

Tape management in a storage networking environment

J. Deicke
W. Mueller

Today, the paradigm shift in storage architectures from direct-attached storage to storage area network and network-attached storage, and to emerging standards such as iSCSI, has a strong impact on the requirements for storage management software. The managed storage resources could be disks (from a few disks to a large disk subsystem) or tapes (from a single tape drive to large automated tape libraries). A key challenge in this field is the management of removable media such as tape, optical, or others. Besides having to keep track of a potentially enormous number of different volumes and maintaining records of important attributes such as media owner, age, I/O errors, media-pool affiliation, and much more, the open storage networking environment raises new questions. In a scenario in which the use of SCSI devices is no longer restricted to a small number of locally attached servers, security and robustness become increasingly important issues. Access conflicts must be carefully controlled, mount operations have to run in a robust manner, and it is mandatory to comply with a security concept that protects data as well as privacy and confidentiality. This paper concerns the new requirements for tape management systems in storage networking environments. It describes the challenges and relates these requirements to the standards that exist today.

Introduction

Tape storage remains a fundamental part of today's corporate data centers. It is well established as a backup medium because of its low cost per megabyte and its inherent reliability. However, applications and operating systems that use tape must support a wide variety of tape drives and libraries. The lack of an industry standard that isolates host applications and operating systems from tape hardware means that each new technology requires adaptations to existing software. The paradigm shift in architectures from direct-attached storage (DAS) to storage area network (SAN), network-attached storage (NAS), and emerging standards such as iSCSI¹ compounds the requirements for technology changes and upgrades.

¹ iSCSI is an industry standard that allows SCSI block I/O protocols (commands, sequences, and attributes) to be sent over a network using the popular TCP/IP protocol.

Instead of storage hardware being directly connected to a computer system, storage devices and hosts are now connected to separate storage networks interconnected by routers and switches. The fact that each host may potentially access each storage device increases the complexity and affects the requirements for storage management software. With the enhanced flexibility and scalability provided by storage networks, customers now have a strong demand for

- Centralized resource management that provides secure sharing of tape resources among multiple heterogeneous applications, enhanced access control mechanisms, an enterprise-wide repository for removable media, and policy-based media management.
- Platform and operating system independence to work with existing computer systems from multiple vendors.

- Application independence to provide media management functionality for a wide variety of applications.

Currently, at least two industry standards address most of these problems and are gaining widespread acceptance:

- IEEE Standard 1244 for Media Management Systems [1].
- The Common Information Model (CIM) [2] developed by the Distributed Management Task Force (DMTF) [3] and the Storage Networking Industry Association (SNIA) [4].

This paper outlines the main requirements for tape management systems in SAN environments and discusses how these standards address customer requirements.

Managing resources in a SAN

Without storage resource management software or without the explicit use of subsystems that support virtualization of resources to overcome the need for software management of shared hardware resources, any application running on any host may access any storage hardware connected to the same SAN in an uncoordinated manner. Coordinating shared access and enforcing access-control mechanisms for tape media, tape drives, and tape libraries are among the most important problems addressed by media-management software.

Robust storage management capability is commonplace in the mainframe world, where products such as the Computer Associates CA-1** and the IBM DFSMSrmm have provided enterprise-class media-management functionality for years. Now the challenge is to bring this mainframe-class media management to the heterogeneous open-systems world.

Media and drive device sharing

Typical storage networks provide physical access control by using techniques, such as logical unit number (LUN) masking or zoning, that enable, disable, or limit host access to specific hardware resources. Additionally, connections to storage devices may be reserved exclusively for a particular host bus adapter that attaches a host computer to the storage network. Because all applications running on the same host may have shared access to the same connection, all applications may potentially share the same storage—and frequently will contend for it at the most unfavorable time in the most unfavorable way. For block-based disk storage accessed through a file system layer, this might not be a problem. Here, the file system, together with the operating system, takes care of ownership and access rights and maps all of the blocks of data representing a file to physical disk storage. But tape

storage is not usually accessed in this manner, especially not in the open-systems world. In a worst-case scenario, one application is used to sequentially write data to a tape, while another application is used to rewind the tape afterwards and, when called upon, overwrite it. Without access control mechanisms, any program is allowed to read data from a tape just because the tape is still mounted in a tape drive.

Tape library sharing

Modern tape libraries such as the IBM 3584 UltraScalable Tape Library [5] allow heterogeneous open-systems hosts to share the libraries' robotics. These libraries may be partitioned into logical libraries to facilitate sharing. Each logical library has its own distinct set of drives, cartridge storage slots, virtual robotics, and control paths.

