workloads in mind (e.g., SCTP) are active areas of research and development.

Application-aware transports for storage would allow improved quality of service at the transport level and benefit storage applications by managing the trade-offs among security, performance, availability, and reliability. For example, it is often the case that increased security comes at the cost of decreased performance. As such, application-aware transports may use similar mechanisms required for application-aware I/O scheduling (e.g., priorities) to tune the transport according to the needs of the application.

The fundamental research questions to be addressed include at least the following:

- Determining the techniques for matching application-level security requirements to capabilities of the underlying transport (e.g., mapping storage security requirements to a secure network transport).
- Exploring techniques for routing storage I/O requests as per application-level scheduling priorities and/or persistency/consistency requirements (e.g., certain workloads may require synchronous remote mirroring for every I/O – a requirement that could potentially be optimized at the transport level). These issues are of most relevance in the context of a storage area network (SAN) or a storage protocol running across a wide area network (WAN).
- How to effectively trade off reliability, availability, security, and performance within the storage transport. An awareness of the application goals (e.g., requirements for performance and security) may enable the storage transport to provide better qualities of service.

Versioning systems

Storage systems that maintain multiple versions of a user's data are desirable for a number of reasons: intrusion detection, recovering from user mistakes, and recovering from system failures. However, the costs associated with versioning systems include at least the extra capacity and management required, mechanisms for purging old data, and methods for constructing consistent views of a system when multiple versions of applications data objects are available.

Application-awareness, as it relates to versioned storage, presents a number of research opportunities. In particular, there are the following:

- Determining how to use old versions of data to recover from system failures, specifically the mechanisms necessary for constructing point-in-time consistent views from versioned data objects.
- Guaranteeing consistency in the face of application concurrency and/or Byzantine faults by using versioning and determining what application knowledge is necessary to guarantee consistent updates in the face of concurrency or to recover from failed/corrupted storage nodes.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Using continuous versioning for intrusion detection and determining what application knowledge is necessary to detect an intrusion.

Volume management

Volume management is an overloaded term, but it generally involves some form of hardware virtualization in order to overcome the physical restrictions of a large collection of devices. Important features that need to be virtualized, for easier device administration and configuration, include at least the device capacity, location, and aggregation information (e.g., part of a RAID5 array). Some of the key benefits of virtualization include hot relocation of failed redundant storage, heterogeneous device support, and grow-in-place storage systems.

Volume management could benefit from application-aware storage in a number of ways. In the context of block-based virtualization, we feel that block relocation events could be facilitated if the storage system possessed knowledge of higher-level metadata structures (e.g., file extents). For example, a capability revocation to a collection of disk blocks could be triggered when the block mapping changed for a particular file. SCSI extensions for block-level metadata might be one method of relaying such information into the storage system. Another way to attack the problem is to change the interface from blocks to a higher-level construct such as objects. Object interfaces give the storage devices control over all low-level metadata structures (i.e., for block allocation) thereby enabling changes in the block mapping without interrupting service to the users of the logical volume. Somewhere between these two solutions is the notion of per-block metadata. For example, application-level checksums stored with each block could assist in the redundancy checks within a storage system.

These specific research opportunities exist for volume management:

- Researching and standardizing the metadata exchange necessary for storage systems to notify applications of block relocation events within volume managers.
- Researching and standardizing methods for maintaining per-block metadata (e.g., checksums) that could provide applications with higher levels of reliability.

In short, more application-level information about the relationship between data blocks is needed to enable many optimizations with volume managers. The application-awareness is simply one of metadata used to describe the data blocks.

Summary

In summary, the research subjects identified in this roadmap form the basis for INSIC pre-competitive collaborate research and development programs. While standards are not within INSIC's purview, they are the backbone of functional and distributed storage systems and need to be considered concurrently with research. Standards organizations such as the Storage Networking Industry Association (SNIA) and the Distributed Media Task Force (DMTF) are ideal environments for exploring standards opportunities further.

Application-aware storage will play a fundamental role in the storage industry in the years to come (say, 5-10 years), as both academia and industry continue to explore solutions to the primary challenges faced by the storage industry today. We have indicated opportunities relating to one or more of research, development, and standardization.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Autonomic storage presents opportunities in all three areas. We must develop the algorithms (policies) that exploit application knowledge to reduce management costs and improve quality of service of the storage, engineer the appropriate mechanisms for transferring application knowledge between applications and components within an autonomic storage system, and standardize the use of such policies and mechanisms so as to facilitate interoperability across the industry. Of all areas discussed, autonomic storage is perhaps the most inclusive, as it covers such a wide range of issues. From the user's perspective, such a storage system will automatically give applications a system configured for their desired goals (i.e., levels of performance, reliability, availability and security).

The opportunities for archive, backup/restore and volume management are more focused in that we need to develop the proper mechanisms for transferring application knowledge into the storage system. Such knowledge includes (at least) block extents and consistency information for these blocks. Similarly, research in replication and consistency protocols can teach storage systems how to optimize replication policies so as to respect the consistency requirements of applications. Using versioning storage for accomplishing this task introduces additional possibilities, but application knowledge as to which versions are consistent is, of course, a necessary ingredient. As to the end-user benefits, archive systems that understood application-level retention policies could automatically manage information lifecycles, application-aware backup/restore systems could operate much more efficiently and reliably relative to modern-day solutions, and volume management systems could reduce application downtime in the event of data movement.

In terms of self-configuring and self-tuning storage, the mechanisms by which data is organized within and across devices and how these disks are modeled will both benefit from information about the applications accessing and using the data. At the very least, spatial/temporal access patterns can be used to select the most appropriate organization of the data. This same information can also be used in making I/O scheduling decisions so as to optimize the order in which data is stored and retrieved. Similarly, direct application-awareness in the storage transports themselves may introduce additional scheduling/routing opportunities, in addition to providing better qualities of services (e.g., security) that are more closely mapped to the applications accessing the storage. Systems that employed such mechanism could more efficiently meet the needs of the end-users, without over-provisioning.

Finally, turning to the mechanisms by which data is named, organized and searched for in a large storage system, application knowledge is a critical ingredient in establishing the relevant set of relationships and contexts in which data will be organized. Of course, the applications will establish the initial organization (e.g., file and directories) of their data. However, the grouping of data by application-level characteristics and the prediction of such characteristics will be a function of, say, a self-organizing storage system that automatically constructs data structures (e.g., database indices) for imposing additional structure within the storage so as to optimize subsequent queries. Effectively searching through large amounts of data is a fruitful area of research and will ultimately lead to storage systems that do not just store, but also help *organize,* user data.

Additional areas not discussed at the thrust's meeting

Although not discussed in the course of the thrust's work, we feel these additional areas will also benefit from application-aware storage. We briefly outline some opportunities:

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Long-term storage: Application-specific information as to how the data was created, the format the data is stored in, or the requirements for processing the data in the future are necessary for the long-term preservation of digital content. Key research opportunities include the mechanisms by which this information is conveyed and preserved within a storage system.

- Metadata consistency protocols: Awareness of application-level consistency semantics would not only benefit backup/restore applications but could also enable storage devices to help enforce consistency in the face of concurrency or preserve consistency in the face of failures.

- Mobile storage: Information with respect to user/application connectivity, network topologies, etc., could be used by a storage system to more effectively serve mobile users/apps.

Reference

[PDL] Excerpted from the Parallel Data Lab (Carnegie Mellon University) web site. Available at http://www.pdl.cmu.edu/SelfStar/index.html as of 12/09/04.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

ACTIVE STORAGE DEVICES

Participants

Erik Riedel, Seagate Research, (Thrust Leader)
Neil Bannister, Silicon Graphics, Inc.
Deepak Ganesan, University of California, Los Angeles
Gordon Hughes, University of California, San Diego
Will Leonard, Xyratex
James Nunez, Los Alamos National Laboratory
Anand Sivasubramaniam, Pennsylvania State University
David Thiel, Hewlett-Packard

Introduction

The topic of active storage devices was one of the six thrust areas discussed at the DS2 Workshop organized by INSIC. The purpose of the workshop was to identify areas of pre-competitive research that hold promise for cooperation among industry and academic research groups. The scope of the terms "active storage" and "devices" was itself a point of discussion, so further details are given below. This report intends to convey a summary of the thrust discussion and the background materials presented.

Active storage devices

The first order of business is a definition of the terms. An ***active*** function in a storage device can be either a fixed procedure, that provides intelligence beyond today's standardized interfaces, *or* an arbitrary function, specified using general-purpose programming. A ***storage device*** may contain other storage devices, but must contain at least some non-volatile storage. As such, active memory [Oskin98, Patterson97] is not included, but active functions in individual disk drives [Keeton98, Riedel98, Archarya98, Hughes02] are, as are active functions performed in storage controllers. Activeness can exist at multiple system levels, just as block aggregation in the SNIA Shared Storage Model [SNIA04A]. Activeness can exist in storage devices, in the storage network, or in hosts; it can operate at the file/record layer, within the block layer, or at the object layer, depending on the function. However, activeness does not *have* to be done at multiple levels, as this would be an impediment to adoption of the technology. Active functions may be new functions, never done before, or a way for existing systems to be more efficient.

Active Storage Devices provide advanced functions that operate on non-volatile data either through a fixed function or through the use of general-purpose programming within the device.

Opportunity

Table 1 outlines the basic classes of potential functions that active storage devices might perform. There may not be a firm boundary between each class, but the intent is a gross classification of where active functions may be applied. Additional detail and examples in each category are given in the later section on *Areas of implementation (additional details)*.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Class	Basics	Related Areas
Indexing and searching	Applicable to many data types and structures; focus on data retrieval and organization	Database; data-aware processing
Quality of service	Matching of device performance characteristics with application performance requirements	Performance management; traffic analysis; adaptive computing
Scientific computing	Large-scale scientific computations and data sets; focus on computation using large data as input and output	Mesh computation; cluster computing; grid computing
Sensor networks	Collection of nodes, often operating together ad-hoc to create a wide-area measurement or monitoring application	Sensors. In-situ processes for data reduction before transmission to upstream nodes
Wise drives	Internal management of storage; aiding the device in its primary function or long-term storage	Reliability; long-term storage; diagnostics

Table 1: Various classes of active functions and examples of applications.

Challenges

This section outlines a set of concerns for deploying active storage devices. Each item details a concern and discusses whether existing solutions are applicable or whether research is needed.

Need for an open, standardized API (fixed function)

The problem of providing application programming interfaces (API) has largely been solved, and is already being done today. The ability to standardize a limited set of fixed functions is shown by bodies such as ANSI T10 for extensions to the SCSI interface (e.g. OSD), and the IETF for networking standards (e.g. NFSv4).

Need for an open, standardized API (general-purpose)

Portions of this problem are solved, although no comprehensive solution exists. It has been shown that a general-purpose programming environment can be deployed in limited resource environments and can be made real-time compatible. The major unknown is the internal storage device API that would be presented to downloaded functions. Related work has been done in programmable routers [Tennenhouse96] and in sensor networks. In the storage area, work on active disks includes a proposed stream-based programming model [Acharya98] that may be a viable departure point.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Isolation for data integrity and security

For read-only functions that maintain only soft state (state that does not survive failures), this can be achieved with today's technology. For functions that write data, additional protection will be required. Such protection will require a mechanism to strongly establish "ownership" of data blocks to ensure that only "authorized" blocks are written (today, data ownership is established at a very high level of the system, is not enforced below that, and is not "strong" in security terms). Note that given these limited security expectations, active functions do not significantly increase the problem (a host that is able to zero-fill and destroy a data block cannot do any worse with general-purpose computation).

Concerns about single-spindle devices

Since our definition allows active functions to exist at multiple levels of the system, masking single-spindle failures should be as straightforward as it is today. The only difficulty arises when a single data item must be split among multiple devices. If the basic unit of access and data structure can be kept small, this problem can be minimized through judicious data layout when data items are placed.

Isolating resources for shared functions

This represents a significant unsolved problem. The ability to manage resources among (potentially) competing active functions is not solved today. If the active functions could be considered soft real-time (almost all on the opportunity list can), this would significantly simplify the problem.

Coherence in a multi-device system

This problem is solved today by various flavors of host-to-host coherence protocols. One or multiple of these mechanisms can be adopted for use in the active storage environment. They may need to be system-specific, but this is not a barrier.

Performing functions with specialized hardware (fixed function)

This concern is unsolved today (how to dynamically determine the hardware/software boundary in a running system). The primary driver of such hardware would be performance considerations. The concern seems at odds with a general belief in the discussion group that a general-purpose environment could also handle any fixed function. This may be an issue of significant research possibilities, but it is not a critical path requirement for active storage.

Correctness or "reliability"

This is the second large unsolved problem. Storage devices today perform a limited set of functions and, therefore, the corner cases are well understood and well tested. In an environment of expanded function, this may no longer be true. The number of dimensions for concern will increase. A corollary to this concern is the question of who (which vendor) to blame when something *does* go wrong. As with security, above, this last concern does not seem to markedly increase with active functions. Even in today's systems, there are many layers of hardware and software from a variety of vendors.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Table 2, below, summarizes all of these challenge areas and identifies the ones that are unsolved today, as well as those which are critical for making active storage devices successful.

Concern	Unsolved?	Critical path?
Demonstration of compelling applications benefits	No	Yes
Need for an open, standardized API – fixed function	No	No
Need for an open, standardized API – general-purpose programming	**Some**	**Yes**
Isolation for data integrity and security	For apps that write	No
Concerns about single-spindle devices	No	No
Isolating resources for shared active applications	**Yes**	**Yes**
Coherence in a multi-device system	No	No
Who to blame when something goes wrong	No	No
Some "fixed" functions may require specialized hardware	Yes	No
Correctness ("reliability")	**Yes**	**Yes**

Table 2: Concerns related to the deployment of active functions in storage devices.

Research issues

The following items are the key open research issues identified by the group:

- A model of distributed computation. There is a need for a theory of how to distribute the functionality in a system around a computing environment to support reasoning about whole-system characteristics and behavior.

- Resource management at the devices. These capabilities are needed for handling multiple executing active functions at the same time. This likely needs to be an event-based soft real-time system.

- Internal device API for active functions. Establish a specification of how active functions interact with the local hardware environment in a way that is both flexible and powerful.

- Correctness/reliability/stability. Presently, in a disk drive or disk array, most of the corner cases are tested and the interface command structure is rather limited. In the new active storage system, that may not be true – there are now many more dimensions to the problem. For example, if a car "system" is redefined to include the driver, how can the same level of control as in the car alone be maintained (it is easy to impose controls, but not by also maintaining function)?

- Specialized hardware for fixed functions. This is not a critical path issue, but offers a rich field of exploration – hardware-optimized functions may prove valuable in some settings.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Debates

This section considers a number of issues that were discussed by the group, but where no definite conclusions were achieved.

Debate – fixed function vs. general-purpose

There was much discussion in the group about whether active storage functions should have fixed, limited specifications or whether a general-purpose programming environment should be provided. We agreed that active storage should encompass both types of systems and the research issues below cover both types.

One useful distinction when considering active functions is separating the *processing* of data from the *transfer* of data. For example, scatter/gather does not process data, but a search primitive would need to examine each byte of data. Error correction codes (ECC) are exactly at the overlap of the two types – examining each bit (and occasionally making corrections), but usually considered a critical part of data transfer.

Another distinction might consider four levels of increasing functionality: 1st) mapping data structures to the way they are stored, 2nd) searches done in the device, 3rd) stream operators and 4th) combination of operators.

A final possible distinction is the resource requirements – a fixed function should use fixed or bounded resources (and this is desirable in practice) while a general-purpose program could perform unbounded computation (which would be undesirable in practice).

Debate – applicability and scope

A set of questions were raised but unanswered by the thrusts members. These questions refer to the motivation and scope for active storage:

- Does Active Storage enable anything that could not be done before?
- Or is it just new packaging?
- Where are the deployment trade-offs?
- Are there better/cheaper ways to deploy the same functionality?
- What can't Active Storage do? How widely applicable it is?

Some of these issues are not inherently technical, and will be decided by the marketplace; however, it is necessary to keep them in mind when developing the technology.

One answer to these first questions is covered by the set of uses outlined in the *Areas of implementation (additional details)* section which appears later in this chapter – clearly there are many uses for active storage devices that improve on today's state of the art. One way to consider the comparison raised by the later questions is to compare the differences among 1) a server with a disk in it; 2) a disk with a server in it; and 3) a disk next to a diskless server. This may help to frame some of the answers.

An extension of this comparison is to consider how each of these packaging choices compares to a cluster of 1,000 nodes of a particular type. Does this change any of the trade-offs?

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Debate – relationship to object storage

While the use of object-based storage [ANSI04, SNIA04B] is not required for many of the active functions discussed by the group, many functions will be enhanced with this level of interface. Attempting to work only at the SCSI block level will limit some of the benefits – particularly for the functions that process the data (e.g., mesh layout or indexing or searching), but also those that attempt to provide quality of service guarantees. Additional semantic information can significantly aid in optimizing device behavior [Sivathanu03].

