Media Management for Audiovisual Digital Archiving

By A. D'Alessio, A. Bertini, F. Ciferri, G. Ferrari, and M. Strambini


Angelo D'Alessio


Adriano Bertini


Filippo Ciferri


Giulia Ferrari

This paper describes an integrated solution for the management and migration of heterogeneous data, hierarchical storage management (HSM), an innovative software/hardware architecture developed specifically for the digital archiving of audiovisual mass content. i-DHSM (Dynamic Hierarchical Storage Management) is a highly scalable and configurable solution for a wide range of applications. It provides full performances on both symmetric multiprocessor (SMP) and cluster systems.

Easy manageability of audiovisual files and moving clips over high-speed connections to high-performance devices is what today's broadcasters, dot-com companies, and telecommunication operators request most when they plan a digital archive. Audio/video clips for large digital storage file require specific management of ingestion and migration procedures when moving files from server to backup devices (automated tape libraries, etc.) and vice versa, no matter what quality or application is used (Media Asset Management and Archive Management, for example).

HSM Systems

Any hierarchical storage manager (HSM) system, engineered to provide integrated classification of audiovisual clips and automated migration of files from server to backup devices and vice versa, has limited space on a local hard disk for content storage, compared to the space available on dedicated devices for backup operations. For this reason, the HSM software application automatically moves less used files to tapes and keeps the most requested clips locally, thus saving space and time.

Because HSM applications are usually very input/output (I/O) demanding, the server has a powerful bus onboard, connected to a local RAID array and able to respond to the requested throughput. Thus, typically, a symmetric multiprocessing (SMP) machine will be used as the server, while a storage area network (SAN) solution is responsible for the fast connection required for the I/O operations. Using this configuration as a starting point, new solutions will be investigated, regarding both the software and the architecture of HSM systems.

In this paper, the term "DFS Manager" (Data File System Manager) will refer to a particular device (or logical block) dedicated to the data ingestion/retrieving inside an HSM system. The term "DFS Storage" (Data File System Storage) will refer to the device (or logical block) that stores the data. A simple description of a traditional HSM system is given.

Cluster Approach

In the last few years many efforts were made to move from the expensive SMP system to a more flexible and economical approach. Clusters seem to be the right choice; with some limitations, they satisfy all the requirements that large SMP systems resolve today.

Typically, in a last-generation cluster architecture, some machines are used as nodes and one or more as front-end units. Nodes and front-end units exchange data on a private network, which is usually a high-speed connection, while the front-end-only is committed to exchanging data with the external world.

Presented at the 35th SMPTE Advanced Motion Imaging Conference (paper no. 35-8), in Washington, DC, February 8-10, 2001. A. D'Alessio, A. Bertini, F. Ciferri, G. Ferrari, and M. Strambini are with SHS Multimedia S.p.A., Italy.

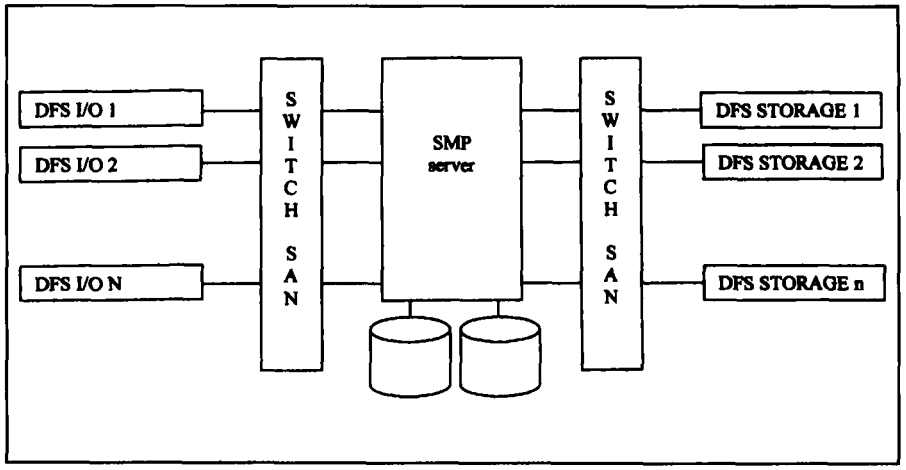

Figure 1. Example of traditional HSM system.

Generally, each node, realized with a personal computer (PC), has a local storage memory that can be shared among all the connected nodes. A cluster might also be composed of heterogeneous nodes, for example, by dedicating N nodes to intensive operations and M nodes to simple processing. In the HSM application this will be recommended whenever there are N data flows that require non-specific treatment prior to migration and M data flows (for example, pictures from satellite), which require intensive CPU processing prior to migration.

The intrinsic fail-safe capacity of the cluster architecture must be emphasized. Every time a node fails, its function can be directed to another working node inside the system. Furthermore it is possible to construct a system that contains more nodes than needed for normal operation. In this way, we obtain redundancy that grows linearly with the number of redundant nodes added into the system. In addition, it would be simple to add more power to the system without interrupting the system's work, since the cluster that will use the added nodes can be reconfigured "on the fly."

