The evolution of storage systems

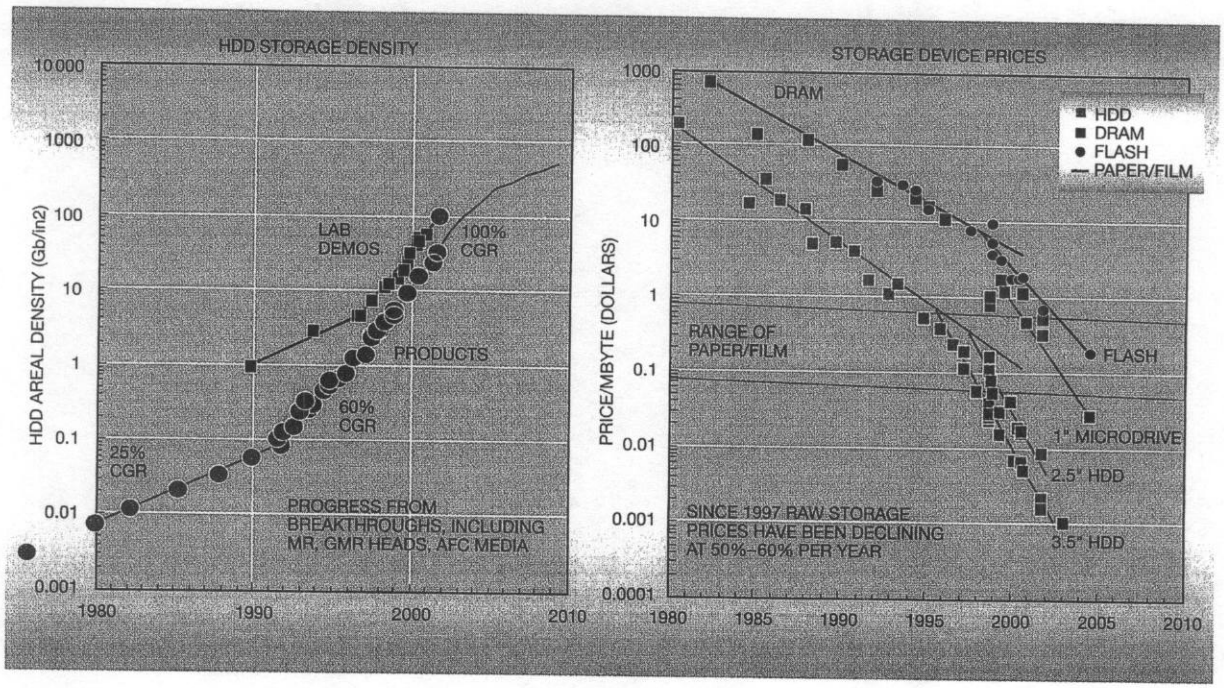
by R. J. T. Morris
B. J. Truskowski

Storage systems are built by taking the basic capability of a storage device, such as the hard disk drive, and adding layers of hardware and software to obtain a highly reliable, high-performance, and easily managed system. We explain in this paper how storage systems have evolved over five decades to meet changing customer needs. First, we briefly trace the development of the control unit, RAID (redundant array of independent disks) technologies, copy services, and basic storage management technologies. Then, we describe how the emergence of low-cost local area data networking has allowed the development of network-attached storage (NAS) and storage area network (SAN) technologies, and we explain how block virtualization and SAN file systems are necessary to fully reap the benefits of these technologies. We also discuss how the recent trend in storage systems toward managing complexity, ease-of-use, and lowering the total cost of ownership has led to the development of autonomic storage. We conclude with our assessment of the current state-of-the-art by presenting a set of challenges driving research and development efforts in storage systems.

The first data storage device was introduced by IBM in 1956. Since then there has been remarkable progress in hard disk drive (HDD) technology, and this has provided the fertile ground on which the entire industry of storage systems has been built. *Storage systems* are built by taking the raw storage capability of a storage device such as the HDD and by adding layers of hardware and software in order to obtain a system that is highly reliable, has high performance, and is easily manageable. Storage systems are sometimes referred to as *storage subsystems* or *storage devices* (although *device* is better used to describe the raw storage component or an elementary storage system). Originally, the storage system was just the HDD, but over time storage systems have developed to include advanced technologies that add considerable value to the HDD. Storage systems have evolved to support a variety of added services, as well as connectivity and interface alternatives. It is for this reason that file systems and storage management systems are often considered parts of a storage system and thus will be briefly treated in this paper.

To understand the evolution of storage systems, it is important to observe the evolution of the HDD. The areal density of the HDD has improved by seven orders of magnitude, and this has resulted in a reduction of the floor space taken by the corresponding storage systems also by about seven orders of magnitude. Figure 1 plots the HDD areal density (on the left) and the price of various storage devices (on the right) since 1980. HDD prices have decreased by about five orders of magnitude since 1980, while the cost of storage systems has fallen about 2.5 orders

Figure 1 HDD storage density is improving at 100 percent per year (currently over 100 Gbit/in2). The price of storage is decreasing rapidly and is now significantly cheaper than paper or film.



of magnitude in the same period.[1] The sharper fall in the price of the HDD implies that the cost of the raw HDD accounts for a progressively smaller fraction of the total cost of the storage system—we will return to this crucial observation later.

Although Moore's law tells us that the number of transistors per unit area of silicon doubles every 1.5 years, we see from Figure 1 that the number of bits stored per unit of HDD media is doubling about every year! But more important than improvements in device density or cost have been the new applications that have been enabled by these advances. On the time line in Figure 1, two milestones stand out. In 1996, digital storage became more cost-effective for storing data than paper, and, in 1998, we reached the point where film used in medical radiology could be economically supplanted by electronic means. Another important milestone, this one in the consumer market, was reached several years ago when it became cost effective to store video content using digital storage systems. Soon after we saw the emergence of HDD-based set-top-box devices (sometimes called personal video recorders) that offered improved management of entertainment video in the home.

Besides the emergence of storage systems that enable digitization and replacement of legacy media, it is instructive to consider how architectural outcomes are affected by the *relative* progress of technologies. Figure 2 portrays the relative advances in storage, processor, and communications technologies, obtained by plotting the cost/performance improvement of widely available technologies in end user products as documented by personal computer (PC) magazines since 1983. (Broadband to the home is considered to have limited availability at present.) The plot shows that since 1990, storage technology has outrun both communications and processor technologies. In fact the availability of inexpensive digital storage has influenced the architectures that we see in place today. For example, because storage technology has become relatively inexpensive while deployment of point-to-point broadband to homes has been slow, HDD-based set top boxes are more prevalent than video-on-demand. The parallel story of technology in the enterprise is not shown but leads to a similar conclusion: the amount of stored data has outrun the ability of communications and processing systems to provide easy access to data. Thus, we see widespread use of multiple copies of data (e.g., Lotus Notes* replication, widespread internet

caching) as well as the deployment of storage systems close to the end user in order to avoid network delays.