Debate – device-to-device communication

Most of the functions discussed do not depend on device-to-device communication. One view is that the amount of communication done between devices could be one criterion for what constitutes a good active storage function and what does not. Those functions that require a large amount of communication might best be done in a more tightly-coupled set of nodes, such as multi-processors on a low-latency interconnect, while those with communication flowing upwards to be aggregated in a host processor are appropriate for active storage. However, if such a low-latency and high bandwidth interconnect were available among the storage devices – for example with switched FC or emerging standards such as Infiniband/AS – then operations that require communication (e.g., sorting or matrix transpose) could also be done among the devices.

Areas of implementation (additional details)

This section provides additional details of the various uses for active storage functions suggested during the workshop, organized into the classes of function introduced earlier and shown in Tables 3-7, below.

Function	Fixed	General-purpose	Details	Applications
Massively parallel database search and data mining	k-closest matches; predicates (SQL)	Predicates (C++)	For example, bit-for-bit match at a record level within data “objects”; maximal substring matching; streams of data	Protein synthesis
Massive indexing and searching	Keywords	User-specific, extendable	Adding metadata – tagging of data on ingest; keyword indexing (inverted indices)	Rapid retrieval; marking metadata for ILM
Database structure	Static schemas	Dynamic or user-defined structure	Optimized layout and retrieval	Database

Table 3: Indexing and Searching.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Function	Fixed	General-purpose	Details	Applications
QoS management – bandwidth guarantees; latency guarantees; isochronous behavior	There are al-ready examples today	No	Data rate is constant (aggregate); latency to first byte; per-request statistical bound; number of streams; similar to QoS in a web server	Streamed data; multi-streams off a single disk; QoS from network through to the storage (end-to-end); high transaction rates with long queues
Allow hiding of continent-wide latency	Yes	Yes		Biologist in Chicago using resources at San Diego Super Computer
Anticipate what the workload is going to do	Yes	Yes		For example, prefetching; AI engine predicts
Aggregation/optimi-zation of requests from multiple sources	Yes	No	Scatter/gather; combined reading; "dynamic sets"	Improved system throughput by better coordination among access streams

Table 4: Quality of Service.

Function	Programming		Details	Applications
	Fixed	General-purpose		
Understand 3-d mesh of data	Data format description – e.g. HDF	This might be too specific for a vendor to provide as "fixed"	Currently, applications do contortions to put data structures into a 1-d format; this function could be generalized to a wide variety of access pattern structures – mapping them to the physical layout on storage; this function could be done with metadata that describes the data structure; many applications have a "tiled" structure that compilers can extract	Scientific modeling (e.g. atmospheric, energy, design)
Refining of mesh during computation	No	Data update based on the science (computation)	The mesh is stored in the devices; let devices update the portion of the computation space that isn't "active"	Scientific modeling
Understand life sciences data	Data formats			Life sciences; genomics

Table 5: Scientific Computing.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Function	Fixed	General-purpose	Details	Applications
End-to-end data integrity	There are already examples today	No	How does host know that what it wrote is what ended up on the platters?; hold user data in RAM and compare what is written	various ways to do it
End-to-end-to-end data integrity (host-to-host, app-to-app)	Not handled today	No	Rarely standardized; need space to store additional info; [validation on writes]; needs to be application-to-application	
Expose hardware characteristics	Fixed parameters	No	Let the host do the scheduling instead; diagnostics	
Leverage of on-disk data format	Yes	No	Data transparency – if the stored data format is known, storage devices can be swapped among systems	Allows moving of disks from system to system
Active power management	Yes	No		
Policy-based error control	Yes	No	Drive only explains where the error is, doesn't do correction; allow errors to be managed at the network level	For example, ATA streaming turns off retries

Table 6: Wise Drives.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Function	Fixed	General-purpose	Details	Applications
On-the-fly pre-allocation	Yes	Different for every file system (?)	Get better performance if layout can be done ahead of time – drive does the pre-allocation; drive does the extent map (drive could also do data movement where necessary)	Quality of the apps is decreasing, in terms of I/O tuning (programmers are not paying attention to it)
DMA straight from drive to the door	Fixed interface	no	Already available today – rDMA directly into memory (device to host) or new function (between devices); this works for FC and SCSI already, unique problem in iSCSI because networking is "not optimal"	Objects might change the way this is controlled (they make it easier to pass the right amount of information)
User-level access			This is likely not a drive function, but an HBA	
Security – pervasive security				In-drive encryption; keys stored w/ drive
Virtual incremental backup			Drive knows what has changed; can perform snapshots	
Accelerate/optimize the basic RAID controller function	Yes	Different for every RAID controller (?)		

Table 7: Other areas of application.

Concerns (additional details)

This section provides additional background to the challenges discussed in an earlier section, as shown in Table 8.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Concern	Unsolved?	Solution Exists?	Critical
Need for an open, standardized API – fixed function	No	Yes, already done today	No
Need for an open, standardized API – general-purpose	Some	No, drive-internal API unknown; yes, how to load function – limited resource environment, real-time compatible; threads/malloc/data types; examples from sensor networks use C; downloadable code on routers; needs to be prototyped	Yes
Isolation for data integrity and security	Reads are solved; unsolved for functions that do writes	Virtualization in the drive-internal API; how to give drive access to do good stuff and not bad stuff; it should not be possible for functions to physically damage the mechanism; internally, must establish ownership of blocks; the problem is simplified if functions are limited to read-only functionality; the problem is simplified if functions have only soft-state (removes recovery problem across failures); there are no security protections today, active storage doesn't make it any worse – need to separate data by user	No
Concerns about single-spindle devices	No	Since we allow multi-spindle active devices, this is not a problem	No
Isolating resources for shared active applications	Yes	How to manage resources; need virtualization of resources; to do dynamic real-time, use event-based systems today; this is a soft real-time problem; perhaps can simplify number of tasks and do less real-time	Yes
Coherence in a multi-device system	No	Not at the level we are talking about; device may interfere with host's view of coherence	No
Who to blame when things go wrong	No	Blame solution provider; need some diagnostic commands	No
Some "fixed" functions may require specialized hardware	Yes	This is an observation that contradicts the truth that general-purpose can do all the fixed functions as well as the "fixed" functions	No
Correctness ("reliability")	Yes	In a disk drive or array controller, most of the corner cases are tested. In the active system, that may not be true – there are now more dimensions to the problem. For example, how does the car deal with the driver and keep them under control (it is easy to impose controls, but not by also maintaining function); user-level requirement for stability is independent of the size of system, meaning that as a system gets larger, it needs to do *more* to be stable/reliable	Yes

Table 8: Further discussion of areas of concern for implementation of active storage devices.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Summary

Research into *active storage* (also called *intelligent storage* and *active disks*) was pursued by several academic research groups in the period 1997 to 2002. This research covered the underlying motivation [Keeton98]; prototyped some of the necessary technology [Riedel98, Acharya98]; and identified a range of applications that show compelling benefits [Acharya98, Riedel01]. The challenge for commercialization of this technology was (and is) the extent to which the technology can allow specialization for particular end-user applications while still providing a generic, standardized external interface and robust operation.

Active storage devices provide a mechanism for executing *active functions* (portions of application or application-specific code) directly on storage devices. This might be done via an enhanced but fixed interface (called *fixed functions* in this report) or through a generic programming capability (for *general-purpose functions*). Clearly, a general-purpose capability can encompass all the fixed functions, but fixed functions might exhibit higher robustness. The benefits of the active storage approach include the ability to compute close to data, pre-processing before data is sent along; the ability to take advantage of local device information; and the scalability of computation at many independent nodes, rather than centralized hosts.

The two main conclusions from the thrust group discussion on Active Storage Devices at the DS2 Workshop are that 1) there are many compelling application areas that could benefit from active storage devices and 2) most of the required underlying technology is available. The biggest research challenges are in providing an appropriate general-purpose programming interface and programming model (there are multiple potential solutions, but several constraints that must be met); providing resource management for and among the added active functions; and providing the level of assurance and reliability that customers have come to expect from commodity storage when additional functions are added.

Examples of functions in scientific computing might include handling data structures for 2-d or 3-d meshes. Similar data structure functions could also allow traditional databases to optimize their storage layout [Schindler03]. Active functions might provide indexing and searching based on fixed matching primitives or using keywords and other metadata. Active functions can enable sophisticated device-level quality of service (QoS), including performance guarantees [Eppe04]; aggregation of request streams; and workload-specific prefetching. Sensor networks also require individual devices to collect and pre-process data looking for patterns or anomalies.

Additional functions could enhance host-to-host and application-to-application data integrity with tailored error detection and correction. Functions that understand stored data can allow devices to be easily moved between systems (data transparency). Active storage can provide enhanced power management and error handling [Hughes02]. Finally, data-, application-, or system-specific security may be implemented via functions such as on-device encryption.

The main challenges for deploying this technology are the distributed programming model, the specific API's available to downloaded functions, resource management among active functions, and the robustness and reliability of the resulting system. Additional concerns focus on the integrity and security of data; the need for data coherence among devices and hosts; the possibility of using specialized hardware to assist particular functions; and concerns about the reliability and mapping of functions into single-spindle (and therefore single-failure) devices.

The active storage devices thrust group feels this is a very promising area for innovation and presents significant opportunity for both pre-competitive research and standardization.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

References

[Acharya98] Anurag Acharya, Mustafa Uysal, and Joel Saltz, “Active Disks: Programming model, algorithms and evaluation,” *ASPLOS* (1998).

[ANSI04] ANSI Project T10/1355-D Working Draft “Information technology – SCSI Object-Based Storage Device Commands (OSD),” (July 2004). Available at http://t10.org/ftp/t10/drafts/osd/osd-r10.pdf as of 12/15/2004.

[Cray04] Cray XD1 High Performance Computer, *Data Sheet* (April 2004).

[Eppe04] Eric Eppe, “Delivering Guaranteed-rate I/O to SANs in Video and Film Postproduction Facilities,” *Silicon Graphics, Inc. White Paper*, (April 2004).

[Hughes02] Gordon Hughes, “Wise Drives,” *IEEE Spectrum* (August 2002.)

[Keeton98] Kimberly Keeton, David A. Patterson and Joseph M. Hellerstein, “A Case for Intelligent Disks (IDISKs),” *SIGMOD Record* 27 (3) (August 1998).

[Oskin98] Mark Oskin, Frederic T. Chong, and Timothy Sherwood, “Active Pages: A Computation Model for Intelligent Memory,” *ISCA* (1998).

[Patterson97] David Patterson, et al., “A Case for Intelligent RAM: IRAM,” *IEEE Micro* (April 1997).

[Riedel01] Erik Riedel, et al., "Active Storage for Large-Scale Data Processing," *IEEE Computer* (June 2001).

[Riedel98] Erik Riedel, Garth A. Gibson, and Christos Faloutsos, "Active Storage For Large-Scale Data Mining and Multimedia," *VLDB*, (August 1998).

[Schindler03] Jiri Schindler, Anastassia Ailamaki, and Gregory R. Ganger “Lachesis, Robust database storage management based on device-specific performance characteristics,” *29th VLDB Conference*, Berlin (April 2003).

[Sivathanu03] Muthian Sivathanu, et al., “Semantically-Smart Disk Systems,” *FAST’03* (March/April 2003).

[SNIA04A] See the Storage Networking Industry Association web site. Available at http://www.snia.org/tech_activities/shared_storage_model/ as of 12/09/04.

[SNIA04B] SNIA OSD Technical Work Group. See http://www.snia.org/tech_activities/workgroups/osd. (2004)

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

[Tennenhouse96] David L. Tennenhouse, et al., “A Survey of Active Network Research,” *IEEE Communications Magazine* (1997).

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

LONG-TERM STORAGE

Participants

Tom Ruwart, University of Minnesota, (Thrust Co-Leader)
Giora Tarnopolsky, INSIC, (Thrust Co-Leader)
Christopher Bock, National Aeronautics and Space Administration
Randal Burns, Johns Hopkins University
James Graber, Library of Congress
Donald Petravick, Fermi National Accelerator Laboratory
Thomas Schwarz, Santa Clara University
John Webster, Data Mobility Group
Gary Zimmerman, BAE Systems

Introduction

The preservation of information existing in digital form, so that future users and future generations may access and use them as comprehensively as at the time in which the information was created, is an unsolved problem. Long-term preservation of digital assets has conceptual parallels with the archiving of traditional, physical documents. The different nature of the preserved assets, physical as compared to digital, presents great challenges. The preservation of a conventional, contemporaneously written text is met by safeguarding its physical integrity. The future comprehension of the text is assured by the immediacy of its visual impression and by the natural evolution of ordinary language. In contrast, there are no obvious solutions addressing the multiple stages of digital assets' preservation. These stages encompass the preservation of the physical medium in which coded digital assets reside, the preservation or substitution of the machinery used to retrieve the data, and the preservation of logical decoding procedures needed to represent that data in natural languages. For digital assets, safeguarding their physical integrity is only a necessary, but not sufficient, part of the solution. Due to changes in technology, the preservation of digital assets may paradoxically require that they be perennially transformed, with only the highest-level, natural-language representations being immutable over historical periods.

The rapid development of digital storage techniques in the last half a century has transformed the nature of record keeping, transmission and access as radically as the invention of movable type changed the reach of printed materials. In the course of this recent revolutionary period, the techniques available for digital data storage have evolved at a pace that essentially avoided considerations of long-term storage of the digital assets, since the machinery used to hold the records is inherently ephemeral. The initial digital era has effected the preservation of digital assets by their migration from systems of lesser capabilities to new ones of exponentially lower cost and higher functionality. The higher functionality manifested itself in higher data storage volumetric density and, to a lower measure, higher bandwidth to access the data. The volume of data migrated typically constituted a small fraction of the capacity of the new system, whose cost decreased from generation to generation. In addition, these developments occurred in the course of a period comparable in length to the duration of a human life. The presence of the creators of the records, or the presence of their immediate heirs (people who knew both the content and the techniques used for storing the assets, and had the ability to interpret and

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

manage the records) enabled the records' preservation. The "Y2K" endeavors provided a good example of the problems arising as computer systems evolve and old techniques undermine the continuity of digital record re-creation. We are now, as a community, only beginning to wrestle with the subtleties of preserving digital records for much longer than the duration of an individual's life.

The adoption of computers and their application to all fields of human endeavor, together with the availability of efficient data transport networks, has caused a high rate of digital asset generation and has stimulated demands for the availability of these assets over wide geographical areas and long periods. As the first half century of the information revolution ends, there is widespread recognition that digital assets will not be available to future generations unless their permanency is purposefully engineered into their data storage and retrieval systems.

In what follows, we will describe some of the issues of long-term preservation, refer to earlier work, and mention the existence of business opportunities driven by the obvious need to preserve digital assets. We will define some of the terms necessary to address the dichotomy between the conceptual information directly meaningful to a user and the abstract representations, in coded form, of the same information. This leads to a discussion of the tiers of preservation; namely, the physical integrity of the medium, preservation of the storage systems, and issues of logical continuity and data migration, the latter assuring that the information remains meaningful over time. A discussion of unresolved issues of costs and resource allocation precedes the enumeration of worthwhile pre-competitive research topics at the conclusion of this chapter.

Issues in long-term preservation

The preservation of digital assets needs to address, among other issues:

- Balance between the rate of asset generation and the available volumetric storage capacity and storage bandwidth.
- Limited useful lifetime of storage media, systems and devices, presently measured on the order of 10 – 20 years.
- Rapid evolution of the logical structures used to codify and retrieve the data.
- Emergence of hitherto unknown intellectual content: for instance, dynamically linked documents from various provenances or digitally animated movies.
- Automated policies for discarding useless records while perpetuating the desired ones.
- Economic and societal resources needed for the long-term preservation of assets.

Long-term digital asset preservation is the subject of initiatives in the public and not-for-profit arena, as well as in academic and industrial research. An important discussion of issues appears in the summary report [NSF & LOC] of the Workshop on Research Challenges in Digital Archiving and Long-term Preservation, April 12-13, 2002. The Workshop was hosted by The Library of Congress (LOC) and the National Science Foundation (NSF) and identified "a number of priority areas for research into new models, methodologies, and tools for digital preservation." There were 51 experts at the workshop, of whom 47 were affiliated with

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

government or academic institutions, and only four with hardware and software companies (Hewlett-Packard, IBM Research, Microsoft Research, and Sun Microsystems). Because the lifetime of the hardware and software extant to date is brief as compared to historic periods of information preservation, quite naturally there has been moderate industrial interest in the subject, except for the backup and (limited-time) archival markets. Recognition of the issues and regulatory record-retention requirements may stimulate business opportunities, already developing in the information lifecycle management (ILM) arena.