When connected to a SAN, the library does not have to be partitioned in order to be accessible from all hosts that have a control path to it. In this scenario, storage management software must provide logical grouping and pooling of cartridges and drives, thus enabling heterogeneous applications to share common media pools, such as a common scratch pool. These shared library subsystems therefore have to support fine-grained access-control mechanisms. Additional utilization enhancements can be achieved by using management software that implements drive-usage and load-balancing strategies.

Centralized resource management

Centralized resource management is the key to handling the issues that arise from sharing tape storage in a SAN environment. An architecture that contains a generic middleware that acts as a layer between tape applications and tape hardware is one way of implementing such a management system. Such an architecture should maintain a repository of all available storage resource objects and provide the capability to control access to these objects. Additionally, it must offer its management capabilities through a flexible application programming interface. Ideally, one should be able to implement such an architecture with minimal impact on existing applications.

To reduce operator interaction, a tape-management system might also provide media life-cycle management capabilities that include

- Automatic tape-cartridge recycling (e.g., when the last volume² on a cartridge has been de-allocated, the cartridge should be returned to the scratch pool).

² In terms of the IEEE 1244 standard, a volume is mapped to a partition. A partition resides on a side, and a cartridge has one or more sides. Tape cartridges will very likely have only one side, while magneto-optical disks will probably have two sides.

- Tape-cartridge retirement based on the age of a tape and the number of times it has been mounted.
- Automated movement of cartridges, e.g., holding a tape in the automated library for four weeks after its last mount, then migrating it to some vaulting location where it rests for one year, and then moving it back to a library again to be recycled.

A policy-based approach for these automated tasks would provide a high degree of flexibility for the storage administrator.

Platform, operating system, and application independence

IT departments today usually contain a wide variety of heterogeneous systems that range from personal computers (PCs) to mainframes. Successful tape storage consolidation must address all of these systems, despite their different architectures, operating systems and, most notably, applications. A very good example for providing such a high level of connectivity may be the Internet. It is based on TCP/IP networks and uses a set of text-based protocols such as HTTP and FTP to exchange messages between clients and servers. Tape-storage middleware is expected to use a similar approach.

IEEE standard for media-management systems

In 2000, the IEEE Computer Society Storage System Standards Working Group [6] released a set of five related standards for a media-management system (**Figure 1**):

- Media Management System (MMS) Architecture [1].
- Session Security, Authentication, Initialization Protocol [7].
- Media Management Protocol (MMP) [8].
- Drive Management Protocol (DMP) [9].
- Library Management Protocol (LMP) [10].

These standards define an architecture for a software component model as well as a set of protocols that describe the interfaces between these components. The architecture allows vendors to build highly scalable, distributed systems that act as generic middleware between client application software and removable media hardware. Furthermore, it centralizes resource management services and provides media-management capabilities for heterogeneous applications that utilize hardware resources in a distributed SAN environment.

A centralized resource manager, the *media manager*, is at the core of the MMS. It acts as a central repository for the metadata that describes the storage resources and provides mechanisms to control and coordinate the allocation and use of media, libraries, and drives among multiple client applications. Instead of directly connecting

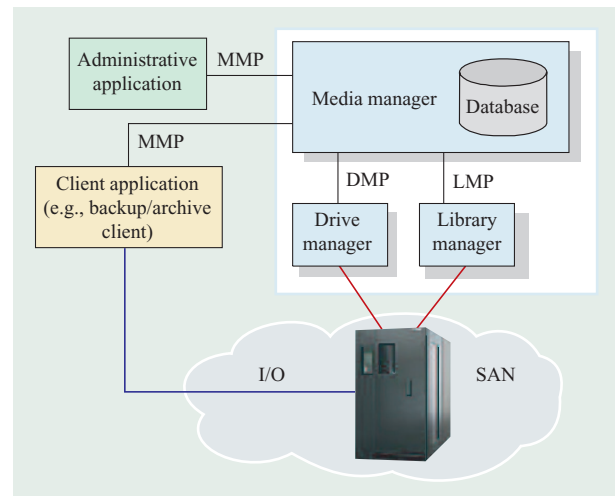


Figure 1

Architecture of an IEEE Media Management System.

tape hardware to the media manager, an IEEE-compliant MMS uses library managers to provide an abstract tape library interface and drive managers to provide an abstract interface to tape-drive hardware. These abstract interfaces provide a highly flexible and extensible way of supporting any kind of tape hardware. They are designed to abstract as many details of the hardware as practicable without compromising efficient resource management.