We have already pointed out that the cost of the HDD accounts for a progressively smaller component of the total cost of a storage system. In fact, in recent years the game has changed not once, but twice. First, the HDD has changed from a differentiating technology to a commoditized component in the typical storage system. As shown in Figure 6 of Reference 1, the HDD components within commercially available medium-to-high-function storage systems typically cost less than 10 percent of the cost of the system. Customer value has migrated to "advanced functions" and the integration of these functions within the system itself. Then, as Figure 3 shows, the cost of managing storage now dominates the total cost of a storage system.[2–4] This means that the value to the customer of a storage system now resides in its ability to increase function beyond what is provided in the bare HDD, and specifically in its ability to lower management costs and provide greater assurances as to the availability of data (e.g., through backup and replication services). Buyers of storage systems are now mainly buying the function that is embodied in the software (sometimes called the firmware) of the storage system. Furthermore, buyers of storage systems are "discounting" the initial purchase price in the buying criteria and weighing the total cost of ownership more heavily.

In the present issue of the *IBM Systems Journal*, we have included papers that document the ongoing evolution of some key storage system technologies. In the rest of this paper, we first discuss how these new capabilities were driven by technological developments, such as local area networking, and by the IT (information technology) requirements for the enterprise. Then we show how today's challenges in the industry are evolving into the challenges of the future.

Figure 2 Improvement factors for PC technologies: since 1990 storage technology has outpaced both processor and communication technologies.

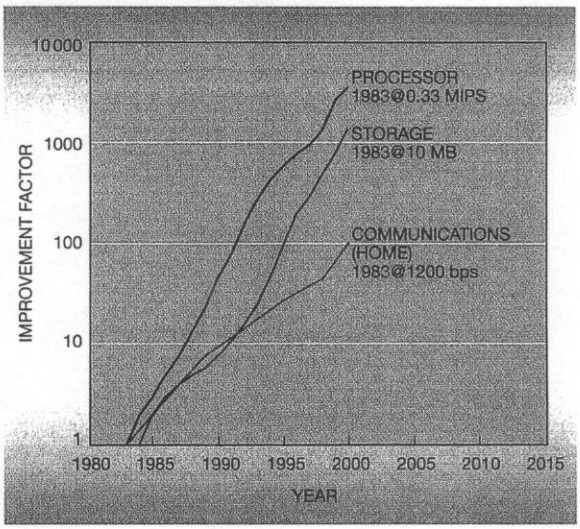
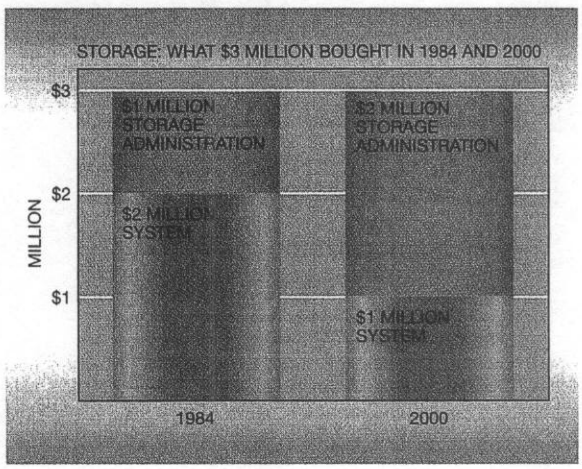


Figure 3 Storage administration costs now dominate purchase costs for a storage system.



Storage systems come of age: from components to systems

It has long been recognized that the disk drive alone cannot provide the range of storage capabilities required by enterprise systems. The first storage devices were directly controlled by the CPU. Although some advances did take place in the interim, System/360* in 1964 was the first offering of advanced functions in an external storage control unit.[5] The key advantage of a control unit (or controller) was that the I/O commands from the CPU (sometimes called the host) were independently translated into the specific commands necessary to operate the HDD (sometimes called the direct access storage device, or DASD), and so the HDD device itself could be managed independently and asynchronously from the CPU. The control unit included buffering that allowed

the CPU and HDD operations to overlap. Over time there was a greater emphasis on availability, and controllers began to support multiple data paths from CPUs to storage controllers. The IBM 3990 Model 3 storage controller consisted of two clusters on separate power and service boundaries and included both a large cache and nonvolatile storage (NVS) memories. The caches were used to improve read response time, whereas the NVS was used to provide a fast write and dual copy capability. In addition, it was possible to perform maintenance on the cluster, cache, or NVS under normal operation.[6]

Storage systems leapt further ahead in the early 1990s when RAID (redundant array of independent disks) technology was introduced. RAID allowed the coordination of multiple HDD devices so as to provide higher levels of reliability and performance than could be provided by a single drive. The emergence of smaller form factor drives (5.25 inches and then 3.5 inches) also encouraged the design of systems using a larger number of smaller drives, a natural fit with RAID technology. The classical concept of parity was used to design reliable storage systems that continued to operate despite drive failures. Parallelism was used to provide higher levels of performance. RAID technology was delivered in low cost hardware and by the mid 1990s became standard on servers that could be purchased for a few thousand dollars. Many variations on RAID technology have been developed; see the survey in Reference 7. These were used in large external storage systems that provided significant additional function, including redundancy (no single point of failure in the storage system) and copy services (copying of data to a second storage system for availability).

Disaster recovery became a requirement for all IT systems, and impacted the design of storage systems as well. The required degree of protection of data requires a solution that may range in technical sophistication from occasional copies onto magnetic tape (and manually transported), through electronic versions of essentially the same principle, to true mirroring solutions and truly distributed systems. Several commonly employed techniques have emerged. A *point-in-time copy* (offered by IBM under the name FlashCopy*) is the making of a consistent virtual copy of data as it appeared at a single point in time. This copy is then kept up to date by following pointers as changes are made. If desired, this virtual copy can, over time, be made into a real copy through physical copying. A second technique, *mirroring* or *continuous copy* (offered by IBM under the name Peer-to-Peer Remote Copy) involves two mirror copies of data, one at a primary (local) site and one at a secondary (recovery) site. We say this process is *synchronous* when data must be successfully written at the secondary system before the write issued by the primary system is acknowledged as complete. Although synchronous operation is desirable, it is practical only over limited distances (say, of the order of 100 km). Therefore, other asynchronous schemes, and other significant optimizations, are used to improve the performance of these basic schemes. For a complete discussion of the emerging requirements and technologies for disaster recovery see References 8 and 9.