A partial sample of initiatives for long-term digital preservation in the US and abroad includes:

Electronic Records Archives Program, National Archives and Records Administration (http://www.archives.gov/electronic_records_archives/index.html)

The International Research on Permanent Authentic Records in Electronic Systems, InterPARES (http://www.interpares.org)

LOCKSS - Lots of Copies Keep Stuff Safe (http://lockss.stanford.edu)

National Digital Information Infrastructure and Preservation Program (NDIIPP) of the Library of Congress (http://www.digitalpreservation.gov)

National Partnership for Advanced Computing Infrastructure (NPACI), led by the San Diego Supercomputer Center (SDSC) (http://www.npaci.edu)

Open Archival Information System, OAIS (http://www.rlg.org/longterm/oais.html, The Research Libraries Group, Inc. (http://www.rlg.org)

Opportunities

A review of the current activity in the field suggests that the data storage systems industry has not yet developed a business model for long-term storage, as measured in multiple decades. True, there have been for many years important products for data backup and restore, and there are products now for information lifecycle management and disk-to-disk based solutions. However, these products are at the core of the unresolved issues in long-term storage: they have limited lifetimes and limited guarantees, and generally are replaced by newer generation products with a lifetime of three-to-seven years. When this replacement is effected, the data is migrated from one (often proprietary) system to another, and the solutions implemented for that migration, sophisticated as they might be, are tailored to a specific manufacturer and time period, and generally involve intense labor and address specific types of digital objects. This situation, where industry has provided satisfactory limited-period solutions, has been facilitated by the increases in storage capacity and concomitant decrease in cost of raw storage. However, the management costs associated with storage systems now vastly exceed the hardware costs. Furthermore, if the volumetric storage density growth rate slows significantly, while the data created continues to expand exponentially, the cost of augmenting storage capacity becomes large. In summary, it has been noted that there is an “absence of proven archival solutions” [Thibodeau].

The multiplicity of research and development efforts in long-term storage issues indicates the criticality of the problem and raises a red flag: how could records be preserved and be accessible for the long term (hundreds of years) if the techniques for implementing that goal present, by themselves, a long-term preservation conundrum?

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

There are instances of industrial involvement in public long-term-storage initiatives. For instance, Lockheed Martin's Transportation and Security Solutions Division and Harris Corporation's Government Communications Systems Division were granted [NARA] approximately $10M each by the National Archives on August 3, 2004, for "designing a technological solution to the challenge of preserving electronic information across space and time." The two companies will compete for a contract "…to build the Electronic Records Archives, a revolutionary system that will capture electronic information, regardless of its format, save it permanently, and make it accessible on whatever hardware or software is currently in use." In general, no commercial solutions assure today the long-term preservation of records. This is a missed opportunity in all tiers of storage, from data centers to scientific data repositories to consumer markets.

The hardware and software technologies for long-term record preservation present a special challenge for industries characterized by short product cycles, limited lifetime devices, and dynamic obsolescence of logical schemes. Research and business opportunities exist at every tier of the endeavor, from the assurance of the longevity of a bit stream to the assurance of the longevity of an intellectual record.

Nomenclature

Digital data "storage" refers to the transmission of data over time, as opposed to networking, which transmits data over distance. The specific interest of long-term preservation is defined as:

Long-term preservation assures the availability of a tangible data record, digitally stored, over periods that vastly exceed the lifetime of the physical and logical system initially used to store and retrieve the record. The underlying assumption is that the record will be available for an indefinite period.

A *tangible data record* is information that is sensorially evident to all users, visually or in natural languages, although, such as a movie, it may require some machinery for its display. It is also called the *conceptual object* [Thibodeau].

Physical preservation is merely the preservation of the storage device containing a bit stream of the object.

Bit-stream preservation is the permanency of the bit stream, the physical object [Thibodeau]. (Notice that even in a single device, the "physical object" as presented to a host is not an immutable entity. The bit stream read out is processed by error-correction algorithms that interpret the physical object for presentation, regardless of any semantic context.) The bit stream will undergo transmutations demanded by the evolution of the storage machinery. The tangible data record should be invariant with respect to these transmutations of the bit stream.

Logical preservation is the permanency of the digital objects, that is, the capacity of future generations to interact with logical objects in a meaningful way. Ideally, future users would be able to access and experience every aspect of the tangible object, whether it consists of straight text or interconnected data in a database, or digital documents extant in cyberspace with active hyperlinks, or the results of dynamic scientific calculations.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Semantic continuity assures the conversion, at some future time, of a bit stream into a tangible data record. Methods are provided for assigning meaning to a bit stream, so that it can be recognized by a machine and rendered into a conceptual object.

If semantic continuity assures the ability of recreating a tangible data record, issues arise concerning the authenticity and provenance of the result; specifically, that the tangible data record reproduced in the future is an unadulterated rendering of the original record and that it was created, in fact, by the purported source.

Tiers of preservation

Physical integrity of the storage system elements

Long-term preservation begins with the integrity of the data in digital storage systems for periods that do not pose problems of semantic continuity because the original operating systems and software are available: namely, when the data resides in a given system, in-between migrations. If the long-term preservation strategy is not based on regular and rather frequent migrations, the physical of integrity of the data is nevertheless a prerequisite for preservation.

Magnetic tape has been and is widely used for backup and recovery, as well as for archival purposes. At this time, it seems that tape will continue to play a critical role in long-term storage. Tape cartridges do not directly consume power while being off-line, and although systems of disks with managed power, as in MAID (Massive Arrays of Idle Disks) architecture, are appearing in the marketplace, there is currently insufficient data on the long-term operability of disks drives that would be kept off-line for protracted periods. Anecdotal evidence from a national laboratory, about recovering data from tapes that were kept offline for fifteen years or more, showed limited recoverability of about 70% of the records. Serious questions remain about the long-term recoverability of data from either tape or disk drives [Hughes & Murray]. It is worth considering whether the storage industry should invest in technologies that promise long-term physical preservation of the original bit stream, such as optical tapes using holes in a gold surface that could survive many physical and chemical challenges.

Bit stream preservation: storage system integrity during the period of its intended lifetime

Bit-stream preservation is most common today in RAID [Patterson, et al.] architectures. In general, we need to store multiple copies of the same bit stream explicitly or implicitly so that the bit stream survives if the storage device fails. Besides mirroring and higher forms of replication, two different strategies are possible, even though they ultimately amount to the same: First, we can use an Information Dispersal Algorithm (IDA) [Rabin], which breaks a bit stream into n chunks such that $m < n$ of the chunks are sufficient to reconstruct the original bit stream. Secondly, we can extend the RAID Level 4 and 5 technologies to include more than a single parity. In such a system, we store data in blocks, group m blocks into a reliability group, to which we add k additional *parity* blocks calculated with the use of an Erasure Correcting Code (ErCC) (a variant of an error correcting code). We can then reconstruct the contents of all blocks in the reliability group from any m out of the $n = m + k$ blocks. Both systems provide the same functionality [Preparata], but the latter allows direct access to any data object.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

When data resides in on-line or near-line disk-drive-based storage systems utilizing RAID [Patterson, et al.] architecture, the failure of any given disk drive does not compromise data integrity if the disk drive is replaced and the bit streams reconstructed in the replacement disk drive using the redundant information extant in the RAID [Katz, et al.]. For the disk drives in use today, with multi-hundred gigabyte capacity, the required time for data reconstruction (mean time to repair, or MTTR) can be long enough that the probability of a second, and hence catastrophic, failure within the same array, while data reconstruction is in progress, is significant. Field experience as well as research [Xin, et al.] shows that, for large-scale storage arrays built from thousands of storage devices, RAID is insufficient to guarantee data integrity.

There are concerns about the rate at which data corruption can occur in disk-based systems and the limited number of discovery and recovery tools that exist today to resolve these problems. Problems can occur, among other reasons, because of:

- user errors (e.g., accidentally deleting a file or many files)
- operator errors (e.g., accidentally initializing a disk, or rezoning a disk incorrectly)
- malicious activity (e.g., viruses)
- lack of storage system reliability and integrity (both hardware and software)

In addition, disk-based solutions are generally proprietary to their vendors, in that file systems are not transportable. We believe that efforts need to be placed in developing file systems and object-oriented storage paradigms that provide solutions to these problems in disk-based long-term archival systems.

A long-lived bit stream can outlive the life of any storage device on which it resides. Because the bit stream presents a coded view of the user information, coded according to ad-hoc conventions, the preservation of the bit stream is insufficient to assure the preservation of the tangible document.

Semantic continuity: logical integrity of the data storage system

Semantic continuity refers to the ability to interpret a bit stream in order to recreate precisely the tangible record from a long-ago past. We illustrate this with a simple example. Different word processors internally store the “same” document in quite different bit streams. The sameness results from abstraction. As users, we are interested in the properties of the document as a document. That is, for us, the document stays the same whether we print it, store it on removable media, or translate it into a .pdf file.

The bit stream – the physical object – incarnates at a lower level the original tangible record, but may bear no physical resemblance to the bit stream first used to preserve the tangible record if the preservation techniques rely on successive migrations to new hardware and software. Our understanding of the relationship between logical and physical objects and of the relationship between physical objects representing the same logical object is of considerable research importance. These are new problems that we (as a society) have never had to deal with.

Changes in technology result in new and incompatible ways in which the same type of logical object will be stored. One way to achieve logical preservation – that is, continuing capacity to

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

interact with a document in a meaningful way – is semantic continuity, i.e., preserving our capacity to interact with the logical object based on the same bit stream.

One alluring solution to the problem of semantic continuity is the concept of emulation [Rothenberg]. Under this scenario, the original bit stream would be preserved intact, and when the decay of the original physical medium requires that the bit stream be migrated, the object of preservation is an identical replica of the *ab initio* bit stream. Together with that record, the original software that created the object is also preserved. The future re-creation of the tangible record consists of running the original software in a powerful future computer that will emulate the performance of the machine that created the record: that is, a machine that will be able to mimic every machine step that resulted in the creation of the *ab initio* bit stream and its conversion into a tangible record. It may be possible to demonstrate in principle that this solution is feasible. However, its practical implementation is complex [Rothenberg], and presents a recursive problem tantamount to the original problem it comes to solve: since the computers and storage media of the future will unavoidably have limited lifetimes, each introduction of new machines will necessitate either the re-creation of emulators, or the emulation (with new tools) of past emulators. Furthermore, the complexity of operating systems and software packages extant in 2004 has resulted in computer systems that are not fully understood and are remarkably prone to error. It is therefore not clear that, pragmatically, emulation is indeed an executable strategy [Bearman].

An alternative proposal distinguishes between the archiving of the bit streams and of the codes required to interpret them [Lorie]. This solution, dubbed the Universal Virtual Computer (UVC), requires much less than the ability of fully emulating a computer and data storage system. Here, the path to future preservation starts with the encapsulation of the data, at the time it is first created, with metadata sufficiently descriptive of itself and of the digital record to be preserved. This scheme relies on preserving, at the time the data is originally stored, a simplified but sufficient set of instructions for decoding that would be interpretable in the future by a UVC.

Migration

The default procedure that today enables the re-creation of older tangible objects is migration. Migration addresses both the limited lifetime of the physical machinery and the ephemeral nature of the logical machinery by regularly re-recording bit streams into new storage systems. The process is usually accompanied by a translation, whereby the tangible object is accessed under the older, but still available, application and operating system, and by creating a new bit stream using the newer application and operating system.

Migration has worked well, notably because the conservation period has been short, about half a century, and because during those fifty years the volumetric density of storage has increased exponentially and has been accompanied by decreasing hardware costs and vastly improved processing power. Migration is a labor-intensive process, and its reliability for records that are rarely accessed is uncertain. Migration also tends to be highly selective of the records being kept by any given institution and, in general, migration evolves records from one proprietary system to another one that is similarly proprietary. The selectivity of the migration process may be desirable if it reflects curatorial judgment, or undesirable if stimulated, for instance, by the obsolescence of means of interpreting the old data.

While far from a universal solution, migration is a method that demonstrably works for limited amounts of data. Among its critical drawbacks is that the effort required to migrate data leads

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

easily to discarding data that has been created in ad-hoc format; for instance, the data created in the course of the modern massive experimentation, simulation, and modeling in the sciences. The value of data has to be asserted essentially at the time of its creation, and curatorial discipline, at high cost, established from that moment onwards. Migration has not yet confronted the possibility that exponential increases in volumetric storage capacity and computational processing may achieve a plateau such that the data to be preserved would need to be migrated to devices of capacity and performance similar to that of those being retired. As the rate of information generation continues to grow, this possibility would impose a severe burden on the time and resources required to effect the migration. If storage volumetric capacity increases faster than bandwidth, then migration will take up a large portion of the total bandwidth of a storage device. For instance, a 10 PB storage device with a sustained bandwidth of 100 MB/sec takes on the order of 30 months to fill with archival data and 30 months to unload of archival data. Steve Louis, Lawrence Livermore National Lab, reported an aggregate storage bandwidth of 1,037 MB/sec achieved in a highly specialized system, not built for commercial purposes. With that sustained storage bandwidth, the migration of 10 PB of data would still take about four months [Louis].

Migration at a lower level preserves bit streams. At a higher level, migration steps can preserve logical objects. For example, a company can access archival documents, use a translator that transforms the documents into ones stored in the current word processing format, and then store them again. Logical preservation is a very active endeavor.

Cost and social constraints

Important policy decisions about long-term digital preservation are likely to be driven by costs. The cost considerations apply to all users, governments, corporations, and individuals. For the data storage systems industry, cost models are essential tools for understanding the attractive market opportunities that would justify R&D efforts. Traditionally, data centers have absorbed the cost of record retention as part of their business overhead, since financial, actuarial, and legal records have very high value for the continuity of the business. Small offices and home offices, including the preservation of digital images, are beginning to confront the complexity of the issues associated with record retention and accessibility for long times.

Current issues with cost include the total cost of ownership (TCO) required to hold a collection of data over a period of time. Issues that arise include:

- What is the size of the market opportunity for the long-term archiving of digital collections?
- What is a sufficient value proposition to ensure that collections are kept in perpetuity?
- What is the “value” of the data being stored relative to the TCO?

During this particular period of evolutionary changes in storage hardware and software, it seems that the TCO will be relatively high due to frequent technology refresh cycles. To compound the problem, the amount of data worth archiving is being generated at an increasing rate. A study [Chapman] has shown the counterintuitive result that, for high-resolution photographic data, digital preservation may be more costly than preserving the original physical specimen. In this study, the costs analyzed did not even include migration and curatorial efforts, but simply the preservation and accessibility of the data.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

The long-term preservation of records presents choices to be made by society and by technology organizations. Topical examples of societal choice are the Sarbanes-Oxley Act of 2002 and the HIPAA regulations relative to medical record retention [Sarbanes-Oxley]. These directives have generated opportunities for data storage systems and information lifecycle management. The industry has responded with solutions that are proprietary to their creators, and that have not been tested by time. If long-term preservation tools proliferate along proprietary channels, the issue of preservation will be compounded by the need to preserve the preservation tools themselves. These tools are, without exception, supported by hardware and operating systems with a lifetime not longer than approximately ten years.

Some of the other issues involve the survivability of collections that transcend the lifetime of the organization(s) that are charged with maintaining the collections. For example, what happens to a collection when the organization responsible for it goes out of business? Is there, or should there be, a federal or world-based organization that is funded to absorb these "orphaned" collections so they are not lost? If so, how does such an organization also provide for access to the newly acquired collections? This can also involve absorbing and maintaining access constraints for legal and regulatory purposes.

This one issue overlaps many other areas including:

- Security – how to maintain access restrictions and modify them when necessary. For example, at some point a piece of a collection may become public at which time the security restrictions are relaxed. Furthermore, how does one track copies (i.e., backups) such that they do not inadvertently fall out of their security realm?

- Privacy – some collections may contain information that should not be released or should be available to only a select few. Access restrictions apply but there is also the problem of tracking all known copies of the data such that access to them can be controlled and the data can be completely erased if necessary.

- Regulatory Compliance – federal regulations place a retention period on certain kinds of data. How is compliance managed and enforced? Also, after the retention period for a piece of data has expired, how can one be certain that it has been removed, in fact, from all recognizable media? This overlaps with both privacy and security.

These issues must be well understood and addressed by each and every organization that inherits responsibility for a collection.

Desirable technology scenarios

Physical Machinery

Since the inception of the digital age, long-term data preservation has been effected by sequential migration of data from an obsolescent platform to a newer one possessing higher functionality. The storage hardware has progressed at such a rapid pace that invariably the newer platform has provided larger storage volumetric density, larger capacity, faster access bandwidth, and lower storage cost per byte. It has been noted, though, that the storage cost per byte does not directly relate to the total cost of preserving the data record [Chapman], the latter being significantly higher than the former.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

The most significant elements in this process have been the finite and rather short life of the equipment and the exponential decrease of storage costs. To date, the rapid advances of digital storage technologies (i.e., optical disk, magnetic tape, and magnetic hard disk) have determined an absolute absence of return on investment for manufacturing long-lived storage devices. An expensive, long-lived device would invariably be superseded by a device of vastly enhanced functionality, and any remnant lifetime wasted. When volumetric storage densities cease to grow exponentially, trade-offs of a different nature will arise between the relative burdens of building long-lived, costlier storage devices and less expensive, short-lived devices of about the same volumetric storage density. To date, there has been a strong motivation for the migration of bit streams, since this process also accomplished enhanced functionality and, especially, reduced cost. As storage technologies mature, however, they may cease to offer exponential decreases in the cost of storing the bit streams; and, at a relatively constant volumetric density and bandwidth, there would be no economic incentive for migrating data, except for the need to preserve it for future use or future generations.