Library managers contain detailed information about the libraries to which they are connected. They depict the contents and capabilities of the libraries to the media manager and control the library on its behalf. On account of this additional resource virtualization, a wide range of tape library hardware can be supported. With specialized library managers, even manually operated off-line storage locations can be integrated.

Drive managers provide similar functions for the tape drives to which they are connected. They manage the configuration of drives and handle drive control tasks on behalf of the media manager. Additionally, they provide an access handle to the drive hardware. Client applications use this handle to read from or write to a cartridge in the corresponding drive. In a UNIX** system environment, such an access handle usually corresponds to a device filename.

All software components of an MMS communicate by using application-layer protocols (MMP, LMP, and DMP) over TCP/IP-based connections. They exchange UTF-8 (Unicode³) encoded text messages in a command-response style similar to well-known Internet protocols such as

³ Unicode is a character set that encompasses all of the world's living languages and is the basis of most modern software internationalization.

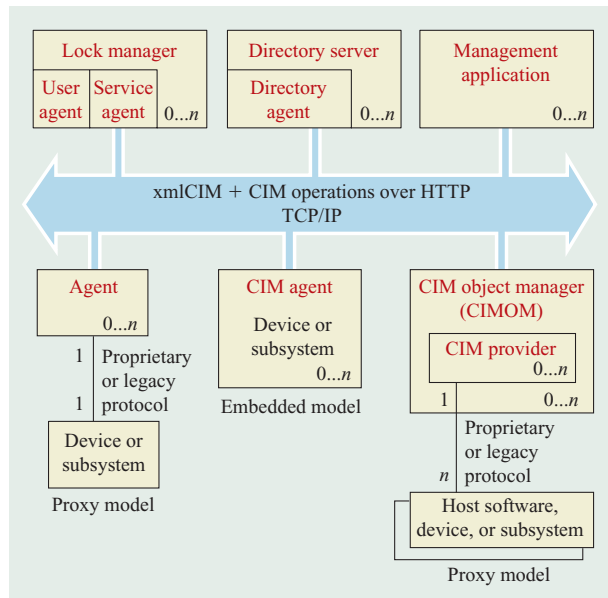


Figure 2

Storage management initiative specification (SMIS).

HTTP, FTP, or NNTP. Commands manipulate objects defined in the data model of the standard. This object-oriented model defines classes for all physical and logical components of an MMS.

Additional standards that complete a storage management system are currently being designed by the IEEE Computer Society Storage System Standards Working Group [6]. They will include a protocol that will enable the interchange of information among autonomous media managers to provide both load-balancing and fail-over capabilities as well as a data-mover architecture that will enable data transfer of two endpoints in a distributed storage system.

Common information model

The Common Information Model (CIM) [2] developed by the Distributed Management Task Force [3] as part of its Web-Based Enterprise Management (WBEM) [11] initiative offers a framework for managing system elements across distributed heterogeneous systems with a hierarchical object-oriented model. It provides a consistent, extensible mapping of elements found in network and system management. CIM inherits some of the concepts from the Simple Network Management Protocol (SNMP) but, unlike SNMP management information bases (MIBs), which often hold proprietary information, CIM provides a rich set of schemas to be used as standards for describing all managed elements in a computer system network. It can thereby be used to model

physical elements such as computers, tape hardware, or even users, as well as logical objects such as applications, processes, and files.

CIM schemas include classes that describe managed objects, class properties that define common characteristics and features of particular classes, class and property qualifiers that provide additional information, and methods that can be invoked on managed objects. Properties describe data, and methods describe behavior. Classes are related by inheritance, whereby child classes inherit both data and methods from parent classes, and by object associations that describe dependencies and component and other connection relationships. Inheritance can be used to extend this model.

The other WBEM standards include an XML-encoding scheme for the UML-based⁴ CIM class descriptions and a mapping of CIM operations onto HTTP that allows implementations of CIM to operate in an open, standardized manner. SNIA, which complements and contributes to the work of the DMTF, is providing a concrete implementation of the Common Information Model, the CIM Object Manager (CIMOM). Management applications can access managed objects through such a CIM object manager. They communicate with the object manager by exchanging request-response messages containing XML-encoded CIM operations over an HTTP TCP/IP connection.

Recently, SNIA has promoted the storage management initiative specification (SMIS), a reference model for a SAN management environment (Figure 2). SMIS [12] uses the WBEM standards and introduces new technology for security, locking, and discovery for SAN management. It contains a *lock manager* that coordinates resources such as tape libraries among multiple noncooperating clients. SMIS also provides *directory agents* for locating services in a management environment.