The requirements for data availability were not completely satisfied by reliable storage systems, even with redundant instances of hardware and data. Because data could be accidentally erased (through human error or software corruption), additional copies were also needed for backup purposes. Backup systems were developed that allowed users to make a complete backup of selected files or entire file systems. The traditional method of backup was to make a backup copy on tape, or in the case of a personal computer, on a set of floppy disks or a small tape cartridge. However, as systems became networked together, LAN-based backup systems replaced media-oriented approaches, and these ran automatically and unattended, often backing up from HDD to HDD. Backup systems are not as simple as they sound, because they must deal with many different types of data (of varying importance), on a variety of client, server, and storage devices, and with a level of assurance that may exceed that for the systems they are backing up. For reasons of performance and efficiency, backup systems must provide for incremental backup (involving only those files that have changed) and file-differential backup (involving only those bytes in a file that have changed). These techniques pose an exceptionally stringent requirement on the integrity of the meta-data that are associated with keeping straight the versions of the backed-up data. In fact, to obtain the needed assurances and performance, the most advanced database recovery technology must be used. The first systems to provide these types of capabilities (including incremental backup) are documented in Reference 10. File-differential backup was subsequently introduced, in which only the changed bytes within a file are sent and managed at the backup server. Since tape still provides the most cost-effective form of saving

backed up data, there are a number of special considerations relating to the append-only nature of tape devices, and these need to be explicitly designed into a backup system. Furthermore, when storage is connected using a SAN (described later), additional performance improvements are possible through bypassing the LAN (LAN-free backup) or the server (server-free backup). These technologies are explained in detail in a paper on Tivoli Storage Manager (TSM) in this issue.[11] A technique that can supplement the backup approach involves making a point-in-time copy as previously described, and using that virtual copy as the backup copy. If and when a physical copy is needed, a backup of the virtual copy can be performed.

Although the raw cost of HDD storage has declined, tertiary media such as tape or optical disks continue to remain important, and therefore hierarchical storage systems that manage these levels of storage are needed. Many of the requirements of hierarchical storage are already dealt with in TSM, because the backup task requires managing across a hierarchy of storage devices, especially disk and tape systems, and involves dealing with the constraints of tape access. However, the criteria for managing a hierarchy in some application environments (e.g., content management systems) do not always coincide with those appropriate for backup applications, and, as a result, further capabilities have been developed.[11]

Besides backup applications, tape storage plays an important role in data center operations, holding files that have been automatically migrated by a hierarchical storage management system (such as DFSMShsm*[12]) and master data sets (used, for example, in credit card processing). All these applications exploit the high sequential bandwidth of tape in batch processing. By virtue of the way tapes are managed, tapes are often vastly underutilized. In fact, because of software constraints and a drive for simplicity, it has been common practice in mainframe applications to store a single file on a tape. This low tape utilization motivated the development of a tape virtualization technology (used in the IBM Virtual Tape Server product) involving a disk cache in front of a tape library, and independently treated files that are cached on disk while in use by the application, and later unloaded and packed onto tape. Thus, by shielding the application from the physical mapping onto the tape drive, significant improvement in performance, cost, and manageability is achieved.

Networked storage

The emergence of low-cost LAN technology drove the most significant trend of the late 1980s and early 1990s in storage systems. PCs became networked and the client/server computing model emerged. While some control of applications migrated from the data center to the PC, key data often had the status of a corporate or institutional resource, rather than a personal resource, and therefore needed to be shared and safeguarded. The PC was unmanaged and notoriously unreliable, and so to achieve sharing of data, rudimentary low-cost PC-class storage servers became common. These systems were more "mission-critical" than the PC-based client, and were thus prime candidates for technologies such as file-serving, RAID, and LAN-based backup systems. The software used for networking was frequently Novell NetWare** or other software available from PC software vendors. At the same time, UNIX** enjoyed a resurgence both in UNIX workstations (an alternative to the PC client) and in UNIX servers. The widespread availability of the NFS** (Network File System) file-sharing protocols caused further specialization and the emergence of file servers. The next step was the emergence of NAS (network-attached storage) systems, which bundled the network file serving capability into a single specialized box, typically serving standard protocols such as NFS, CIFS (Common Internet File System), HTTP (HyperText Transfer Protocol), and FTP (File Transfer Protocol). NAS systems were simple to deploy because they came packaged as "appliances," complete with utilities and management functions.

At the same time, IT organizations were attempting to "regain control" over the dispersed assets characteristic of client/server computing. The data center took on a renewed importance in most enterprises. To that end, multiple servers in a machine room sought the capability to access their backend storage without necessarily having individual storage directly attached and therefore dedicated to an individual server. This caused the emergence of storage area networks (SANs). SANs had the advantage that storage systems could be separately engineered from the servers, allowing the pooling of storage (statically configured) and resulting in improved efficiencies and lower risks for the customer (storage investment was not tied to a particular server hardware or software choice). SAN technology opened up new opportunities in simplified connectivity, scalability, and cost and capacity manageability. Fibre Channel became the predominant networking technology[13] and large

storage systems, such as IBM TotalStorage* Enterprise Storage Server*[14] (ESS), support this protocol and use it to attach multiple servers. To avoid confusion, the reader should think of NAS systems as working with files and file access protocols, whereas a SAN enables access to block storage (the blocks of data may be stored on a storage system or an HDD). This distinction is illustrated in Figure 4 which shows the data paths for direct-attached storage (A), SAN-attached storage (B), network-attached storage (C), and a mixed NAS and SAN environment (D). Appliances having both SAN and NAS interfaces are available on the market; the IBM Storage Tank*, described later, has this capability.

The widespread adoption and commoditization of Ethernet LANs running TCP/IP (Transport Control Protocol/Internet Protocol) has caused increasing interest in the use of this technology in the SAN. The unifying of networking under TCP/IP and Ethernet offers the benefits of standardization, increased rate of innovation, and lower costs, in both hardware and device support. Management costs are reduced since staffs need to be familiar with only one type of network, rather than two. The iSCSI (Internet Small Computer System Interface) standard has been introduced to allow the SCSI protocol to be carried across a TCP/IP network.[15] However, in order to realize the benefit, considerations such as discovery, performance, and security must be addressed. The resolution of these concerns is discussed in detail in Reference 16.

Although the availability of networked storage provides improved access to data, it still leaves some key issues unaddressed. SANs enable arbitrary connectivity between using system and storage, but they still pose operational problems. Although a SAN may not be hard-wired to its attached storage, it is still "soft-wired" in the sense that the configuration is typically static, and changes cannot be made to the attached storage or using system without disruption. The addition of a layer of indirection between the using system and storage, provided either through a hardware switch or an intermediary software-based system, allows the using system (typically a server) to deal with nondisruptive changes in the storage configuration. Additionally, virtual volumes can be created that cut across multiple storage devices—this capability is referred to as *block virtualization*. Block virtualization allows all the storage on a SAN to be pooled and then allocated for access by the using systems.[17] This also simplifies various storage management tasks, such as copy services, varying storage devices on-line or off-line, and so on.