The total cost of ownership of digital repositories is thus a significant unknown, which encompasses technology progression and curator's costs, leading to the re-creation of the data record. At this time, it is easier to estimate the cost of preserving a printed book for the next 500 years than the cost of preserving the digital typesetting records upon which the book was manufactured. Technological advances in other industries, for instance commercial aircraft, have made long-lived machines cost effective and advantageously amortized over long periods. Such a scenario would represent a radical change for the data storage systems and devices industry.

Logical Machinery

As discussed, the issue of semantic continuity, to date, has been addressed by migration or by discarding the digital asset. The rate of development of operating systems and of storage management software does not show signs of abating, and there is strong contention between proprietary and open-software systems. The multiplicity of applications and operating systems leads to the consideration, at least in principle, of solutions such as the Universal Virtual Computer or emulation schemes.

Research questions

The issues of long-term preservation present significant business opportunities. The relatively recent applications for information lifecycle management and enterprise content management, now running in the data center, are precursors to the widespread application of preservation techniques, much as computers evolved from computer-center mainframes to computers in every household. The goal of data preservation is the ability to completely reproduce the original tangible record at some undefined future time. Many issues need to be addressed to accomplish this long-term digital preservation. The degree of difficulty and uncertainty as to how to accomplish long-term preservation goals makes this field appropriate for pre-competitive research efforts. Topics that are worthy of attention include:

- The longevity of the physical bit streams on various archival media and storage mechanisms.
- Integrity of data bit streams in large arrays of storage devices or libraries.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Diagnostic means for data integrity, early enough to avoid failure. Self-diagnosis and self-repairing data systems.

- Hardware and software strategies for constant, or only slowly increasing, volumetric densities. How feasible does migration become if the volumetric storage capacity ceases to grow?

- What is the optimal storage architecture for long-term preservation? Is the current spectrum of off-line tape to near-line to on-line storage a valid scheme for indefinite preservation? What should be the distribution of storage tiers?

- Is migration a reliable preservation technique, when the original bit streams and applications cease to exist in the migration-target system? What are the risks of multiple successive migrations?

- Preservation strategies. Is emulation technologically and economically feasible? How would a Universal Virtual Computer be developed, and what changes in procedures for creating digital objects would this approach demand?

- How should operating systems be documented to enable emulation? Is emulation of the hardware a better approach? Will emulation lead to a recursive long-term preservation cycle, with the newer generation machines being required to emulate the emulators running in older-generation systems?

- Curatorial strategies. What techniques will assure accessing specific records at some indefinite time in the future? What metadata should be automatically created with each new digital object to enable this process? How is the authenticity and provenance of digital records to be assured in the future?

- What are the costs of long-term preservation? Can a long-term preservation utility be created and run profitably?

- How is the preservation of non-traditional documents (active documents, dynamic simulation results, computer animations, algorithm-dependent records) going to be accomplished?

- What are the limits of long-term preservation? What is the volume of data that can be preserved for any finite investment in the preservation infrastructure?

Summary

The preservation of digital records for indefinite periods has obvious value in the financial, commercial, legal, governmental and human services arenas. It has also a high perceived value for preserving the cultural heritage from generation to generation. The prevalence of digital data creation and the need for accessing it have already stimulated the creation of data storage systems that, for relatively limited intervals, preserve digital records. The issues confronting true indefinite digital preservation are largely unresolved and represent a significant business opportunity and a fertile field for collaborative, pre-competitive research.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

References

[Bearman] David Bearman, "Reality and Chimeras in the Preservation of Electronic Records," *D-Lib Magazine*, **5**, 4 (April 1999). Available at http://www.dlib.org/dlib/april99/bearman/04/bearman.html as of 11/01/04.

[Chapman] Stephen Chapman, "Counting the costs of digital preservation: Is repository storage affordable?," *Journal of Digital Information* (May 2003). Available at http://jodi.ecs.soton.ac.uk/Articles/v04/i02/Chapman/ as of 07/14/04.

[Hughes & Murray] Gordon F. Hughes and Joseph F. Murray, "Reliability and Security of RAID Storage Systems and D2D Archives Using SATA Disk Drives," to be published in *ACM Transactions on Storage*, (December 2004).

[Katz, et al.] Randy H. Katz, Garth Gibson, and David A. Patterson, "Disk system architecture for high performance computing," *Proceedings of the IEEE*, **78**, 2, pp. 1842-1858 (1990).

[Lorie] Raymond A. Lorie, "Long Term Preservation of Digital Information," *Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries* (JCDL '01), pp. 346, 352 (June 2001).

[Louis] Steve Louis, "NNSA Advanced Simulation and Computing: Past, present, future," *presented at the THIC meeting*, San Jose, CA, (March 9-10, 2004). Available at http://www.thic.org/pdf/Mar04/llnl.slouis.040310.pdf as of 12/06/04.

[NARA] NARA Press Release (August 3, 2004). Available at http://www.archives.gov/media_desk/press_releases/nr04-74.html as of 10/29/2004.

[NSF & LOC] "It's about time: Research challenges in digital archiving and long-term preservation," *The National Science Foundation and the Library of Congress* (August 2003). Available at www.digitalpreservation.gov/repor/HSF_LC_Final_Report.pdf as of 7/09/2004.

[Patterson, et al.] David A. Patterson, Garth Gibson, and Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 109-116 (1988).

[Preparata] Franco P. Preparata, "Holographic dispersal and recovery of information," *IEEE Transactions on Information Theory*, **35**, 5, pp. 1123-1124 (September 1989).

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

[Rabin] Michael O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," *Journal of the ACM*, **36**, 2, pp. 335-348 (1989).

[Rothenberg] Jeff Rothenberg, "Avoiding Technological Quicksand: Finding a viable technical foundation for digital preservation," *Report to the Council on Library Information Resources*, (January 1999). Available at http://www.clir.org/pubs/abstract/pub77.html as of 11/01/04.

[Sarbanes-Oxley] Sarbanes-Oxley Act of 2002, enacted into law on July 30, 2002. Health Insurance and Portability and Accountability Act of 1996. Both pieces of legislation require the retention of electronic records for decades.

[Thibodeau] Kenneth Thibodeau, "Persistent object preservation for archives," (February 23, 2000), http://www.sdsc.edu/NHPRC/Pubs/Meeting1/thibodeau2.ppt as of 10/28/2004.

[Xin, et al.] Qin Xin, et al., "Reliability Mechanism for Very Large Storage Systems," *Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies*, pp. 146-156 (2003).

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

PERVASIVE STORAGE

Participants

Craig Harmer, VERITAS Software Corporation (Thrust Co-Leader)
Paul Massiglia, VERITAS Software Corporation (Thrust Co-Leader)
Larry Krantz, EMC Corporation
Reagan Moore, San Diego Supercomputing Center
Hans van Rietschote, VERITAS Software Corporation
Paul Siegel, University of California, San Diego
Boleslaw Szymanski, Rensselaer Polytechnic Institute
Hu Yoshida, Hitachi Data Systems

per·va·sive** (pər-vāˈsĭv, -zĭv) **adj.** Having the quality or tendency to pervade or permeate: the pervasive odor of garlic. [From Latin* **pervāsus, past participle of* **pervādere***, to pervade. See PERVADE.]* ***- per·vaˈsive·ly** **adv.** - **per·vaˈsive·ness** **n.***

— The American Heritage Dictionary

Introduction

Pervasive storage is the subject of a *thrust*, or discussion group, one of six such groups that deliberated as part of the DS2 Workshop organized by INSIC during the period from April 27 through April 29, 2004. The purpose of the workshop was to reach consensus among representatives of industry, the academic community, and the U.S. federal government on the most important topics for pre-competitive (collaborative) storage system-related research over the coming decade. INSIC's intention is to use the results of these six discussion groups to motivate and procure funding for research into what are thought to be the most potentially fruitful topics, from the point of view of long-term impact on the storage system industry and market.

Pervasive Storage defined

The pervasive storage discussion group quickly concluded that members did not have a clear common understanding of what was meant by the term *pervasive storage*, and thus the first topic tackled was to arrive at one, to bound the discussion space. The group's consensus was that even today, storage is cheap enough that it *could* be pervasive, and in some situations it is (it was estimated that the workshop's 62 attendees' laptop computers contained a total of about 3TB of storage, of which perhaps half or more was available for disposition). The interesting research questions are not, therefore, about making smaller or cheaper devices - these will happen in the natural course of device technology evolution. Potential research topics lie in whether the (possibly hundreds of millions of) devices that will collectively comprise pervasive

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

digital storage in the future can be integrated into federated or otherwise unified systems that could provide services not available in today's storage environment. This led to the defining description of pervasive storage in the group's report-out presentation:

Pervasive storage is about pervasive access to information which must support "disconnected" as well as connected operation.

The mention of disconnectedness in the defining description reflects the group's recognition that while integrating large numbers of storage devices into a kind of 'storage web' relies heavily on much more pervasive connectivity of clients to storage that is remote from them than is common today, throughout the foreseeable future there are clearly meaningful cases in which clients are not connected to remote storage for significant intervals (hours to days). The group did not consider these scenarios to any great degree due to lack of time, but recognized that they are worthy of further discussion as a possible field of future research.

Terminology

It also became clear early in the discussion that the group did not completely share terminology. After some discussion, the following fundamental definitions were agreed upon for use within the group's discussions:

> The term ***storage*** denotes what are more commonly called virtual disks or object storage devices; that is, receptacles in which digital data and metadata can be stored directly and from which it can be non-destructively retrieved, also directly.
>
> The term ***data*** denotes what is more commonly called the data portion of a file or data object, that is, an ordered string of bytes or bits that is treated as an entity by applications and data management functions.
>
> The term ***metadata*** denotes contextual information about data. Each piece of data is associated with certain metadata that describes the data as an object, policies associated with managing it, or how applications make use of it.
>
> The term ***information object*** denotes what is more commonly called a data object; that is, the combination of data and its associated metadata. The group defined this term primarily to assert that any useful system based on pervasive storage must have some failure-proof way of maintaining a lifetime association between each piece of data and its associated metadata.

The metadata discussion was more complex than might be imagined, because metadata has historically been treated differently depending upon the nature of the information it conveys.

> Metadata conveying ***information that is independent of data's content*** is typically managed by file systems. Such metadata includes ownership, significant times in the data's history (creation time, time of last access, etc.), size, location, type (for associating data with applications that process it), and so forth.
>
> Metadata conveying ***information about policies to be applied to data*** has typically been managed by storage applications like backup and hierarchical storage management (HSM), or directly by storage administrators in some ad hoc fashion (e.g., by formatting file systems on mirrored or RAID volumes). Historically, there has not

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

been a consistent storage location for this type of metadata. Some of it is stored by file systems (e.g., significant times co-opted by HSM applications); some is stored in separate databases (e.g., backup schedules and catalogs), and some is stored in run books or administrators' heads (e.g., associations between file systems and the types of storage devices on which they should be formatted). Clearly, for an integrated system based on pervasive storage to be workable, this class of metadata must be regularized and stored in a universally consistent fashion.

Metadata conveying ***information about how an application or business function should interpret data*** has historically been application-specific, and has typically been stored as part of the data itself. Such metadata include, for example, the audio characteristics and provenance of MP3 files, the schema of a file or database, and so forth. Clearly, this metadata, too, must be regularized and extracted from the data it describes for a system of pervasive storage to be useful. The bright spot for this type of metadata is the recent emergence of XML and a number of taxonomies based on it (e.g., XBRL, SWIFT,…) that provide application-related descriptions of the business content of data external to the data itself.

The group's discussion of metadata was lengthier than might be imagined, and in essence pointed up the need for a much more far-sighted and comprehensive look at the nature of metadata and the question of how it should be stored, managed, and interpreted in order for computing to move from a close association between information objects and the virtual storage devices that contain them to an environment in which information objects exist independently of where they happen to be stored at any given moment.

Assumptions

The group's discussions were predicated upon two assumptions:

Storage will be cheap enough and easy enough to acquire and use for it to be pervasive. To some extent, this is an acknowledgment that today's storage cost and availability situation is expected to carry forward into the future. The group's discussions were implicitly predicated upon an environment in which personal appliances could be equipped with non-trivial amounts (tens of gigabytes) of persistent storage, and in which raw storage would be so cheap that storing multiple (2-10) online replicas of even low value data objects like video and music tracks or personal photographs could become the norm rather than the exception. The consequence of this assumption, of course, is that pervasive storage is physically and economically possible, so the interesting research questions are about how it could be integrated and used to deliver capabilities that are not available today. Thus, the group spent the bulk of its time discussing what could be done with pervasive storage if it existed rather than whether it could exist.

It very quickly became apparent to the group that in order for pervasive storage to significantly change the way digital data (particularly data that is primarily used directly by individuals) is processed and managed, nearly-continuous network access at bandwidths adequate for transmitting the data objects of the future (A/V clips, structured objects, etc.) will be a requirement. The group's discussions about how pervasive storage might change the nature of digital data processing were predicated on this assumption.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Belatedly, the group recognized that it had not dealt adequately with the reality that completely continuous connectivity is probably not a practical assumption, and that any evolution in the way data is processed and managed will have to accommodate the possibility that individuals, and even data centers, can be disconnected from the Internet for hours or even days. The importance of disconnected operation, capabilities that can be provided to disconnected users and systems, as well as inherent constraints on functionality in a world of integrated pervasive storage, are clearly topics for future pervasive storage discussion groups.

System configuration scenarios

To formulate meaningful research questions about applications for integrated pervasive storage, it is necessary to speculate about how it might be used. How pervasive storage might be used is in turn predicated upon how it might be deployed. The group postulated two possible scenarios for deployment of pervasive storage that it called "storage cells" and "interconnected storage farms."

> The ***storage cells*** envisioned by the group are very analogous to wireless telephony cells in that they would form a grid covering a wide geographic area. Users of storage cells would be passed from cell to cell as they moved about within the grid, much in the way that a mobile telephone conversation is passed from cell to cell as a handset moves about today. Practically speaking, the typical "last mile" between cell and client would be wireless. Storage cells would essentially serve as a cache for data, delivering it between "home" locations and end-point clients on demand. While the group did not discuss the business model for storage cells extensively, it is clear that they would have to exist on a nationwide scale or greater to be practical. Thus, established telecommunication companies are more likely to provide any such service than the recently failed generation of startup SSPs and ASPs.
>
> From a user's standpoint, storage cells have three functions: 1) they make the latency of accessing data (approximately) independent of the client's distance from the data; 2) they reduce the amount of storage required by the client, and 3) they foster "single-instance storage" so that clients are much more likely to be dealing with up-to-date data than they are in today's environment.
>
> The ***interconnected storage farms*** envisioned by the group are "home" locations for data. These storage farms would be distributed persistent repositories for data, owned and operated by large organizations and institutions, as well as by service providers. The concept of storage farms is inherent in the abovementioned storage cell model; the difference between the two models lies in the presence or absence of the caching layer formed by the storage cells. In the pure storage farm model, clients would communicate directly with the home locations of their information objects, using either an LDAP-like scheme or wireless phone number encoding to identify each object's home location. In a sense, the storage cell model is a hedge strategy against latency and the cost of connectivity. If long-distance high-performance digital connectivity becomes ubiquitous and cheap, and applications and users can evolve to deal with long-distance latencies when accessing data, then storage cells may be superfluous. If, however, either of these conditions does not materialize, there may be a genuine role for the storage cell concept.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Usage scenarios

Armed with these two hypothetical models for how online storage might be configured and used differently if the storage devices that already pervade the environment today could be integrated into a cohesive whole, the group then tried to classify clients for pervasive storage in terms of typical usage scenarios in order to assess the benefits that different types of users would derive. Four classes of users with unique usage potential and unique benefit potential were identified:

Consumers — individuals acting on their own behalf. Today, these are largely (but not entirely) hobbyists who use storage primarily for entertainment, in the form of graphical and A/V clips, games, and so forth. In the future, one can imagine this type of user becoming more digitally dependent, for example digitally storing financial, household inventory, product warranty, and other personal records on which they place some significant level of reliance.

Corporate citizens — these are also individuals, but acting in their capacity as knowledge workers in the employ of enterprises or institutions. These individuals form a different user class because by and large they operate on data that belongs to their employers rather than to themselves. Moreover, inasmuch as the financial consequences of their ability (or inability) to access data are likely to be greater, their quality of service requirements from pervasive storage are likely to be more demanding than those of consumers.