The SNIA Storage Media Library (SML) Working Group modeled the logical and physical aspects of tape libraries and library devices in CIM and, in 2002, began modeling media-management operations in coordination with the IEEE Storage System Standards Working Group.

With the help of a consistent CIM-based infrastructure, detecting, controlling, and managing tape hardware in a storage networking environment will become much easier. Software will be able to access all storage devices in a uniform way, while the extensibility of the class framework provides enough opportunities for modeling diversity.

Summary

Whereas the Common Information Model offers a standard way of managing all elements in a computer

⁴ Extensible Markup Language (XML) is a flexible, text-based language designed for electronic publishing, and Unified Modeling Language (UML) is a standard notation for modeling object-oriented systems.

system network, the IEEE standard concentrates on the elements needed to manage removable media and the corresponding storage devices. Major vendors have recently committed themselves to the CIM/SMIS standard and have announced CIM-based storage resource management tools and interoperable CIM-enabled storage components. Because media management plays such a significant role in storage resource management, we expect that within the next year the objects specified by the data model of the IEEE standard and their associations will be found in the CIM data model as well, providing a consistent view of all storage resources.

Because of its much broader scope, CIM does not specify any media-management-related logic as defined in the IEEE standard. With the open and flexible design of the IEEE Standard 1244, it is possible that different media and library managers may use the CIM bus to manage storage resources and provide management services for client applications.

**Trademark or registered trademark of Computer Associates International, Inc. or The Open Group.

References

1. *IEEE Standard 1244.1-2000*, "IEEE Standard for Media Management System (MMS) Architecture," © 2000, IEEE Computer Society.
2. Common Information Model (CIM); see <http://www.snia.org/cim/>.
3. Distributed Management Task Force, Inc. (DMTF); see <http://www.dmtf.org/>.
4. Storage Networking Industry Association (SNIA); see <http://www.snia.org/>.
5. IBM Tape and Optical Storage; see <http://www.ibm.com/servers/storage/tape/>.
6. IEEE Computer Society Storage System Standards; see <http://www.ssswg.org/>.
7. *IEEE P1244.2/D041900 Standard Draft*, "IEEE Standard for Media Management System (MMS) Session Security, Authentication, Initialization Protocol (SSAIP)," © 2000, IEEE Computer Society.
8. *IEEE Standard 1244.3-2000*, "IEEE Standard for Media Management System (MMS) Media Management Protocol (MMP)," © 2000, IEEE Computer Society.
9. *IEEE Standard 1244.4-2000*, "IEEE Standard for Media Management System (MMS) Drive Management Protocol (DMP)," © 2000, IEEE Computer Society.
10. *IEEE Standard 1244.5-2000*, "IEEE Standard for Media Management System (MMS) Library Management Protocol (LMP)," © 2000, IEEE Computer Society.
11. Web-Based Enterprise Management (WBEM) Initiative Common Information Model; see <http://www.dmtf.org/standards/standardwbem.php/>.
12. SMIS specification; see <http://www.snia.org/smi/>.

Received July 23, 2002; accepted for publication January 7, 2003

Juergen Deicke *IBM Speichersysteme GmbH, Hechtsheimer Strasse 2, Gebaeude 82, Abteilung 4686, 55131 Mainz, Germany (deicke@de.ibm.com).* Dr. Deicke received the Dipl.-Ing. and Ph.D. degrees in electrical engineering from Darmstadt University of Technology, Darmstadt, Germany. He joined IBM in May 2000, working both as a software architect and a development manager in the area of storage management software. Since April 2002, he has headed the Software Development Open Systems Department in Mainz, Germany, where the focus is on software development projects supporting open standards such as CIM/WBEM. Dr. Deicke chairs the Interoperability Committee of the Storage Networking Industry Association Europe (SNIA Europe) and lectures at the Bingen University of Applied Science in Bingen, Germany. Before joining IBM, he worked in the area of multimedia communications, especially MPEG-4. Dr. Deicke is a member of the IEEE.

Wolfgang Mueller *IBM Speichersysteme GmbH, Hechtsheimer Strasse 2, Gebaeude 82, Abteilung 4686, 55131 Mainz, Germany (wolfmuel@de.ibm.com).* Mr. Mueller received his Dipl.-Inform.(FH) degree in computer science from the University of Applied Sciences, Darmstadt, Germany. He joined IBM in March 2001 and is currently working as a software architect in the Software Development Open Systems Department in Mainz, Germany, where the focus is on software development projects supporting open standards such as CIM/WBEM. Mr. Mueller is a member of the IEEE Computer Society.