But access is not enough, because the using systems often assume sole access to data and are not equipped to share data or free space they own with other systems. Thus, each system is allocated its own supply of free space, a wasteful scheme. These problems can be overcome by removing the task of meta-data management from each of the client systems and creating a global capability that manages the mapping, sharing, and free-space management of storage on all attached storage devices. Besides simplifying configuration and free-space management, new capabilities are added such as sharing of data, nondisruptive change, and automation of many storage management functions as described later. This concept is referred to as a *SAN file system*. Because the meta-data are under control of a meta-data server, new capabilities are made possible, and a wide range of services are introduced that simplify management and improve the availability of data. For example, policies associated with data determine how data are managed for performance, availability, security, cost, and so on. This is further described later under "Autonomic storage."

An early implementation of a SAN file system is the IBM Storage Tank technology, illustrated in Figure 5 and described in Reference 18. An implementation of a quite similar concept is CXFS**, developed by SGI.[19] Storage Tank presents a file system to the client system by installing code at the VFS (virtual file system) or IFS (installable file system) layer. This code intercepts all file system I/O and manages it according to meta-data, which it obtains (and keeps locally cached) from the meta-data servers. While Storage Tank is a SAN file system, it also can aid in NAS management, because its clients can be gateways that run NFS or CIFS code and thus provide NAS service to other clients.

Putting it all together, the left side of Figure 6 shows how storage systems leverage RAID to improve the basic functions of the HDD device, and exploit local area networking technology in a SAN to gain physical connectivity with a variety of storage devices. But the current evolution goes further, as shown on the right of Figure 6. Block virtualization provides flexible logical-to-physical mapping across storage devices, and a common file system with separate and consolidated meta-data management allows dynamic policy-based resource management and added capabilities in a heterogeneous systems environment.

Figure 4 The data paths for direct-attached, SAN, and NAS storage

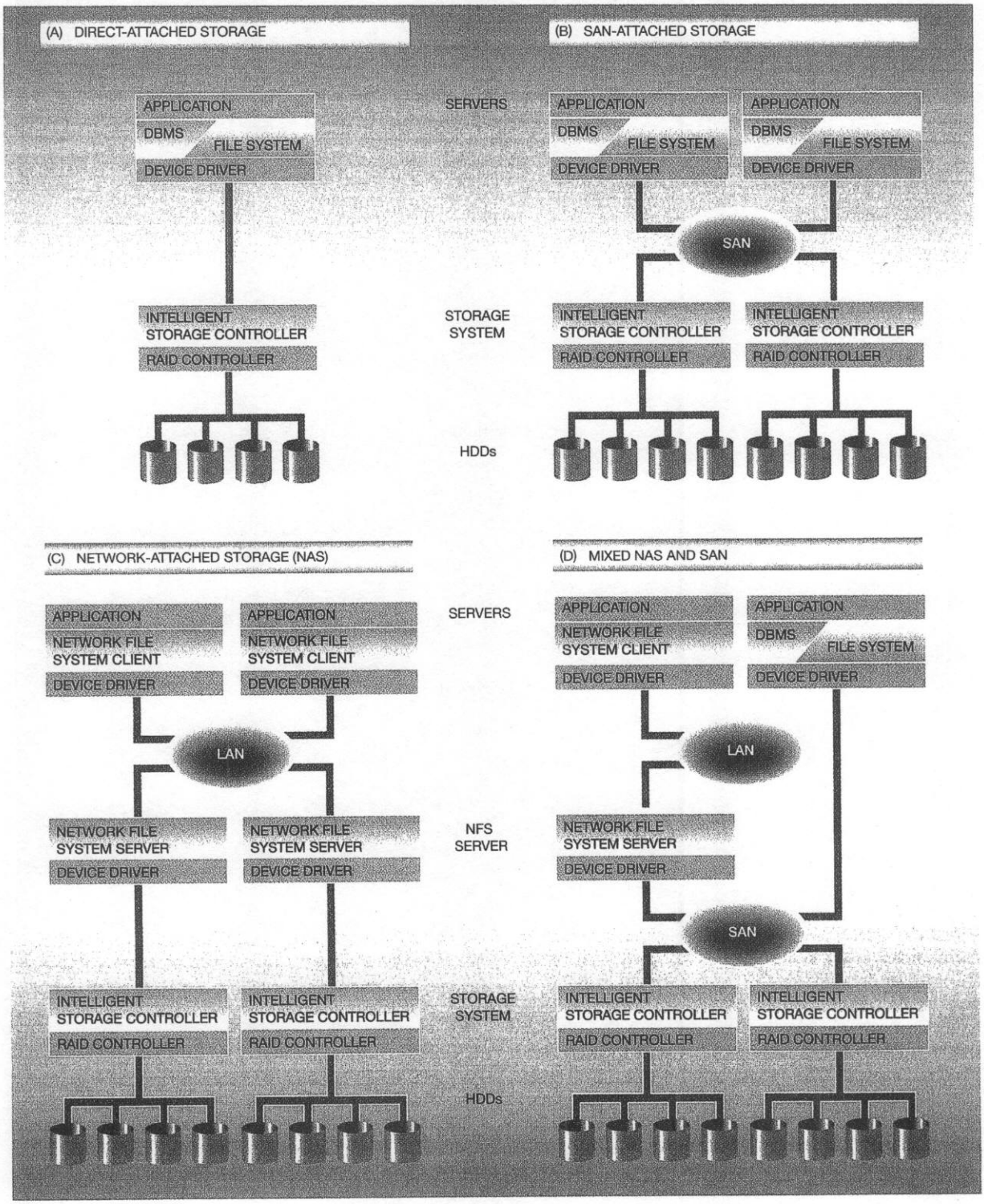
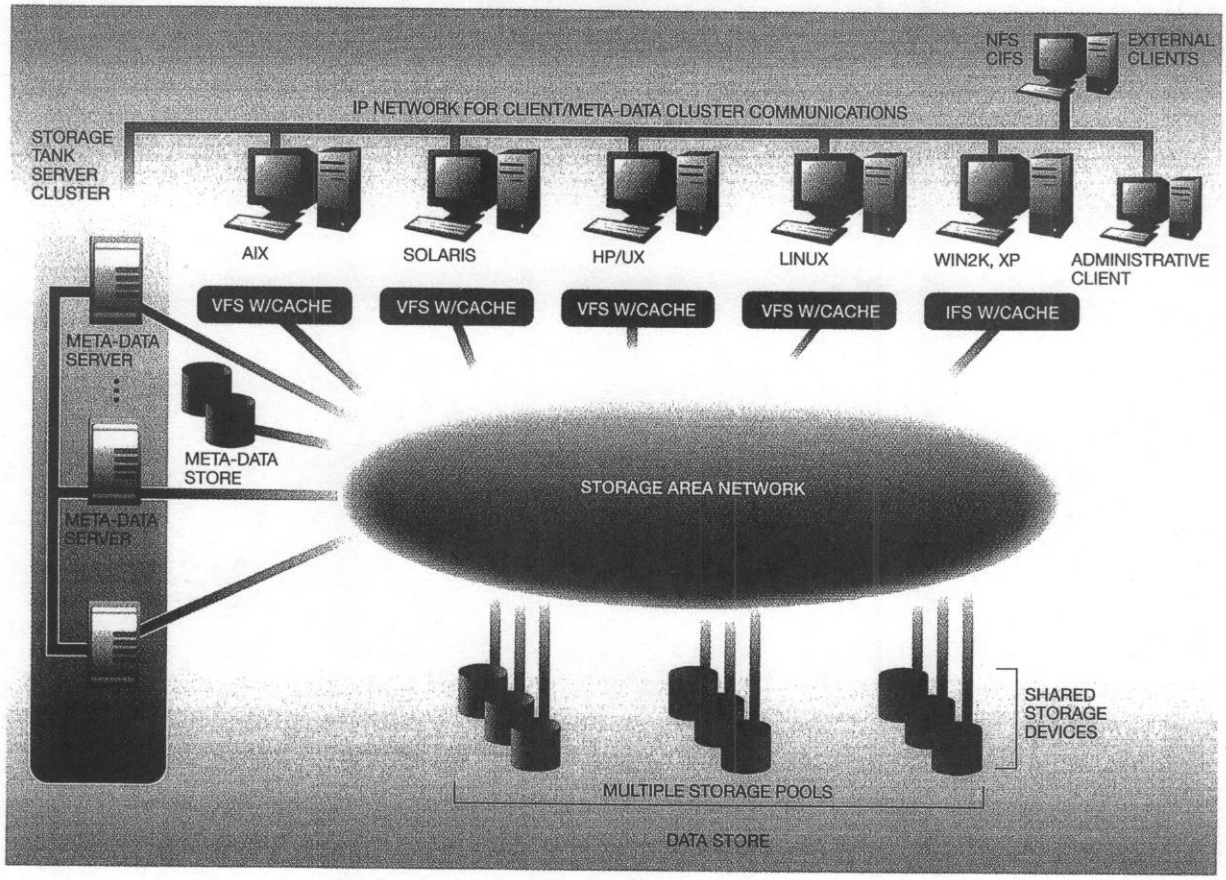