Sensor networks — as mankind places more and more stress on the planet, it becomes increasingly important to "know what's going on" in a wide variety of cases, from air quality in major cities to security in public places. Technology provides the ability to sense an impressive variety of environmental conditions, from people in motion to temperature and humidity, to air quality, and so forth. Typically, these sensors must be coupled to a pre-configured network so that they can deliver data to a computer system that can store, filter, and process it. With pervasive storage as described in the preceding section, sensor placement would become significantly more flexible; sensors could simply connect to storage cells (or to wireless communication hubs in the pure storage farm model) and deliver their data. They could be placed literally anywhere within range of a cell or hub, and would require little or no storage of their own. Moreover, in the storage cell model, it is conceivable that a cell will contain active storage devices. In that case, data filtering could be performed at the source sensor, thereby significantly reducing the long-range bandwidth and home-location storage requirements for sensor-based data.

Data centers — these are how important enterprise data is stored and processed today (there was a general consensus among the group that while desktop and laptop computers may hold data that is important to the enterprises that own them, such data is not generally treated as important, in the sense that it is seldom backed up or synchronized with master copies in a "home" location). While pervasive storage would probably represent less of a change for this class of user than for the three abovementioned classes, there are nonetheless some potential benefits, such as dynamic short-term capacity expansion using a communal pool of raw storage, more transparent data replication and recovery from disasters, and reduced latency for distributed enterprises accessing remotely stored data.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

The following sections discuss usage scenarios and potential benefits of the integrated pervasive storage configurations discussed above for these four client types.

Consumers

The primary consumer value proposition of the pervasive storage configurations outlined in preceding sections is that it enables more functional personal appliances with access to more data and effectively unlimited storage, with longer battery life and at no increase in bulk or weight. Thus, for example, one can imagine an audio or video player which, instead of holding program content locally on a small hard drive as is common today, would receive streamed content from a storage cell (or directly from a content provider in the pure storage farm scenario). Not only would the appliance be smaller and lighter, but the content selection would effectively be infinite, rather than what could be downloaded to the consumer drive of the day. Instead of having to accommodate an HDD, personal appliance physical design would be driven by human interaction (input and display) and battery considerations. In principle, this should allow for more data-intensive functions in smaller and lighter multi-purpose appliances (or for very small, light, and low-power single purpose appliances, such as an MP3 player the size of a small cell phone with access to one's entire music collection).

As another application, imagine a video recorder which, instead of storing data on a tape, streamed it directly to a storage cell, which would relay it to the owner's "home" location. One can envision modes of operation in which the storage cell would downsample the input on demand, to economize on permanent storage and bandwidth (and therefore presumably cost to the consumer) before relaying it.

Carrying this principle further, the group discussed the possibility of "storage collectives," following the Ocean Store model. In this model, people with appliances and computers could contract to make a given amount of storage available to a "storage grid." Storage on the grid would be available to any subscriber on demand for short-term data retention. Presumably, this would reduce peak capacity and bandwidth requirements for storage cells – capacity in storage collective nodes would serve as an overflow buffer. Clearly, for this "storage collective" model to work, the entire process would have to be transparent to the average consumer (with the obvious exception of chargeback).

Corporate citizens

Corporate citizens (employees of enterprises or institutions) are like consumers in that their interactions with digital data are highly likely to be increasingly "unwired" – performed when they are not physically connected to enterprise data. They are unlike consumers in that the data they access is highly likely to be held in corporate databases rather than in individuals' personal stores or in the stores of large-scale content providers.

With access to pervasive storage delivered over a wireless network, intermittently connected knowledge workers ("road warriors" as well as job-site and other outdoor workers) and their employers could have instant access to up-to-date corporate data which they manipulate using smaller, lighter intelligent graphical terminals that resemble personal computers. Benefits to the worker would include:

> ***Less to carry*** – without storage, appliances can be smaller and lighter. As with consumers, an alternative might be more functional appliances of a size and weight comparable to today's devices.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Up-to-date access to any enterprise data and applications – no need to spend time determining what data will be required while disconnected and downloading it. No guessing wrong and missing opportunities because the right data is not available.

Longer battery life – translating either into less battery bulk to carry, more functional appliances for a given weight, or longer intervals between recharges.

As data becomes increasingly important to enterprises and institutions, equipping workers with "thin client" workstations would benefit enterprises as well:

Lower device cost – the cost of an HDD or other storage could be replaced by the (presumably lower) cost of wireless network electronics.

Better control over enterprise information – with no persistent storage at the client, loss or theft of a mobile device would be much less of a concern. Moreover, data would be less apt to "walk out the door" with disgruntled or negligent employees.

Better employee performance – knowledge workers can do a better job with up-to-date data.

As Remzi Arpaci-Dusseau noted during the discussion group's report-out at the conference, both of these mobile usage scenarios run counter to the dominant trend in pervasive storage today – that of equipping personal appliances with increasing amounts of persistent storage. The counterargument, of course, is that more local storage engenders more redundant copies of data (e.g., with entertainment content), more obsolete copies of data (e.g., with enterprise data), and more time spent by users (generally not computer specialists) managing local copies of content. Moreover, local copies of data on small, portable (susceptible to theft or loss) devices inherently put more data at risk of unauthorized use. Many factors, including how pervasive wireless connectivity develops in the future, will determine which solution (local data or pervasive access to managed pervasive storage) ultimately wins out. It does seem, however, that the potential benefits of integrated pervasive storage are sufficient to justify research on the implementation problems to which the technology gives rise.

Sensor networks

The primary value proposition of pervasive storage for networks of sensors is a decrease in the cost of deployment with an accompanying increase in flexibility. A significant cost factor in the deployment of sensor-based networks today is the network infrastructure required to upload the data they collect for analysis and permanent storage. With ubiquitous access to persistent storage, sensors could be deployed anyplace where wireless connectivity exists, and could download collected data to storage cells (or directly to data's home sites in the case of the pure storage farm scenario). Systems with access to data's home sites would have instant access to up-to-date data from an entire network of sensors. In the case of storage cells, downloaded filtering and preprocessing software could discard unneeded data (e.g., below-threshold samples) or otherwise filter and preprocess it (this would be especially apropos if storage cells were equipped with active storage devices capable of executing downloaded processing modules.)

Sensor-based networks, particularly those deployed reactively in response to quickly arising situations, have increasing application in a variety of fields like health care, agriculture, traffic control, law enforcement, and RFID tag tracking. The flexibility to deploy sensors without constructing a custom network should increase the usefulness of this paradigm manyfold.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Data centers

The discussion group anticipated that, as is the case today, digital information that enterprises regard as important will continue to be held in physically secure, environmentally controlled, professionally managed data centers. In that sense, pervasive storage would represent less change in this usage scenario than in the three discussed in the preceding sections. Nevertheless, the availability of pervasive storage could result in significant change in the way that enterprise data centers operate, manage data, and protect themselves against disasters.

The major potential operational changes for enterprise data centers are rooted in the assumption that data's "home" location would lie somewhere within a distributed "farm" of persistent storage owned and operated by enterprises or by service providers. The discussion group's assumption is that replication technology, already fairly advanced today, will evolve to enable transparent migration of data's "home" location to a site that the farm determines to be optimal according to some policy (e.g., "distance" to most intensive users), as well as to automatically maintain an appropriate number of replicas in appropriately separated locations, again according to an automatically implemented policy.

The assumption of remote replicas offers potential for improving both data protection and disaster recovery strategies beyond what is customary today. Backup, already being supplanted today by snapshot-based replicas, can be replaced by replicated data change logs, in principle enabling nearly-instant recovery of data to any point in time. Remote replicas of operational data enable rapid recovery from data-destroying disasters by re-routing application requests to a surviving replica, which becomes the "live" data. In the storage cell configuration scenario, cells could provide buffering for especially intensive operations such as mirror synchronization or replica initialization, thereby reducing enterprises' need to configure private resources to handle these peak load periods.

The discussion group also suggested the possibility that an enterprise version of the "storage collectives" mentioned earlier could arise. Enterprises joining a collective would become eligible for short-term "loans" of storage to accommodate short-term peak storage requirements. Again, the value proposition would be elimination of the need for each enterprise to outfit itself with sufficient storage to meet its peak requirements.

Finally, locating storage cells at physically secure locations (digital "vaults") would reduce the effort required to archive data, an increasing concern for enterprises as regulatory authorities increase requirements for rigorous corporate governance.

Research questions

Neither the pervasive storage configuration scenarios nor the usage scenarios described in the preceding sections defy credibility. By and large, the underlying storage and data management technologies required to implement either of the pervasive storage configuration scenarios described either exist or are within reach. Networking technologies are a bit more of a stretch, but with 1-Mbit wireless being test marketed today, it seems plausible that affordable, near-ubiquitous connectivity could be available within a decade. As with many things, however, the devil is in the details. The pervasive storage configuration and usage scenarios outlined in the preceding sections will require that some significantly difficult questions be answered before implementation is practical. This section enumerates the discussion group's assessment of the most important research topics that arise from consideration of pervasive storage. Issues

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

inherent in the basic pervasive storage model, as well as those specific to each of the usage scenarios are listed.

Research questions related to the basic models

- Storage cells vs. pure storage farms – The most fundamental question raised by the integrated pervasive storage models described in this paper is the question of which of the two models is preferable, in terms of bandwidth and latency delivered throughout the network, scalability, and economics.

- Name space management – both of the models described implicitly depend upon a universal naming scheme that allows information objects to be uniquely identified and associated with a (possibly changing) "home" location that is responsible for tracking object state and administering policies defined for the object. In the pure storage farm model, enterprise name spaces might be disjoint, but would clearly have to scale to $O(10^{15})$ objects. In the case of storage cells, presumably operated as a networked service for many storage farms, the name space would be even larger. One set of research questions deals with how to construct such name spaces that are simple enough to implement, flexible enough to change with subscriber requirements, and scalable enough to grow with demand. Other research questions involve name space consistency (especially in the presence of replicas), federation (name spaces that are managed privately, but can be aggregated with the global name space so that the information objects they represent can participate in integrated pervasive storage), and locating objects in a gigantic name space with practical levels of responsiveness.

- Security – the integrated pervasive storage models described herein rely completely on very high security. In his presentation to the workshop, Robert Gray of IDC made the point that there is precedent for trusted electronic repositories in the worldwide banking system. For either of the pervasive storage models described herein to succeed, a comparable level of trustworthiness must be established, especially in the storage cell model, where private data is cached on "public" cells. Security includes authentication and authorization, as well as security of data against "man in the middle" and denial of service attacks. For acceptance by consumers and, to a lesser extent, corporate citizens, there is also a question of making the security model simple enough for non-computer professionals to understand and manipulate. The security of integrated pervasive storage is partly a technology question and partly a socio-cultural question; both aspects represent valid research questions.

- Metadata – earlier, this paper alluded to different types of metadata and different ways of managing and using it. While generally adequate solutions exist for today's environments, it was not apparent to the discussion group that a comprehensive architectural approach to metadata exists. In an environment of integrated pervasive storage, there must be clear rules surrounding metadata, including what is intrinsic (must be present), how optional metadata (attributes) can be defined and interpreted (e.g., for attribute-based searches), and how metadata and policies should be allowed to interact. Even today it is clear that some metadata is used for management and other reflects the "business meaning" of data, and is used by applications. The ability of arbitrary applications to "make sense of" arbitrary information objects could increase the value of integrated pervasive storage, particularly for sensor network and data center usage models, and so seems worthy of research.

- Data consistency – two of the key benefits of the pervasive storage models described earlier are the ability to replicate information objects for resiliency and the ability of multiple "users"

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

to share objects (possibly using separate instances). These capabilities, constrained by the proverbial "speed of light", require research into the issue of data consistency. What can be said about an information object that is being modified by two users who are widely separated, unaware of each other, and operating on separate instances of the object? As with security, data consistency is probably partly a technology and partly a cultural issue (as they have with DNS services and online stock quotes, users and applications may have to "learn" to expect different data properties in a pervasive storage environment), and both appear to be worthy of research.

- Physical data management – the concept of integrating pervasive storage brings with it the usual set of distributed system questions around cache management (particularly in the storage cell configuration scenario), replication algorithms, the "home" of information objects, alternate access paths, recovery and work-around from failures, and so forth. These questions are well known from a large body of existing research on distributed file systems; the pervasive storage environment only extends the scale on which they must be considered.

- Legacy – as with any evolutionary technology change, pervasive storage will have to accommodate itself to the private data environment existing when it is introduced. Research questions in this area include merging name spaces, and encapsulating information objects so that they appear "native" both to the pervasive storage environment and to the applications that use them.

- Data filtering – part of the attraction of the storage cell configuration model is in the work of one of the conference's other discussion groups – the active storage devices group. Active storage is envisioned to be programmable in some sense, possibly with application-specific functions. This gives rise to the possibility that storage cells incorporating active storage devices could perform some significant processing on data as it passes through them. For example, video taken by a consumer might be compressed to conserve bandwidth and storage at the consumer's home location. Sensor data might be scanned for exceeded thresholds or other anomalies, and discarded if not worthy of transmission to a home location for analysis. Again in this case, the attraction is reduced bandwidth consumption. Research questions in this area are probably identical to those in the active storage devices thrust; the pervasive storage concept is likely nothing more than an additional perspective from which to view active storage devices.

- Disconnected operation – as mentioned in the introduction, the discussion group did not have adequate time to explore the question of disconnected operation in a pervasive storage environment. Intuitively, however, it is clear to any mobile telephone user that universal connectivity is an unrealistic assumption. This raises research questions around how important disconnected operation is likely to be (e.g., would a "storage cell busy" signal be an adequate solution), as well as how to deal with users' inability to reach needed data.

- Economics – pervasive storage represents a radical departure from today's closely held concept that data is private and should be stored on private facilities. Even if the technical, security, and cultural challenges can be overcome, there is still the question of whether either of the massively interconnected storage models described in this paper can be made economically viable. Clearly, for integrated pervasive storage to be viable, it must be implemented on a pervasive scale. A whole series of questions about balancing up-front investment, deployment strategies, and steady-state operational economics are worthy of research in parallel with, or even prior to, the technical questions.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Research questions related to consumer usage scenarios

- Storage collectives – one usage scenario of probable interest to consumers is the short-term loan of storage to meet burst needs. For someone recording raw video, it may "borrow" storage from a pool, to be returned when editing produces a finished product. The storage exchange concept was introduced by UC Berkeley's Ocean Store Project (which is, in essence, an exploration of the pervasive storage concept). For a practicable pervasive storage exchange, much more research is required into questions of service quality, data reliability, access performance, security, and space management.

- Data streaming – a major attraction of pervasive storage to consumers is the vision of lighter weight, more robust, more functional personal appliances with longer battery life. As outlined earlier in this paper, this would fundamentally be achieved by streaming data to and from appliances on a more-or-less "just in time" basis. This raises a set of research questions dealing with energy consumption (substituting broadcast energy for energy to spin a disk drive), ability to stream with adequate performance using foreseeable wireless technologies, local buffering, and mobile users (e.g., passing a user from one cell to another during a stream) that would need to be answered before consumer data streaming applications could be considered practical.

Research questions related to corporate citizen usage scenarios

- Thin clients – while the consumer-oriented personal appliance model has some attraction for corporate citizens, a greater attraction for today's usage patterns is a "thin client" model for business applications. Part of the attraction of a thin client with no local application data storage is the smaller, lighter device with access to more and better data, but at least as large a consideration for enterprises is data security. With no data going "off premises," enterprises need to worry much less about loss or theft of data. The benefits seem obvious, if the model can be implemented successfully. Two sets of research questions arise, however. First, is user acceptance. Technically successful thin clients have been implemented in the past (e.g., by Digital, Oracle, and Sun), but have not enjoyed widespread success in the market. The question of whether these earlier efforts were simply before their time or whether they failed for other reasons deserves exploration.

- Application fragments in storage cells – for optimal responsiveness of business applications to thin clients connected to pervasive storage, it seems reasonable to partition them among the data home location (business logic), the thin client (display and input), and the storage cell (data delivery from cache, input filtering and editing). Downloading application fragments (applets) is one thing in a controlled environment; it is quite another in a storage cell that is accessible to the public at large. The research questions here are about providing a secure environment in which many applications and data chunks can be cached and executed on behalf of many unrelated users at adequate performance levels without compromising security. Another set of research questions is about finding an optimal partitioning for the storage farm and storage cell environments.

Research questions related to sensor network usage scenarios

The research questions raised by the sensor network usage scenario are similar (but not identical) to those for corporate citizens.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Managing unbounded data flows – in principle, a large network of high-frequency sensors can generate a very large amount of data, potentially swamping its home storage location. In this environment, storage cells can easily become a bottleneck – if they can't send collected data to the home location fast enough, their capacity could eventually be saturated and the difficult choice between discarding input and discarding cached data would be forced. The research questions in this area are about managing potentially unbounded data flows with limited resources, presumably using a combination of cache, flow control mechanisms, data filtering at the storage cell (using application-specific downloaded filters), and data discarding.

Research questions related to data center usage scenarios

The discussion group's working assumption was that the data center, either owned or provided by an ASP, would continue to be where enterprise data "lives." Information asset security, professional management, economies of scale, and the large amount of data owned by enterprises relative to the fraction typically required by individual employees all militate in favor of the data center, approximately as we know it, being the primary repository of enterprise information assets for the foreseeable future. A pervasive storage environment could significantly change the way data centers operate, however, largely due to opportunities for advanced management and application functionality.

- Reconciliation of regulatory requirements – pervasive storage would make it easy to replicate data across jurisdictional boundaries, or to view or operate on data stored in one jurisdiction using applications running in another. This raises both import/export questions and questions of how to handle data that may be required in one jurisdiction and forbidden in another. To be sure, an enterprise can set policies to deal with any given case, but this solution cannot possibly scale. The research questions here are about standardization of business usage metadata so that enterprises can define general policies for handling data that is replicated across jurisdictional boundaries.

- Legacy data – while incorporating legacy data, whose name space may be private, and which may lack full metadata, into a pervasive storage environment is a challenge for the concept as a whole, it primarily surfaces in the data center environment, since data centers are at the server and repository end of pervasive storage information service environments. There are techniques for mapping private name spaces to a global name space; the larger challenges lie in scaling, and in supplying business-oriented metadata (e.g., XML-based descriptions of data's business significance) so that data can be used by multiple applications.

- Data management – data centers whose storage farms are interconnected and operated as an integrated whole create data management challenges. One set of challenges lies in access security (e.g., through strong authentication, rigorous authorization, and robust encryption), another in preserving data from loss or destruction (e.g., by maintaining adequate replicas in appropriate locations), another in performance (e.g., by judicious selection of virtual device type and intelligent caching, replication and migration), a fourth in protection against corruption (e.g., by logging and reconstruction), and a fifth in the destruction of data when its lifetime has passed (e.g., by atomic deletion and overwriting). At some level, solutions to all of these problems have been developed. The research challenges for pervasive storage lie in solutions that scale to very large distributed storage environments, and that operate in environments where enterprises' storage may be shared

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

with other enterprises (the "enterprise collectives" mentioned earlier), and data may pass through publicly accessible storage cells on its path between client and server.

Reagan Moore of the San Diego Supercomputer Center (SDSC) pointed out to the discussion group that several of the questions arising primarily in the data center context had answers in the form of SDSC's *Storage Resource Broker (SRB)*, a clearing house for (largely scientific research) data in the high-performance computing community. Due to shortage of time and fluctuating membership, the group did not have the opportunity to explore whether the solutions embodied in the SRB might extend and scale to the more general pervasive storage environments postulated earlier in this paper. This is surely an area for further discussion before energy is expended on reinventing these particular wheels.

Summary

The pervasive storage discussion group believes that integrated pervasive storage environments, as hypothesized earlier in this paper, are nearly feasible today, but this by no means makes it a certainty that these environments will ever be implemented. There are apparent benefits, in terms of greater "connectedness" to data for individuals, and greater data asset security from both misuse and destruction for enterprises. But as the latter sections of this chapter enumerate, there are many significant unknowns that would have to be resolved prior to approaching realistic implementations. Some of these are technical, for example developing security and caching facilities that keep data safe while it is being accessed at realistic levels. Others are economic – does the balance between cost and benefit really fall in favor of integrated pervasive storage? Still others are cultural – will individuals ever give up their personal copies of data, and will enterprises ever allow their digital data to reside on the devices that hold their competitors' data as well? There appear to be many potentially fruitful areas for research in the pervasive storage space. Because no vendor is likely to supply all of a pervasive storage environment, and because the elements of such an environment must inherently cooperate closely, much of the research on the basic issues outlined in this paper could clearly be conducted in a collaborative, pre-competitive environment.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

SECURITY AND PRIVACY

Participants

James Hughes, StorageTek (Thrust Leader)
Roger Cummings, VERITAS Software Corporation
Howard Gobioff, Google
Robert Thibadeau, Seagate Research
Jeff Wisted, Agere Systems

Introduction

Security can be approached from many directions. Functionally, it provides safety, which is freedom from risk or danger, and confidence, which is freedom from doubt, anxiety or fear. There are many examples, including building guards that note and act on suspicious behavior, and technical and legal measures to prevent theft, espionage, sabotage or attack. There are measures by businesses and homeowners to prevent burglary or assault.

Privacy is the quality or condition of being secluded from the view of others or a state of being concealed. Privacy is often employed in achieving a measure of security. There is also personal privacy that is a type of security applied against the theft of personal information.

The topic of security and privacy for storage systems was cited as a natural key element of every other thrust with good reason. Security and privacy are concerns when theft is of concern, and it is well known that the data being stored is most often more valuable than the equipment that is storing it.

When approaching pre-competitive problems for security and privacy in storage, there are a large number of difficult problems. The list includes problems of analyzing security in enterprises employing a large number of firewalls, natural holes due to individual dial-ins, carry-ins, carry-outs, and foreign VPNs. In many ways, insiders represent the ultimate threat. The system administrator in some ways has to be the most trusted individual in an enterprise, but this is not a desirable state of affairs. Then there is the question of what happens if a breach occurs. How can the enterprise measure and correct the damage?

Security in depth needs to look at security at all the interfaces in storage systems management. It cannot simply overlay existing systems. This includes intrusion detection at all layers, data protection in transit and at rest, uses of cryptography to achieve privacy and confidentiality, and limitation on the capability of insiders to maliciously or accidentally compromise a system resource. We believe it is desirable to be able to administer storage without having access to the information stored, to be able to delegate reading capability without the authority to read and, in effect, to more carefully define the roles of the administrator and other agents in a storage system.

With holes at essentially all levels of storage system management, it is important to position security and privacy research at all levels so as to avoid single points of security failure that can, in effect, threaten all the assets.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Corporations are reactive and standards are evolutionary. Research should focus on discovering new directions that are usable and transparent for tactical and evolutionary success.

Research Topics

The basic research topics of security are privacy, authorization, destruction, audit, integrity, keys, intrusion prevention and detection, availability, economic issues, operational risk, and transparent security. The remainder of this section will describe these pre-competitive research opportunities in turn.

- Privacy and confidentiality

  Privacy and confidentiality are hard to achieve in practice. The major reason is information leakage owing to insider attacks, lack of proper access controls, improper policies, or architectural deficiencies. Each of these areas can be substantially improved over existing implementations.

  Cryptographic control over privacy is always desirable. There is nothing wrong with cryptographic control (i.e., encryption) at the drive level, above the drive, and/or above the file system, as ways of layering protection, as long as the performance penalty is within acceptable bounds. Each level has its own cryptographic demands and may have optimal cost/performance/privacy tradeoffs.

  We assume that when cryptography or, for that matter, nearly any form of access control, uses keys, these keys must be at least as secure as the data.

- Authorization

  Authentication and authorization is a central component of storage security. We need to authorize both people and devices. The system management of multi-tiered authorization is clearly a research problem.

  The “voting problem” is not well addressed in current storage systems security. This is the problem of who is really taking a particular action. Is it the correct person, or is the action being done at a terminal that the correct person stepped away from, or is the person under coercion to perform certain acts? Banks have long incorporated such things as silent alarms to allow a person under coercion to give the appearance of complete cooperation. Similarly, honey-pot systems that attract malicious actions may be used to trap potential insiders. Work in these areas is certainly pre-competitive, and important.

- Destruction

  Data destruction is desirable both to insure absolute privacy of information from discontinued elements of a storage system and to insure compliance with laws and policies that require data destruction after a time. Cryptography can clearly aid in data destruction by providing secure destruction through the destruction of the keys and by providing for rapid destruction. A key of 128 bytes can be destroyed in order to destroy arbitrary amounts of encrypted data that may otherwise take hours to destroy by repeated disk erasure. There are issues about granularity of destruction that may interact with key systems, and it is certainly true that self-protecting storage systems may require special features in order to support assured data destruction.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

The hard problem in data destruction in a storage system is in identifying all of the cached copies, backup copies, of the files to be destroyed. This requires traceability that current storage system architectures do not provide. The nature of the traceability is open to research.

- Audit in storage

  Audit logs are essential in security systems because they provide a means to investigate possible attacks. Audits should include user activity, device activity, and management activity. From a system management perspective, audit logs should be guarded. They may be write-append only for write operations, and read only by forensic investigators, such as administrators. Obtaining an overall view of audits and providing interoperable audit formats so that a single sink can accept meaningful audit information from multiple sources is essential.

- Integrity

  Integrity assures that the data is what you think it is even in the face of adversaries. Integrity assurance is essential both for the wire and the media. Data integrity on media can be provided in many ways, but certainly should include a way to prevent un-auditable change to the data, if desired. Cryptographic means of providing for data integrity include the storage of hashes in different access environments or through signing. Developing comprehensive, system-level notions of data integrity and data integrity management are worthwhile research endeavors.

- Keys

  Key manipulation and management provide rich areas for improvement. Key sharing is the opposite of key security, but key sharing must, to some extent, be allowed and, therefore, must be properly managed.

  Key management for communications is different than management for data at rest. Since data is persistent, there is the 'first key' issue. You may not know who, in five years, should have access to the data. How do you pass the proper credentials forward and make sure that the data made private is always recoverable by the currently authorized agents? Conversely, how do you revoke keys with comprehensive effect, and how do you avoid revoking a 'last key'?

  The Windows Encrypting File System is an interesting prototype. In it, each file has a header composed of a number of user-agent records and a number of recover-agent records. Each of these records contains a public-key-encrypted symmetric file-encryption key. The symmetric file-encryption key is always the same for the file. But to recover and use this symmetric key, the user-agent or the recovery-agent must use his private asymmetric decryption key to reveal the symmetric key. Thus, any number of users or recovery agents can have file encryption/decryption rights. This system was noted to be highly scaleable, but it does leave leaks in the system if the key management is imperfect for user and recovery agent public-private key pairs.

  An interesting question comes up whether there can be such a thing as "provable" key management that proves the correctness and completeness of a particular system instantiation.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Intrusion prevention/detection

  As with the other research topics, this topic is best viewed in terms of security in depth, across multiple interface levels. In some sense, each interface can contribute to providing different, complementary, types of intrusion prevention and detection. Identifying and managing these opportunities represents a rich research area.

- Availability

  Availability is traditionally important in quality of service (QoS), but has special meaning in security to include control over denial of service (DoS) attacks, as well as the traditional issues of equipment or software failures. Research in this area should take an infrastructure focus, with attention to interfaces against which DoS attacks may take place. Knowing where the system is and is not protected from DoS attacks is essential in understanding the overall security structure.

- Economic issues

  Is storage security a price-elastic market? It seems clear that it is difficult to put a value on security, since a single security measure may not have value except in comparison to other security measures and the fluctuating value of the data itself. However, we should also take any information security measures and compare these with physical security measures. Most enterprise storage systems first rely on physical security and have traditionally relied less on information security, but as storage systems become more pervasive and more internationalized, it seems clear that physical security cannot stand alone. An interesting study would be to determine what insurers have been able to identify in terms of the economics of various information security features of an enterprise storage system.

- Operational risk

  Operational risk must be dynamically assessed, but can also have static attributes. Common criteria protection profiles subject to laboratory certification may be a highly desirable outcome of research aimed at level-setting operational risk. The dynamic aspects of controlling operational risk involve a strong commitment to transparent security. If it is difficult for an operator to incorporate security, then he or she may choose to violate security policy. Transparency should be engineered into the entire enterprise security system so that people are not put in the situation where they must sacrifice security in order to meet deadlines or departmental goals.

Summary

We believe there is enormous opportunity to substantially improve storage systems security through pre-competitive research on the issues highlighted in this section of the roadmap. Total systems awareness, coupled with security at every interface and specialized security for data at rest, will be hallmarks of this research.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

AUTONOMIC STORAGE

Participants

Remzi Arpaci-Dusseau, University of Wisconsin, (Thrust Leader)
David Du, University of Minnesota
Robert Gray, IDC
Les Hodge, StorageTek
Arnold Jones, Storage Networking Industry Association
Dilip Kandlur, IBM
Ethan Miller, University of California, Santa Cruz
Wayne Rickard, Seagate Technology
Ahmed Tewfik, University of Minnesota
Curt Tilmes, National Aeronautics and Space Administration
Geoff Voelker, University of California, San Diego
Qian Wang, Pennsylvania State University
John Wilkes, Hewlett-Packard Laboratories

Introduction

The field of computing systems, both in industry and in research, is changing. In the past, improving system performance was at the forefront. However, as the years advanced and Moore's Law continued to improve processor performance at incredible rates, performance is no longer the primary trouble for many large-scale applications.

Indeed, many luminaries in the field of computing are advocating new directions for systems research. For example, Jim Gray suggests that a major goal of systems research is the development of “trouble-free” systems. In such a system, a human simply sets the goals, high-level policy, and the budget, and the system does the rest [Gray]. Dave Patterson advocates Recovery-Oriented Computing (ROC) which places emphasis on availability, changeability, maintainability, and evolutionary growth (and not on performance) [Patterson]. Finally, Butler Lampson extols us to build “systems that work”; such systems are highly-available, adaptable, easy to grow, and (as an afterthought) deliver the needed levels of performance [Lampson].

From IBM came a similar vision of the future of computing, under the scientific name of “autonomic” computing. Put simply, autonomic computing systems manage themselves. The term autonomic is a reference to the autonomic nervous system (ANS) of the human body, which regulates many of our organs and muscles, all without conscious thought. The analogy is clear: autonomic computing systems should manage themselves automatically and without human intervention.

In this document, we present a summary of the autonomic roadmap, as developed through discussions at the DS2 Workshop. In our discussion, we focused on the following questions:

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Why do we need autonomics?
- What's really important in data management?
- Can we build autonomic systems with middleware?
- Can we focus solely on the storage system?
- Can we ignore the psychology of autonomics?
- What are other important issues?

As a result of our discussion, we have derived a set of research directions that we believe should be pursued to take autonomic computing toward the next level of real products and solutions. The following is a brief summary of the main thrusts of our research roadmap:

- Transparency: Techniques to expose and describe the decisions made by autonomic systems must be developed. If computer systems are to make important decisions, said systems must be able to explain their decision-making process to human administrators.
- Devirtualization: Virtualization is commonplace, and yet it interferes with performance tuning and problem diagnosis. Tools that can discover the underlying physical nature of drives and larger-scale systems are required.
- Reliability: System reliability was signaled as perhaps the most important aspect of storage systems. Hence, further research into methods for building robust and reliable systems in the presence of complex and heterogeneous systems is needed.
- Evaluation and metrics: For a marketplace of autonomic technologies, standard benchmarks must be developed in order to enable competing products and approaches to be rigorously compared.
- Study of processes and practices: To understand how to build autonomic technology, the actual processes that exist in today's systems must be studied.
- Management policies: As storage automation systems become commonplace, there must be research not only into the systems that enable policy-based automation, but research into the actual policies themselves.
- Specialized storage systems: As storage uses become more diverse and specialized, a different approach to building low-cost manageable systems is to specialize the storage systems to the needs of important application classes. Hence, further research into the design and implementation of such specialized systems is recommended.

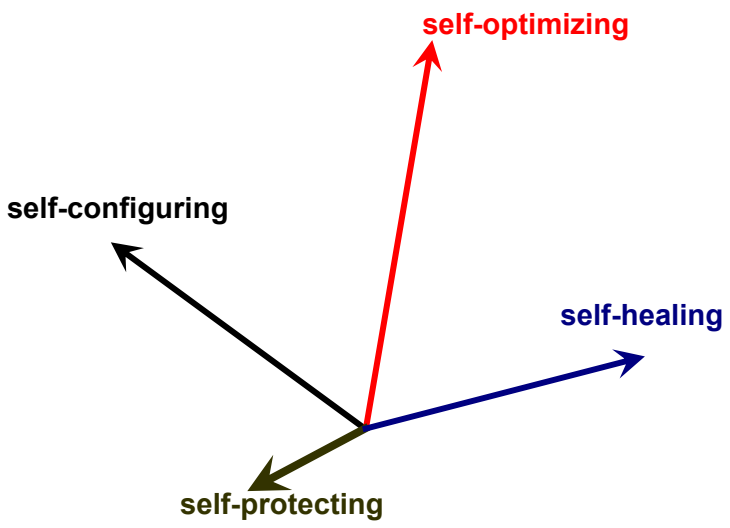
The structure of this chapter is as follows. We first present background on autonomic computing, and then discuss some important issues as related to storage systems. Finally, we present a set of research directions that we believe are crucial for the development of autonomic storage technology.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Background

We first present some background on the field of autonomic computing. Much of this discussion is drawn from Kephart and Chess's overview of IBM's autonomic computing initiative [Kephart & Chess]. Although our interest is in the realm of data storage, autonomics are not limited to this aspect of computing systems. Indeed, the field of autonomic computing is broad enough to encompass virtually any computing task where the burdens of management are high. However, our focus in this document will be on storage systems and the autonomic issues therein.