Figure 5 An example of a SAN file system: IBM Storage Tank file system



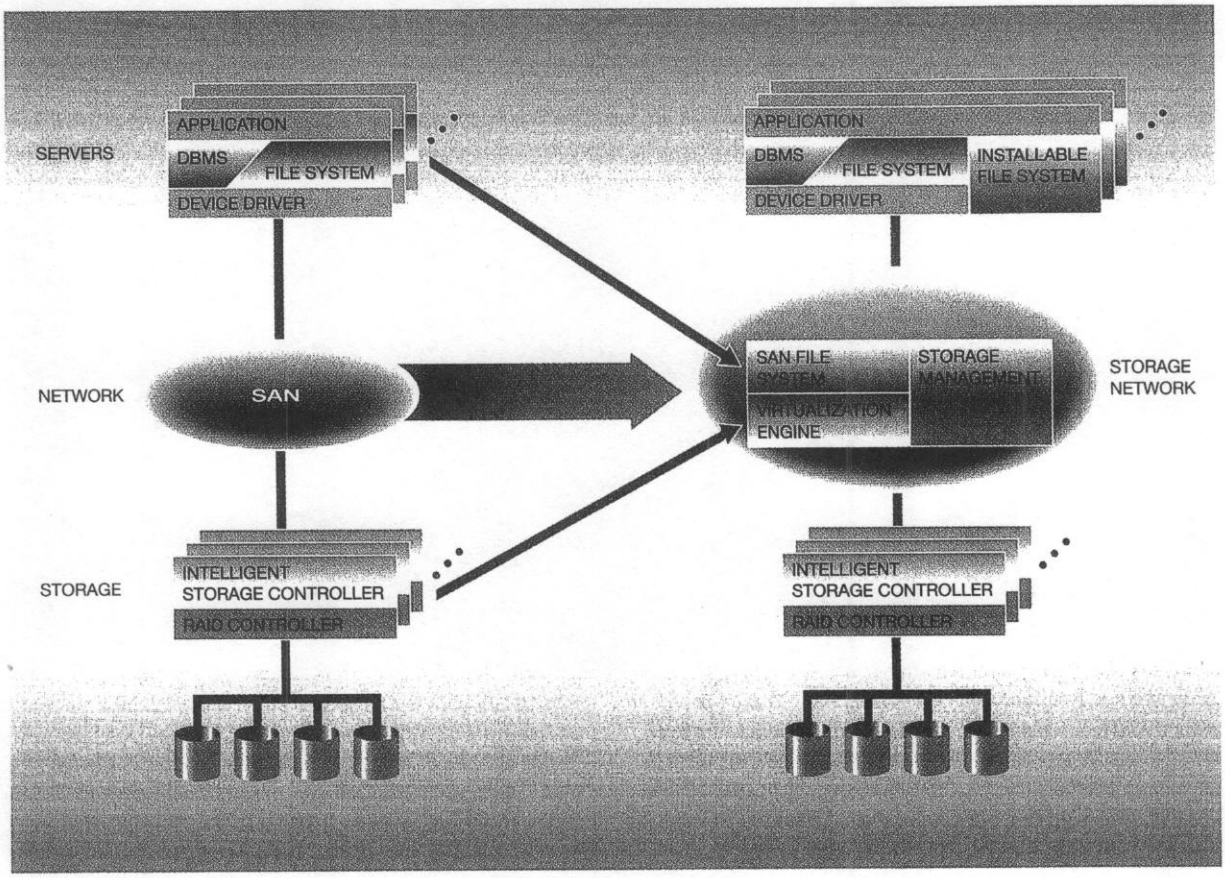
Systems management tools exploit these capabilities in order to facilitate the resource management task of the administrator.