Autonomic computing is often broken down along four axes: self-configuring, self-optimizing, self-healing and self-protecting systems, as shown in the diagram below. We discuss each in turn, presenting examples from research and industry, as well as some potential difficulties with each.



It should also be noted that the main ideas behind autonomics are not new; indeed, automation of tasks has been at the forefront of computing throughout its history. For example, in older computer systems, the scheduler was a human operator, to whom you handed punch cards and who then decided which job would run next. Modern systems make scheduling decisions themselves, using sophisticated policies such as multi-level feedback to improve throughput and response time.

This example highlights two potential benefits of autonomics. First, automation sometimes reduces the need for human involvement. Second, and perhaps more importantly, it illustrates how the system itself can observe more about what the system is doing and therefore potentially make better decisions than a human could. In our example, a scheduler can make a decision of what job to run every clock tick, something no human could approach (at least, no human of whom we are aware).

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Self-configuring systems

In the current world, large-scale data centers are comprised of systems from multiple vendors and run a wide variety of software. Installing, configuring and integrating such systems is both a time consuming and error prone process; much human effort is required.

In contrast, a self-configuring system will automate the installation, configuration, and integration problem. In such a world, an administrator specifies high-level policies (e.g., minimize cost for a given performance level, avoid single-component failures at any cost, etc.), and the system will follow those guidelines in doing its work.

A state-of-the-art example from research is a system known as Hippodrome from HP Labs [Anderson et al.]. In this work, the system is given a set of workload descriptions and information about the underlying hardware. A constraint-based solver is then applied to the problem, which searches the space of possible configurations and converges upon a suggested configuration and data layout scheme. In one example, the solver produced a storage system for a TPC-D workload that performed at the same level as a system that a trained database expert designed; the major difference was that the solver achieved that level of performance with 16 disks, whereas the human-based design required 30.

Of course, general-purpose self-configuring systems will not easily be built. One problem that will certainly arise is in how policies are specified. Garth Gibson points out that rules-based policy management could be troublesome [Gibson]. These types of systems allow one to specify tests and triggers, which then can be used to "automate" certain types of functions. For example, a rule could specify "if workload is becoming more sequential, increase prefetch chunk size by 25 percent." However, a system such as this could quickly become a morass of triggers with no clear intelligent outcome. As Gibson asked, will you get the salary-free IT guru, or the 4-year old after whom you have to clean up?

Self-optimizing systems

In the current world, systems have hundreds of tuning parameters. These tuning parameters or "knobs" are often non-linear in their behavior, and interact in complex ways. Worse, as systems mature, the number of knobs to tweak rarely decreases.

In contrast, a self-optimizing system does not leave the hard work of tuning parameters to the system manager. Instead, the system continually adjusts and adapts itself in order to improve its efficiency, all without human involvement.

Recent research has produced a number of examples of this kind of self-optimizing system. For example, research by Ganger et al. at Carnegie Mellon University [Schindler et al.] shows how a file system can probe a disk to discover its low-level geometry; files can then be laid out in a "track-aligned" manner, improving performance. More recently, work by Denehy et al. at the University of Wisconsin [Denehy et al.] shows how a file system can probe a RAID system to understand its inner workings; the file system can then adjust its behavior to improve bandwidth, for example by performing stripe-aligned writes.

A different approach has been taken in the NCR Teradata parallel database. This system uses hashing to distribute data across the nodes of the system; it does not export knobs to users which could be used to tune its performance. As one observer has reportedly said about it:

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

"The performance is bad, but it never gets worse." Consistency is a valued property, and the lack of human involvement is a desirable feature.

One likely problem that will arise with systems that probe and adjust the behavior of their components is that not all knobs are equivalent, i.e., the semantics of the knobs is important. For example, consider the "immediate report" bit on a disk, whose function is to determine whether writes must be immediately forced to the disk media or can be buffered in RAM first. A self-optimizing file system could automatically discover that by turning it on, write performance improves, because writes simply deposit data into the disk's memory buffers. However, the file system may not be as satisfied when it discovers the meaning of the bit: that it implies the disk does not immediately commit data reliably to disk. Hence, there is contention between competing criteria; to make a better decision, an understanding of what the knobs mean is required.

Self-healing systems

In the current world, bugs and failures in complex systems are hard to find, and even harder to reproduce. In the worst case, it might take a team of many people many weeks to diagnose and solve such a problem.

In contrast, a self-healing system will automatically detect, diagnose, and repair software and hardware problems. Human operators may not even be aware that a problem has occurred, or may have some minimal involvement as the system informs them to "replace disk 33, please."

A classic example of a self-healing system is found in RAID storage systems [Patterson et al.]. When a single-disk failure is detected, the RAID may decide to instantiate a hot spare. The system then reconstructs the failed disk onto the hot spare, and when finished, returns to fully normal operation. During the recovery process, performance may be slightly degraded, but otherwise the entire process is transparent to the host system – a perfect example of a self-healing system.

More recent work in this area is that of Chen et al. on path-based diagnosis [Chen et al.]. In this work, the authors develop techniques to trace requests through both complex distributed systems as well as in component-based server applications. They then apply statistical analysis to diagnose failures and find bottlenecks. This technology clearly would be useful in any autonomic diagnosis system.

Automation of bug fixes and failure handling will be difficult for a number of reasons. It is well known that automation does not cure human error [Patterson]. Indeed, it often shifts human errors to design errors, by making the design of the automated system more complex than the system that is not automated. Design errors are generally more difficult to detect, tolerate, and fix, as automation tends to address the easy tasks. The result is that the remaining tasks are complex and unfamiliar ones for the humans in charge.

Self-protecting systems

In the current world, dealing with security problems is a human-intensive task. Security problems arise through malicious attacks that may alter the contents of storage media. To diagnose and fix the system, operators typically collect logs of activity, sift through logs and graphs to detect problems, and manually correct what is wrong, hopefully restoring the system to an uncompromised state.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

In contrast, a self-protecting system will automatically detect and defend against malicious attacks. Although humans may be notified, the system itself should be able to cope with a wide range of attacks upon it, all without disrupting mainline service noticeably.

An excellent example of self-protecting technology comes from [Forest et al.]. In this work on "computer immunology", the authors show how to measure and classify common system call patterns that are made by typical applications. Then, if the operating system detects a non-standard pattern, it can assume that the new workload is malicious and act accordingly; perhaps it would terminate the offending process, perhaps merely slow it down.

However, self-protecting technology introduces problems as well. One example that showcases some potential difficulties has to do with software complexity. In general, to make systems more secure, one must write more complex software to detect and react to attacks. However, it is a well-known adage of computer security that "complexity is the enemy of security." Similarly, it is also well known that diversity aids in security [Forest et al.]; for example, Akamai emphasizes how it uses many different brands of OS, router, PC, etc., in order to be more resilient to attacks. However, it is also well-known that diversity increases the difficulty of system management, which is what autonomics is attempting to reduce.

Issues

We now present a summary of the issues we discussed at the DS2 Workshop. We discuss such basic issues as why we need autonomics, what is really important in storage management, the role of middleware, the storage-only focus, and the "psychology" of autonomics.

Why do we need autonomics?

One of the first issues we addressed was the basic question of why there is a perceived need for autonomics. One quick but perhaps specious answer was that it is simply about lowering costs.

Certainly total cost of ownership as compared to raw hardware costs are quite high, ranging anywhere from 3x to 10x (depending on which source you cite). Perhaps this is a result of the drive industry's success; as one of us put it, "disks are essentially free."

But where do the costs come from? We believe cost is a direct result of the complexity of modern data centers. There are simply too many human-intensive tasks that must be performed, which are not only expensive, but error prone as well. As one customer stated, "Please get me out of the enterprise janitorial business."

However, we also came to see that autonomics can have benefits beyond the simple reduction in total cost of ownership. This outcome is a good (and perhaps necessary) outcome for companies that develop autonomic technology, which of course would like to make money on the technology they build.

There are many reasons that autonomics goes beyond simple cost savings. First, autonomic systems can automate a range of tasks to which humans can't respond. Humans have reaction times measured in seconds, whereas computer systems can make many fine-grained decisions at the millisecond level. Further, computer systems are excellent at streaming through vast quantities of data in order to explore the best possible options; humans often have to resort to

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

"rules of thumb" or "best practices." In all of these cases, autonomics has the potential to speed up and improve the state of the art and lead to new business opportunities.

What's really important in data management?

We also discussed what was most important about storage management. Without a doubt, the answer was "don't lose the data." As one of us succinctly put it, "data loss implies job loss."

Beyond this guiding principle, we agreed that finding and fixing problems (i.e., the self-healing axis of autonomics) stood out as most pressing. In large, complex systems, failures occur but are not easy to find, much less cure. Worse, failures often manifest themselves as slowly degrading performance rather than an absolute breakdown.

We also agreed with our initial guess that performance isn't as important as it used to be for most consumers of storage technology (as one of us put it, "95% of customers don't care about performance"). What is important for many customers is predictability; a consistent level of performance (as in the Teradata example above) is quite desirable. With all of that said, one of us did say that when performance *does* matter, it is often a good revenue source.

Can we build autonomic systems with middleware?

One view of the world puts forth autonomic technology as a layer of middleware that makes everything in an existing data center work together in an automated and self-managed fashion. We believe that this approach will likely be limited in applicability and success. Rather, we believe that the components of these systems will have to change as well.

Consider the simple example of a RAID storage system. Internal to it (and hidden from the outside in many cases) is a sophisticated machine that reacts to handle a variety of internal failures. However, when deployed in a larger setting, the RAID does not typically expose any of its internal state, thus complicating decisions made in higher levels of the system (e.g., data layout across multiple RAIDs). If an external agent knew, for example, that a disk-intensive RAID reconstruction were ongoing, it may choose to lay out a new dataset across another RAID system instead.

The general problem here arises from how systems are typically designed – in multiple layers of abstraction. Layering is an age-old and successful technique, but it has its costs as well. One such cost is information hiding; components often do not expose much of the information they have gathered. In an increasingly integrated world, one must consider how to access and utilize this information in a vertically-integrated fashion. Access to such information may be of great import to an autonomic system.

Can we focus solely on the storage system?

Given the storage bias to our group (and indeed, the DS2 Workshop as a whole), there may be a temptation to focus on storage systems alone when considering autonomic technology. Indeed, narrowing the focus from entire systems down to just the storage component likely makes the problems simpler, easier to understand, and hence more tractable.

However, we believe that understanding applications and how they fit into the picture is crucial in developing autonomic machinery. Applications are fundamentally intertwined with the rest of the system; indeed, they are the place where failures first manifest themselves to users and

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

managers of the system. Hence, there is likely a need to understand the types of applications that are running on data centers, how they behave and, more importantly, what they expect from the storage systems with which they interact.

Can we ignore the psychology of autonomics?

One of the major outcomes of our discussion was the realization of the important "social" aspects of autonomic computing – that is, the reluctance to give up complete control. We call this the "stick shift" phenomenon; administrators of high-end storage systems are used to having full control over the variety of decisions that exist in storage systems, to the extent of knowing where every last byte was placed on disk.

In an autonomic world, in order to relinquish such control, trust is required; administrators have to allow the system itself to make decisions that the administrator is used to making. Previous experiences indicate that such trust does not often exist, or is not easily obtained. As a specific example, let us consider AutoRAID, an early example of an autonomic storage array developed by Hewlett-Packard. Although it is an excellent piece of technology, automating the process of determining which RAID level to use for individual data blocks, it did so in a manner that was opaque to end-users. The result was mixed; the technology worked, but the product was not accepted by system managers who wanted more control.

Hence, we believe that the "psychology of autonomics," that is, how to get users and administrators of these systems to be comfortable with them, is a primary and important challenge for autonomic research. This problem should be addressed through technical means; that is, we must all think about how to build systems that address the psychological aspects of autonomics and "giving up control," in addition to the other automation problems we plan to solve.

Fortunately, there are examples of other systems that have dealt with these kinds of sociological problems. For example, consider cycle-stealing systems such as Condor [Litzkow & Livny]. These systems determine when your desktop machine is idle and then use it to run other people's jobs. Not surprisingly, these systems faced a huge sociological hurdle; people did not like that they were seemingly not in control of what was running on their machine. The technical solution to the problem was simple: provide more controls to the owner, to let them determine the level of computer usage with which they feel comfortable.

What are other important issues?

Systems expand and grow over their period of utilization, such that a system at the end of its useful life is much larger and more complex than upon initial installation. In other words, the monotonic growth of a system is accompanied by increasing heterogeneity of its components. Similarly, vast complexity is commonplace. Thus, we envision that a major challenge will be not only to develop autonomic technology, but to demonstrate that it works on vastly disparate systems, and continues to work well over time.

We also discussed security and its importance. All agreed that security concerns were becoming more important in the storage domain, especially with the move to IP-based storage. Further, it often is difficult and complicates what you are trying to accomplish. That said, it seemed most orthogonal or auxiliary to the general goals of self-managed storage.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Research directions

In this section, we outline a number of suggested research directions for autonomic storage systems. We discuss transparency, devirtualization, reliability and robustness, evaluation and metrics, the study of processes and practices, the development of policy languages and management policies themselves, and the construction of specialized storage systems.

Transparency

"We need transparency." — Kofi Annan

One of the clearest outcomes of our discussion was that autonomics needs to take the psychology of administrators and users into account. Therefore, "black box" approaches that simply hide complexity in a manner that is inaccessible to system managers is unlikely to be successful.

We believe that "transparent" technology is crucial to the success of autonomics. Specifically, transparency refers to the extent which an autonomic system reports the details of its internal policy decisions to the system manager. Fundamentally, if computers are to be trusted to make important decisions, they need to be able to explain why they did something, e.g., "I did this because this condition arose."

Many issues must be addressed. For example, how should systems express what they are doing? A language is needed that is fully expressive and understandable to humans. At what granularity should this decision-making be expressed? Too much information will overwhelm a human, too little will not be useful. How much historical data must be tracked in order for a human to come back at a later date and understand the history of the system's decision making?

Devirtualization

"Hide not your talents, they for use were made. What's a sundial in the shade " — Benjamin Franklin

Modern data centers are comprised of many interacting systems, and are increasingly becoming virtualized. On top of the simple disk, there may not only be a RAID virtualization layer, but also a virtualizing switch (which aggregates many RAID boxes together), and a front-end which may add yet another layer on top.

One key to an autonomic system is to be able to understand how the components of a system are working, e.g., how they are performing, why they are failing. Virtualization tends to hide these aspects of the components of the system, and thus there is a need for technology to "devirtualize" systems, to make available details of the system.

Devirtualization refers to access to behavior and control of the physical subsystems of the storage system, which may be otherwise managed under a virtual device protocol that hides the actions of discrete elements of the system. An example of a devirtualizing tool is found in many disk and RAID characterization tools, as discussed above [Schindler et al., Talagala et al., Denehy et al.]. All of these techniques are first generation tools, and further research is needed to keep up with the virtualization that is increasingly common in data centers.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Another research direction within the scope of devirtualization is the design of component-level mechanisms to aid higher-level tools and systems. For example, although a disk presents a simple linear array to higher-level systems, it may be useful to consider how to export aspects of its inner workings, thus obviating complex disk extraction tools. Instead of hiding information, components in these systems should aid higher levels by exposing some of their internal machinery and state. There are some examples of this in other layers of the system [Arpaci-Dusseau et al.], but today's storage system remains stubbornly closed.

Reliability and robustness

"The shifts of fortune test the reliability of friends." — Cicero

One of the outcomes of our discussion was the clear need for self-healing systems technology. Therefore, it is our belief that there should be more research into techniques to build robust and reliable systems.

Reliability and robustness refer to the capacity of the system to maintain a given, although possibly not optimal, level of service when confronted with erratic behaviors stemming from the complexity of the system itself. There has been some promising recent work in this direction from the Recovery-Oriented Computing (ROC) Project at UC Berkeley and Stanford; those researchers have been investigating new ways to build robust systems in spite of the failures that are common [Chen et al., Candea et al.]. However, as this body of work presents first steps in this direction, more research is needed.

One direction that should definitely be pursued is to better study how modern storage systems behave when they fail. What happens when a disk block becomes unreadable? How does the file system or database above react? Such studies will be a key first step into understanding what types of machinery we need to build to enable truly robust systems.

Evaluation and metrics

"When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind." — Lord Kelvin

The autonomic management of storage systems emphasizes that there is more to storage systems than performance. Unfortunately, for years, the basic method we use for comparing systems from different vendors is to apply standardized benchmarks to them (e.g., Laddis, SDET, Andrew, etc.), which are, alas, only performance comparisons.

Evaluation and metrics address the need of to establish quantitative canons of system behavior, which would enable the precise comparison of the levels of user service delivered by each of the systems under scrutiny. In this new era, we will require methods to compare systems not just on their performance capabilities but also on their ability to optimize themselves, configure themselves, heal themselves, and even protect themselves. Deriving benchmarks that quantitatively assess systems along these axes is crucial to the success of autonomics; as Dave Patterson has reportedly said, "for better or worse, benchmarks shape a field." Without a good target, designers of autonomic systems will not be able to compare their systems against the state of the art in a reliable and reproducible way.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

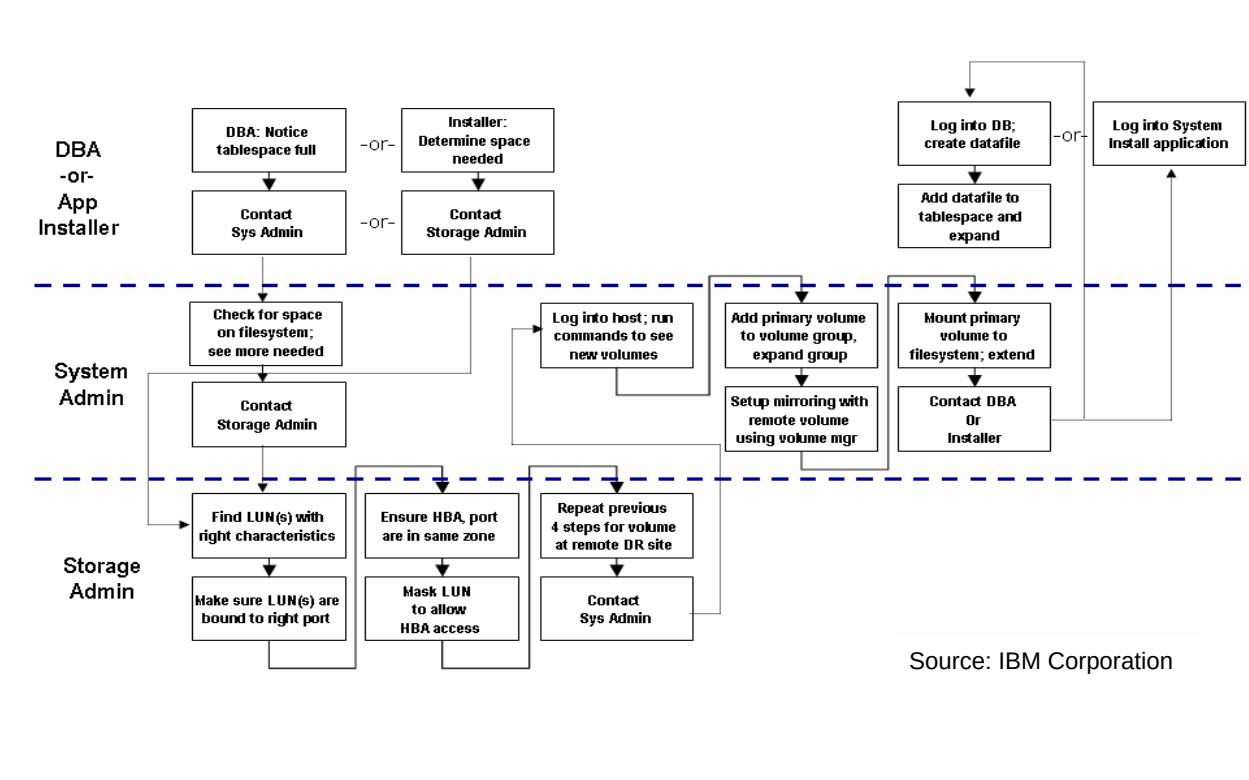
Further, good benchmarks should take into account the fact that systems change over time, and thus evaluate the ability for the autonomic technology under test to evolve as well. Heterogeneity and its impact should be an important axis of any autonomic benchmark suite.

There has been some early work in this area (see, for example, [Brown & Patterson]), but many questions remain. Perhaps the most important challenge to address is how to evaluate aspects of autonomic systems that relate to humans.

Study of processes and practices

"Furious activity is no replacement for understanding." — H.H. Williams

In order to automate aspects of systems management, we must first understand what it is that needs automation. In the figure below, one can see the task breakdown for the addition of storage underneath a database. The figure is surprisingly complex, and illustrates the need to comprehensively establish an understanding of the processes that managers typically must follow in order to achieve various ends.



Source: IBM Corporation

What makes this understanding difficult is that a single task may span multiple "administrative" domains (e.g., storage, database, and even network administrators). Hence, this type of study must strive to include not only storage experts but also the tasks of important applications (such as databases).

Thus, we believe there must be a serious study of the common processes that administrators typically enact to perform everyday storage system maintenance tasks. Such a study is by its very nature cross-disciplinary, as it spans boundaries of storage, network, and general system administrators.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

Management policies

"The statesman who yields to war fever must realize that once the signal is given, he is no longer the master of policy but the slave of unforeseeable and uncontrollable events." — Sir Winston Churchill

As Garth Gibson noted in his panel speech at FAST '03 [Gibson], there must be research not just into the mechanisms for storage automation, but also the policies. How does one write these policies so that they are robust and provide the type of management that is needed? Thus, a central challenge for future policy-based systems is not only to demonstrate a set of mechanisms but to develop and quantitatively analyze the effectiveness of a set of policies.

Many challenges arise. What type of language should be used to express these policies? What specific policies should be developed? Some early work from IBM provides some promising first steps [Uttamchandi et al.]. Further, how should we evaluate and compare different approaches? Given that many of these policies depend on the workloads of the system, understanding the types of workloads modern systems will be placed under will also be important.

Specialized storage systems

"Specialization is for insects." — Robert Heinlein

Most of the previous research directions deal with the difficulties of storage systems as they are currently built. Although pragmatic, this approach constrains in numerous ways. Therefore, we believe it is important to consider new ways of building storage systems as an approach to reducing the burdens of management.

Storage systems have long been built as general-purpose data engines. However, to reduce complexity and increase management, it may be worth investigating context-specific storage technologies. One example from industry is the EMC Centera, a content-addressable storage system that is intended for certain classes of applications (e-mail archival, medical imagery, etc.). We believe it is worth considering other classes of "specialized" storage systems; these systems may not be suitable in all environments, but can exploit application-specific knowledge to reduce the burdens of management (i.e., fewer knobs to tweak, application-specific failure modes, and the like).

Summary

Systems management remains a source of real problems that limit the scope of services provided, and is an area in need of real solutions. In this chapter, we have outlined a number of research directions that should be pursued in order to automate storage systems management and hence achieve the vision of autonomic computing. The main research directions we have identified are:

- Transparency: Techniques to expose and describe the decisions made by autonomic systems must be developed. If computer systems are to make important decisions, said systems must be able to explain their decision-making process to human administrators.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

- Devirtualization: Virtualization is commonplace, and yet it interferes with performance tuning and problem diagnosis. Tools that can discover the underlying physical nature of drives and larger-scale systems are required.

- Reliability: System reliability was signaled as perhaps the most important aspect of storage systems. Hence, further research into methods for building robust and reliable systems in the presence of complex and heterogeneous systems is needed.

- Evaluation and metrics: For a marketplace of autonomic technologies, standard benchmarks must be developed, to enable competing products and approaches to be rigorously compared.

- Study of processes and practices: To understand how to build automation technology, the actual processes that exist in today's systems must be studied.

- Management policies: As storage automation systems become commonplace, there must be research not only into the systems that enable policy-based automation, but research into the actual policies themselves.

- Specialized storage systems: As storage uses become more diverse and specialized, a different approach to building low-cost manageable systems is to specialize the storage systems to the needs of important application classes. Hence, further research into the design and implementation of such specialized systems is recommended.

One major point we make is that one can't take a "field of dreams" approach – if you build the technology, the users won't necessarily come. One must instead think through the "psychology" of autonomics, i.e., how to get existing system managers to want to use new automation technology.

As autonomic technology develops, we believe it will go beyond the simple desire to lower costs. Indeed, automation can lead to new business opportunities as systems become more agile in how they react and adjust to their current workload and environment.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

References

[Anderson, et al.] Eric Anderson, et al., "Hippodrome*:* Running circles around storage administration,*" FAST '02* (January 2002).

[Arpaci-Dusseau, et al.] Andrea Arpaci-Dusseau, et al., "Transforming Policies into Mechanisms with Infokernel," *SOSP '03* (October 2003).

[Brown & Patterson] Aaron Brown and David Patterson, "Towards Maintainability, Availability, and Growth Benchmarks: A case study of software RAID systems," *USENIX '00* (June 2000).

[Candea, et al.] George Candea, et al., "Microreboot - A technique for cheap recovery," *OSDI '04* (December 2004).

[Chen, et al.] Mike Chen, et al., "Path-based Failure Evolution and Management," *NSDI* (March 2004).

[Denehy, et al.] Timothy E. Denehy, et al., "Deconstructing Storage Arrays,*" ASPLOS XI* (October 2004).

[Forest, et al.] Stephanie Forrest, Steven Hofmeyr, and Anil Somayaji, "Computer Immunology," *Communications of the ACM*, **40**, 10, (1997).

[Gibson] Garth Gibson, *Panel Presentation at FAST '03*.

[Gray] Jim Gray, "What Next? A dozen remaining IT problems," *Turing Award Lecture, FCRC* (May 1999).

[Kephart & Chess] Jeffrey Kephart and David Chess, "The Vision of Autonomic Computing," *Computer*, **36**, 1, (January 2003).

[Lampson] Butler Lampson, "Computer Systems Research – Past and Future," *Keynote Address, SOSP 17* (December 1999).

[Litzkow & Livny] Michael Litzkow and Miron Livny, "Experience With the Condor Distributed Batch System," *Proceedings of the IEEE Workshop on Experimental Distributed Systems* (October 1990).

[Patterson] David Patterson, "Availability and Maintainability >> Performance: A New Focus for a New Century," *Keynote Talk in FAST '02* (January 2002).

[Patterson et al.] David Patterson, Garth Gibson, and Randy Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *SIGMOD '88*.

[Schindler et al.] Jiri Schindler, et al., "Track-aligned Extents: Matching access patterns to disk drive characteristics," *FAST '02* (January 2002).

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

[Talagala et al.] Nisha Talagala, Remzi Arpaci-Dusseau and David Patterson, "Microbenchmark-based Extraction of Local and Global Disk Characteristics," *UCB Technical Report CSD-99-1063* (1999). Available at http://www.cs.wisc.edu/~remzi/Postscript/disk.pdf as of 12/10/04.

[Uttamchandi et al.] Sandeep Uttamchandani, et al., "Polus: Growing storage QoS management beyond a 'four-year old kid'," *FAST '04* (April 2004).

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

LIST OF ACRONYMS

A/V	**A**udio/**V**ideo (or **A**udio/**V**isual)
AI	**A**rtificial **I**ntelligence
ANS	**A**utonomic **N**ervous **S**ystem
ANSI T10	**A**merican **N**ational **S**tandards **I**nstitute **T10** Technical Committee
API	**A**pplication **P**rogramming **I**nterface
AS	**A**dvanced **S**witching
ASP	**A**pplication **S**ervice **P**rovider
ASPLOS	International Conference on **A**rchitectural **S**upport for **P**rogramming **L**anguages and **O**perating **S**ystems
ATA	**A**dvanced **T**echnology **A**ttachment
D2D	**D**isk to ("**2**") **D**isk
DB	**D**ata **B**ase
DBA	**D**ata **B**ase **A**dministrator
DMA	**D**irect **M**emory **A**ccess
DNS	**D**omain **N**ame **S**erver/**S**ervice
DoS	**D**enial **o**f **S**ervice(s)
DR	**D**isaster **R**ecovery
DS2	**D**ata **S**torage **D**evices and **S**ystems
ECC	**E**rror **C**orrecting **C**ode (or **E**rror **C**orrection **C**ode)
ErCC	**Er**asure **C**orrecting **C**ode
FAST	Conference on **F**ile **a**nd **S**torage **T**echnologies
FC	**F**ibre **C**hannel
FCRC	**F**ederated **C**omputing **R**esearch **C**onference
GB	**G**iga**b**yte (10^9 bytes or 2^{30} bytes)
HBA	**H**ost **B**us **A**dapter
HDD	**H**ard **D**isk **D**rive
HDF	**H**ierarchical **D**ata **F**ormat
HIPAA	**H**ealth **I**nsurance and **P**ortability and **A**ccountability **A**ct
HSM	**H**ierarchical **S**torage **M**anagement
I/O	**I**nput/**O**utput
IDA	**I**nformation **D**ispersal **A**lgorithm
IETF	**I**nternet **E**ngineering **T**ask **F**orce
ILM	**I**nformation **L**ifecycle **M**anagement
InfiniBand/AS	**Infini**te **Band**width/**A**dvanced **S**witching
INSIC	**In**formation **S**torage **I**ndustry **C**onsortium
IP	**I**nternet **P**rotocol
IR	**I**nformation **R**etrieval
IRAM	**I**ntelligent **R**andom **A**ccess **M**emory
ISCA	**I**nternational **S**ymposium on **C**omputer **A**rchitecture
iSCSI	**I**nternet **SCSI**
IT	**I**nformation **T**echnology
JCDL	**J**oint **C**onference on **D**igital **L**ibraries
LDAP	**L**ightweight **D**irectory **A**ccess **P**rotocol
LUN	**L**ogical **U**nit **N**umber
MAID	**M**assive **A**rray of **I**dle **D**isks
MB	**M**ega**b**yte (10^6 bytes or 2^{20} bytes)
MB/sec	**M**ega**b**ytes per **sec**ond
MIME	**M**ultipurpose **I**nternet **M**ail **E**xtensions
.mpg	**MP**E**G** format file
MP3	**MPEG** layer- **3** audio
MPEG	**M**oving **P**icture **E**xperts **G**roup

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited

MTTR	**M**ean **T**ime **t**o **R**epair
NASD	INSIC's **N**etwork-**A**ttached **S**torage **D**evices Program
NFS	**N**etwork **F**ile **S**ystem
NFSv4	**N**etwork **F**ile **S**ystem **v**ersion **4**
NSDI	Symposium of **N**etworked **S**ystems **D**esign and **I**mplementation
OS	**O**perating **S**ystem
OSD	**O**bject-based **S**torage **D**evices
OSDI	Symposium on **O**perating **S**ystems **D**esign and **I**mplementation
PB	**P**eta**b**yte (10^{15} bytes or 2^{50} bytes)
PC	**P**ersonal **C**omputer
.pdf	**P**ortable **D**ocument **F**ormat file
QoS	**Q**uality **o**f **S**ervice(s)
RAID	**R**edundant **A**rray of **I**ndependent **D**isks (or **I**nexpensive **D**isks)
RAID5	**RAID** level **5**
RAM	**R**andom **A**ccess **M**emory
RDBMS	**R**elational **D**ata**b**ase **M**anagement **S**ystem
rDMA	**r**emote **D**irect **M**emory **A**ccess
RFID	**R**adio **F**requency **Id**entification
ROC	**R**ecovery-**O**riented **C**omputing
SAM	**S**CSI **A**rchitecture **M**odel
SAN	**S**torage **A**rea **N**etwork
SATA	**S**erial **A**dvanced **T**echnology **A**ttachment
SCSI	**S**mall **C**omputer **S**ystem **I**nterface
SCTP	**S**tream **C**ontrol **T**ransmission **P**rotocol
SDET	**S**oftware **D**evelopment **E**nvironment **T**hroughput benchmark
SIGMOD	**S**pecial **I**nterest **G**roup on **M**anagement **o**f **D**ata (Conference)
SOSP	**S**ymposium on **O**perating **S**ystems **P**rinciples
SQL	**S**tructured **Q**uery **L**anguage
SRB	**S**torage **R**esource **B**roker
SSP	**S**torage **S**ervice **P**rovider
SWIFT	**S**ociety for **W**orldwide **I**nterbank **F**inancial **T**elecommunication
TB	**T**era**b**yte (10^{12} bytes or 2^{40} bytes)
TCO	**T**otal **C**ost of **O**wnership
TCP/IP	**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol
TPC-D	**T**ransaction **P**rocessing Performance **C**ouncil, Benchmark **D**
UML	**U**nified **M**odeling **L**anguage
UVC	**U**niversal **V**irtual **C**omputer
VLDB	**V**ery **L**arge **D**ata **B**ase Conference
VPN	**V**irtual **P**rivate **N**etwork
WAN	**W**ide **A**rea **N**etwork
XBRL	E**x**tensible **B**usiness **R**eporting **L**anguage
XML	E**x**tensible **M**arkup **L**anguage

TRADEMARKS

iPod® is a registered trademark of Apple Computer, Inc.
InfiniBand™ is a trademark and service mark of the InfiniBand (SM) Trade Association.
Centera™ is a trademark of EMC Corporation.
Teradata™ is a trademark of NCR.

All other trademarks, if any, are the property of their respective owners.

© Information Storage Industry Consortium – All Rights Reserved
Reproduction without Permission is Prohibited