Autonomic storage

Recognizing the fact that progress in base information technology had outrun our systems management capabilities, in 2001 IBM published the Autonomic Computing Manifesto.[20] The manifesto is a call to action for industry, academia, and government to address the fundamental problem of ease and cost of management. The basic goal is to significantly improve the cost of ownership, reliability, and ease-of-use of information technologies. The cost factor has already been highlighted in Figure 3; we return later to the issue of increasing reliability needs. To achieve the promise of autonomic computing, systems need to become more *self-configuring*, *self-healing* and *self-protecting*, and during operation, more *self-optimizing*. These individual concepts are not new, but the need for their deployment has dramatically increased as operational costs have increased and our dependence on systems heightened. The increased focus of autonomic computing is on implementing self-configuring, self-healing, self-protecting, and self-optimizing capabilities not just at the component level, but holistically, at a global level.

Interestingly, storage systems have incorporated autonomic computing features at the component level for some time. RAID, as described earlier, is an excellent example of a self-healing system. Further, high-function storage systems such as IBM's ESS have numerous autonomic computing features.[14] It is instructive to think about autonomic computing at three levels.

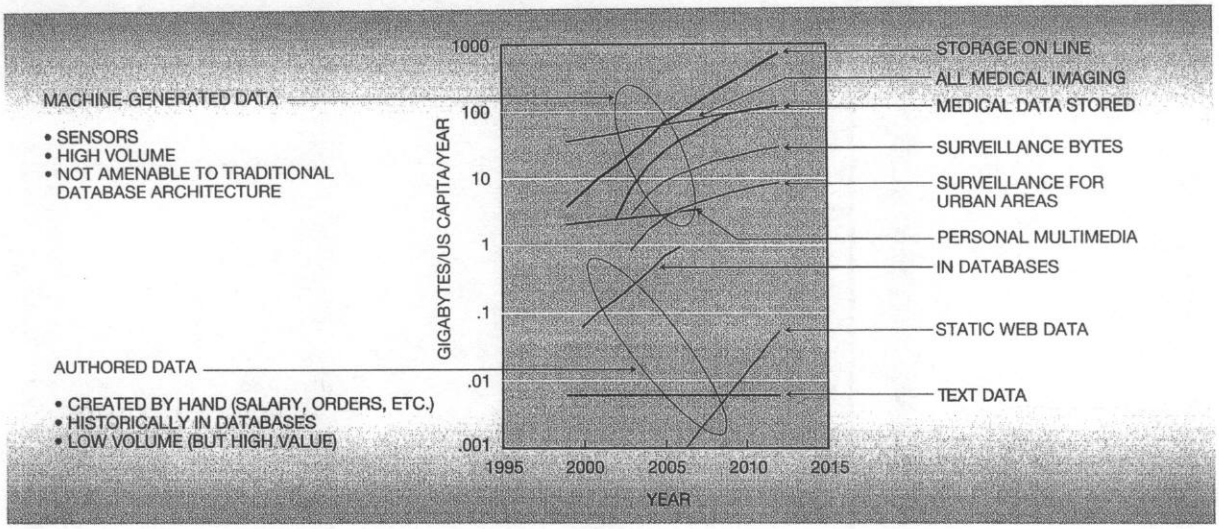
Figure 6 The evolution of storage networks from added physical capabilities to flexible administration of the storage resource



The first level is the *component level* in which components contain features which are self-healing, self-configuring, self-optimizing, and self-protecting. At the next level, *homogenous* or *heterogeneous systems* work together to achieve the goals of autonomic computing. As an example, consider the copy services described earlier, when systems at distant sites can replicate data on a primary system in order to provide continuous operation. Another example is the adaptive operation within the Collective Intelligent Brick systems, now being researched at IBM,[21] that allow "bricks" within a structure to take over from one another, or help each other carry a surge of load. There is an interesting side benefit in autonomic computing capabilities that use redundancy and reconfiguration in case of failure. Systems behaving this way can be designed and packaged for "fail-in-place" operation. This means that failing components (or bricks) need not necessarily be replaced; they can be worked around. Using this capability we can package storage in three dimensions rather than two, because failing bricks in the interior of the structure do not have to be replaced.

At the third and highest level of autonomic computing, heterogeneous systems work together toward a *goal* specified by the managing authority. For example, a specified class of data is assigned certain attributes of performance, availability, or security. The data specification might be based on a file subtree, a file ending, file ownership, or any other characteristic that can define a subset of data. These ideas owe their heritage to the pioneering concepts of system-managed storage.[2] Because it intercepts all I/Os and because it uses a separate control network to access the central repository of meta-data, the Storage Tank SAN file system has the capability to perform at this highest level of autonomic computing.[18] It can

Figure 7 Machine-generated data from a variety of sources are increasing at an exponential pace and dominating manually authored data.



achieve this while obeying certain *constraints* on the data, for example, location, security policies in force, or how the data may be reorganized. Given the virtual nature of data and because of the existence of a separate meta-data manager, each file can be associated with a *policy* that describes how it is to be treated for purposes of performance, security, availability, and cost. This composite capability of autonomic systems working toward goals, while adhering to constraints, is referred to as *policy management*. Thus Storage Tank can arrange that files be placed on high-performance block storage systems with QoS (Quality-of-Service) management when needed, or on highly secure storage with encryption and access control in place, or on high availability systems that are replicated across remote sites and have diverse access paths.

Future challenges

Certain known requirements for storage systems are accelerating and new paradigms are emerging that provide additional challenges for the industry. The volume of data being generated continues to increase exponentially, but a new source of data is becoming increasing prevalent: machine-generated data. Until now most data have been authored by hand, using the keyboard, and widely distributed and replicated through networking. This includes data in databases, textual data, and data on Web pages. However, these types of data are now being dwarfed by machine-generated data, that is, data sourced by digital devices such as sensors, surveillance cameras, and digital medical imaging (see Figure 7). The new data require extensive storage space and nontraditional methods of analysis and, as such, provide new challenges to storage and data management technologies. Effective data mining may provide important new opportunities in security, customer relationship management, and other application areas.

Another important trend is in the growth of reference data. Reference data are commonly defined as stored data that are only infrequently retrieved (if at all). As an example, consider the archived copies of customer account statements—these are rarely accessed after an initial period. This phenomenon is also driving new directions in research, such as implementing reliable storage systems from low-cost (e.g. ATA) drives.

These trends in the growth of stored data are driving new challenges in storage subsystems and the management of data. The traditional methods of accessing data through a location or hierarchical file name are being reexamined in an attempt to design new access methods in which data may be used wherever the data reside, based on content or some other attribute. New crawling, search, and indexing technologies are emerging to address that challenge, and

new storage system requirements are emerging to efficiently support those applications. Additional requirements follow from the scale of data and the new usage models that they must support. In some cases the repository for data must provide additional functions for the security and access control for those data. An interesting development that addresses this need is the object storage technology of Reference 22, which allows access control at a much finer granularity than current methods for SANs.

We discussed disaster recovery earlier. A combination of trends is making the requirement of business continuance (the business continues to operate in the face of IT failures or disaster) more challenging. Pending SEC regulation, insurance company requirements, and concerns about national and international security are placing increased demands on storage architectures. The costs of down time for many enterprises have generally been acknowledged to be large and increasing. Systems can no longer be taken down for purposes of backing up data. As shown in Reference 1, whereas disk drive capacity has been almost doubling every year, seek time has only been improving about 12 percent per year and transfer rate about 40 percent per year. Hence it is not surprising that times to complete the backup have increased to the point where backup cannot be carried out "off shift," and in many cases there is no off shift. This means that customers need to employ technologies where a point-in-time copy can be made and then a backup from this copy. Even that is not sufficient in some cases as, without special measures, the backup may not be completed before it is time to start the next one. While backup time is clearly a problem, an even more severe issue is restore time. In some cases data must be restored in order to resolve a service disruption, and, as a result, the mismatch of data volume and access rates becomes even more of a problem. Much research and development is underway to address these issues; new techniques are proposed in References 8, 9, and 11.

A related trend is the growing importance of higher levels of availability for storage systems. It is generally acknowledged that decreasing the down time (e.g., an increase of availability from 0.99999 to 0.999999) represents a significant engineering and operational challenge as well as additional expense. Traditional RAID systems (e.g., RAID 5) are reaching a point where they cannot support the higher levels of reliability, performance, or storage density required. Indeed, storage space on HDDs has outrun the rate at which data can be accessed, rebuild times have increased correspondingly, and, therefore, the likelihood of a damaging second failure during rebuild has increased. Second, although the probability of undetected write errors on HDDs is small, the massive increase in storage space will over time increase the likelihood that problems from these errors are encountered. New approaches to storage structures are being researched that rely on alternative coding schemes, methods of adaptively creating additional copies of data, and also super-dense packaging of disk drives and associated control circuitry.[21]

Another problem area of growing interest concerns the long-term preservation of data. This problem was effectively described by Rothenberg in Reference 23. Although paper is often decried as an inferior storage medium because it is susceptible to water and fire damage, it may well outlast our best electronic technologies—not because the media is not long lasting, but because the formats of our digital records are subject to change, and the devices and programs to read these records are relatively short-lived. This problem can be addressed in a number of ways, and the most viable solution is to create data in a form that is self-describing; that is, it comes with the data structures and programs needed to interpret the data, coded in a simple universal language, for which an interpreter can easily be created at some later time. This approach is described in Reference 24. The eventual resolution of this issue overlaps with the problems of scale above: there is no use for data that cannot be found or understood.

Coming full circle to the beginning of the storage industry, perhaps the most significant potential change after five decades may take place in the role of the HDD. The HDD still appears to have considerable life left in it, and although a slowing in the rate of progress is projected due to significant challenges, there is a widely held view that no alternative technology is likely to provide serious competition in the enterprise for about the next ten years.[25] Nevertheless, there is increased interest and activity in alternative storage devices.[26] We noted that HDD cost is now 10 percent or less than the cost of the system it goes into, but we also saw in Figure 1 that presently available alternative device technologies, such as semiconductor memory (DRAM [dynamic random-access memory] or Flash), are still about two orders of magnitude more expensive than the HDD, thus ruling them out as storage devices in enterprise storage systems. Alternatives to HDD based on radically new ideas or lower-cost manufac-

turing technologies will be needed to supplant the HDD. While these may first find application in pervasive or consumer devices, eventually they may be successfully applied in enterprise storage systems.

Although the above challenges are influencing the evolution of storage systems, the greatest need is for new technology that lowers management costs and improves the ease of use and dependability of storage systems.

Conclusion

Storage systems have become an important component of information technology. There are projections that many enterprises will routinely spend more on storage technology than on server technology. Over a period of 50 years, and building on a base of giddying advances in hard drive component technology, the storage game has shifted to one where "systems" technology is needed to meet new requirements and where storage systems technology is a key element in coping with the information overload and in getting management costs under control. IT users' assets are embodied in their data, and storage systems are evolving to allow increased exploitation and protection of those assets. The storage system is no longer just a piece of hardware, but a complex system in which significant function is implemented in software.

If we assess where we are in storage systems today, it is clear that we now have reliable component subsystems that can be flexibly and reliably interconnected into networked storage systems. Autonomic storage is here today and being extended to the next level, where storage components work together to get the job done, whatever the challenge—load variability, disasters, and so on. All these technologies are increasingly paying off in managing complexity (and therefore cost) where it is most needed—in the tasks for which people are responsible. We are on a track to further develop these advanced storage systems technologies. But storage is not the only part of the IT stack, and storage systems need to increasingly play their part in getting all the automation to work, so that a business can respond quickly, flexibly, and at much lower cost to a range of new challenges. Institutions are facing change at a new rate, whether it is coming from supply and demand, labor costs, changing customer preferences, or new levels of integration. These institutions need the ability to respond in real time, with all the resources at their disposal—including those of their employees, suppliers, partners, and distributors—and using all parts of the IT stack. This on demand requirement will cause the rate of innovation we have described in storage systems to continue to accelerate.

Acknowledgment

The authors thank Jayashree Subrhamonia for her help in assembling the special issue of the *IBM Systems Journal* and Alain Azagury for his careful reading of this paper and his many suggestions.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Novell, Inc., The Open Group, Sun Microsystems, Inc. or Silicon Graphics, Inc.

Cited references

1. E. Grochowski and R. D. Halem, "Technological Impact of Magnetic Hard Disk Drives on Storage Systems," *IBM Systems Journal* **42**, No. 2, 338–346 (2003, this issue).
2. J. P. Gelb, "System-managed storage," *IBM Systems Journal* **28**, No. 1, 77–103 (1989).
3. "Storage on Tap: Understanding the Business Value of Storage Service Providers," *ITCentrix, Inc.* (March 2001).
4. *Server Storage and RAID Worldwide*, SRRD-WW-MS-9901, Gartner Group/Dataquest (May 1999).
5. C. P. Grossman, "Evolution of the DASD Storage Control," *IBM Systems Journal* **28**, No. 2, 196–226 (1989).
6. J. Menon and M. Hartung, "The IBM 3990 Model 3 Disk Cache," *Proceedings of the Thirty-Third IEEE Computer Society International Conference* (Compcon), San Francisco, March 1988, IEEE, New York (1988), pp. 146–151.
7. P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: High Performance, Reliable Secondary Storage," *ACM Computing Surveys* **26**, No. 2, 145–185 (1994).
8. A. C. Azagury, M. E. Factor, and W. F. Micka, "Advanced Functions for Storage Subsystems: Supporting Continuous Availability," *IBM Systems Journal* **42**, No. 2, 268–279 (2003, this issue).
9. A. Azagury, M. Factor, W. Micka, and J. Satran, "Point-in-time Copy: Yesterday, Today and Tomorrow," *Proceedings of the Tenth Goddard Conference on Mass Storage Systems and Technologies/Nineteenth IEEE Symposium on Mass Storage Systems and Technologies*, Maryland, April 15–18, 2002, IEEE, New York (2002), pp. 259–270.
10. L.-F. Cabrera, R. Rees, and W. Hineman, "Applying Database Technology in the ADSM Mass Storage System," *Proceedings of the 21st International Conference on Very Large Databases* (*VLDB*), September 1995, Zurich, Switzerland, Morgan Kaufmann Publishers, San Francisco (1995), pp. 597–605.
11. M. Kaczmarski, T. Jiang, and D. A. Pease, "Beyond Backup Toward Storage Management," *IBM Systems Journal* **42**, No. 2, 322–337 (2003, this issue).
12. L. L. Ashton, E. A. Baker, A. J. Bariska, E. M. Dawson, R. L. Ferziger, S. M. Kissinger, T. A. Menendez, S. Shyam, J. P. Strickland, D. K. Thompson, G. R. Wilcock, and M. W. Wood, "Two Decades of Policy-Based Storage Management for the IBM Mainframe Computer," *IBM Systems Journal* **42**, No. 2, 302–321 (2003, this issue).

13. A. Benner, *Fibre Channel: Gigabit Communications and I/O for Computer Networks*, McGraw-Hill, Inc., New York (1996).
14. M. Hartung, "IBM TotalStorage Enterprise Storage Server: A designer's view," *IBM Systems Journal* **42**, No. 2, 384–397 (2003, this issue).
15. J. Satran et al., "Internet Protocol Small Computer System Interface (iSCSI)," RFC 3385, Internet Draft, IETF, 2002, http://www.ietf.org.
16. P. Sarkar, K. Voruganti, K. Meth, O. Biran, and J. Satran, "Internet Protocol Storage Area Networks," *IBM Systems Journal* **42**, No. 2, 218–231 (2003, this issue).
17. J. S. Glider, C. F. Fuente, and W. J. Scales, "The Software Architecture of a SAN Storage Control System," *IBM Systems Journal* **42**, No. 2, 232–249 (2003, this issue).
18. J. Menon, D. A. Pease, R. Rees, L. Duyanovich, and B. Hillsberg, "IBM Storage Tank—A Heterogeneous Scalable SAN File System," *IBM Systems Journal* **42**, No. 2, 250–267 (2003, this issue).
19. "SGI CXFS: A High-Performance, Multi-OS SAN File System from SGI," White Paper, Silicon Graphics, Inc., Mountain View, CA (2002).
20. P. Horn, *Autonomic Computing: IBM's Perspective on the State of Information Technology*, IBM Corporation (October 15, 2001); available at http://www.ibm.com/autonomic.
21. R. Merritt, "IBM Stacks Hard-Disk Bricks to Build Dense Storage Cube," *EE-Times*, May 1, 2002, www.eetimes.com/at/news/OEG20020423S0091.
22. A. Azagury, V. Dreizin, M. Factor, E. Henis, D. Naor, N. Rinetzky, O. Rodeh, J. Satran, A. Tavory, and L. Yerushalmi, "Towards an Object Store," *Proceedings of the 20th Symposium Mass Storage Systems and Technologies* (April 2003).
23. J. Rothenberg, "Ensuring the Longevity of Digital Documents," *Scientific American* **272**, No. 1 (January 1995).
24. R. A. Lorie, "Long Term Preservation of Digital Information," *Proceedings of the First Joint Conference on Digital Libraries*, Roanoke, VA, June 2001, ACM, New York (2001), pp. 346–352.
25. D. A. Thompson and J. S. Best, "The Future of Magnetic Data Storage Technology," *IBM Journal of Research and Development* **44**, No. 3, 311–322 (May 2000).
26. "Move Over Silicon," *The Economist Technology Quarterly*, Dec. 14, 2002, pp. 20–21.

Accepted for publication March 9, 2003.

Robert J. T. Morris *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (rjtm@us.ibm.com).* Dr. Morris is the director of the IBM Almaden Research Center where he oversees scientists and engineers doing exploratory and applied research in hardware and software areas such as nanotechnology, materials science, storage systems, data management, Web technologies and user interfaces. He is also Vice President for personal systems and storage research, managing this worldwide research work within IBM. A computer scientist, Dr. Morris has over 20 years of experience in the IT industry. Before coming to Almaden as lab director, he was a director at the IBM Thomas J. Watson Research Center in New York, where he led teams in personal and advanced systems research. He began his employment with IBM at Almaden, working on storage and data management technologies. Previously, he was at Bell Laboratories where he began his career in computer communications technology. Dr. Morris was named chairman of the Bay Area Science Innovation Consortium (BASIC) in 2002. He has published more than 50 articles in the computer science, electrical engineering, and mathematics literature and has received 12 patents. He holds a Ph.D. degree in computer science from the University of California at Los Angeles and is a member of the IBM Academy of Technology and a Fellow of the IEEE. He was an editor of the *IEEE Transactions on Computers* from 1986–1991 and serves on various university advisory committees, including the Government University Industry Research Roundtable.

Brian J. Truskowski *IBM Systems Group, Route 100, Somers, New York 10589 (ski@us.ibm.com).* Mr. Truskowski is general manager, Storage Software, IBM Systems Group. He was named to this position in January 2003. He is responsible for developing and implementing the storage software strategy, as well as the delivery of all storage software products and services. Previously, as Chief Technology Officer of the Storage Systems Group, he was responsible for overall technical strategy, ensuring that business and product implementation plans were consistent with the overall technical direction of SSG. He and his team also monitored developments in the storage industry, studying trends and new technologies to determine how to apply them to the IBM storage business. Prior to that, Truskowski was vice president, storage subsystems development, responsible for the technical strategy, development, and delivery of the IBM worldwide storage subsystems solutions, including all subsystem hardware platforms and storage management software offerings. He was named to that position in January of 1998. Mr. Truskowski joined IBM in 1981 and held several technical and management positions in the Rochester, MN development laboratory. In 1989, following the completion of advanced degree work, he was assigned to division staff in Somers, NY, before returning to the Rochester development laboratory in late 1991. In 1994, he was named director, AS/400® systems development, and had responsibility for AS/400 software development, including OS/400®. Two years later he moved to the chairman's office as a technical assistant to Louis V. Gerstner, IBM chairman and chief executive officer. Mr. Truskowski earned a B.S. degree in electrical engineering from Marquette University, and a M.S. degree in management of technology from the Massachusetts Institute of Technology. He also has an M.B.A. degree from Winona State University (Minnesota).