The advantages of the cluster over the SMP system can be summarized in:

- Fail-safe architecture
- Progressive redundancy
- Simple expandability and scalability
- Investment savings
- Dynamic reconfigurability
- Accomplishment of heterogeneous tasks

At this point it is easy to transform the architecture by applying the rules of a cluster system (Fig. 1). The diagram in Fig. 2 is an illustration of the result. This simplified diagram shows that every node is attached to a local storage unit that may be used as a buffer for I/O operations. In this way redundant nodes are guaranteed over the disk, because they are a part of the cluster node's management.

During the normal workflow, a node can be assigned to a well determined couple comprised of a DFS I/O module and a DFS storage module, freely selected from that connected to the SAN switch, every time it is needed. Data coming from/to the DFS I/O are processed by the assigned node, then passed/retrieved to the DFS storage or saved, in the meantime, onto the local node's disks. Thus these disks could substitute the RAID found in the SMP approach (Fig. 1).

The limitations of this process reside in the I/O capacity of the single node; as indicated, PCs are generally used as nodes. In the typical use of a cluster (i.e., computational intensive) the single CPU power of a PC is comparable to that of the SMP machine, conversely, the bus of a PC machine is absolutely not comparable with the SMP. For the PC, a maximum throughput of 150 MBytes/sec was considered a good result for the typical bus. Keeping this parameter in mind, we must remember that a node, with intensive I/O use, must support a connection to the cluster (private network), as well as a connection to the disk and the Fibre Channel that connect it to the SAN switches.

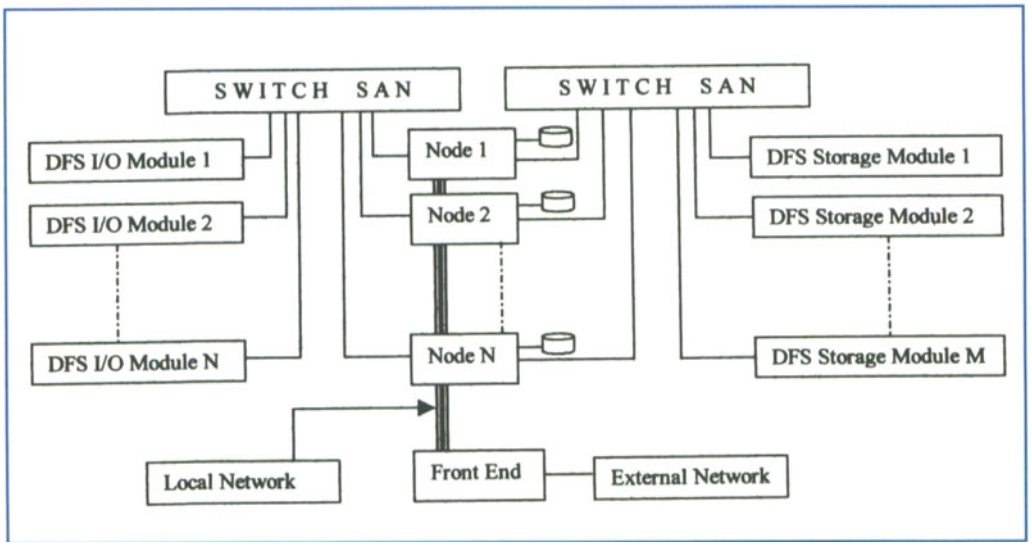

Figure 2. HSM system based on cluster architecture.

i-DHSM

For satisfying both new architectural schemes and more complex user requests, we attempt to describe a software solution that rises from traditional HSM systems to the latest cluster configurations. The "i-DHSM" (i-Dynamic Hierarchical Storage Manager) is a software toolset fully implemented and engineered for integrated media asset management.

The basic idea behind the product was to create a software that could be scalable on different operating systems, and one way to achieve this was to build a simple, small kernel with all the basic functions implemented inside it. The kernel uses a custom protocol to exchange outside data, thus implementing all the high-level functions of different modules as a way to comunicate with the kernel and any other module that may became a part of the application.

Another advantage of this approach is its capacity to analyze or change specific data that normally resides in memory, but that, in this case, also becomes available to every external module not included in the main application. This feature simplifies, for example, the construction of external monitoring programs, as it becomes very simple to access the application data over a distributed network.

In both approaches, SMP and cluster, it is important to emphasize the requirements of high parallelism, which is how the kernel accommodates synchronizing the access to internal/external data. Every module attached to the kernel has the capacity to be run concurrently with the other module present in the application. In this way it is possible to exchange the modules within different hardware architectures. It is also possible to write software drivers dedicated to interfacing specific devices with the system. This ability is fundamental due to the number of different devices typically used in HSM systems.

In addition, the scalability of the achitecture allows easy integration of particular modules for the processing of data into the system; for example, a specific application tool has been developed for fast access to a portion of a clip requested by the user. In the case of content stored in MPEG-2 format, for instance, compressed data are processed and the selected sequence is extracted preserving the consistency and accuracy of the data in a transparent way for the user.

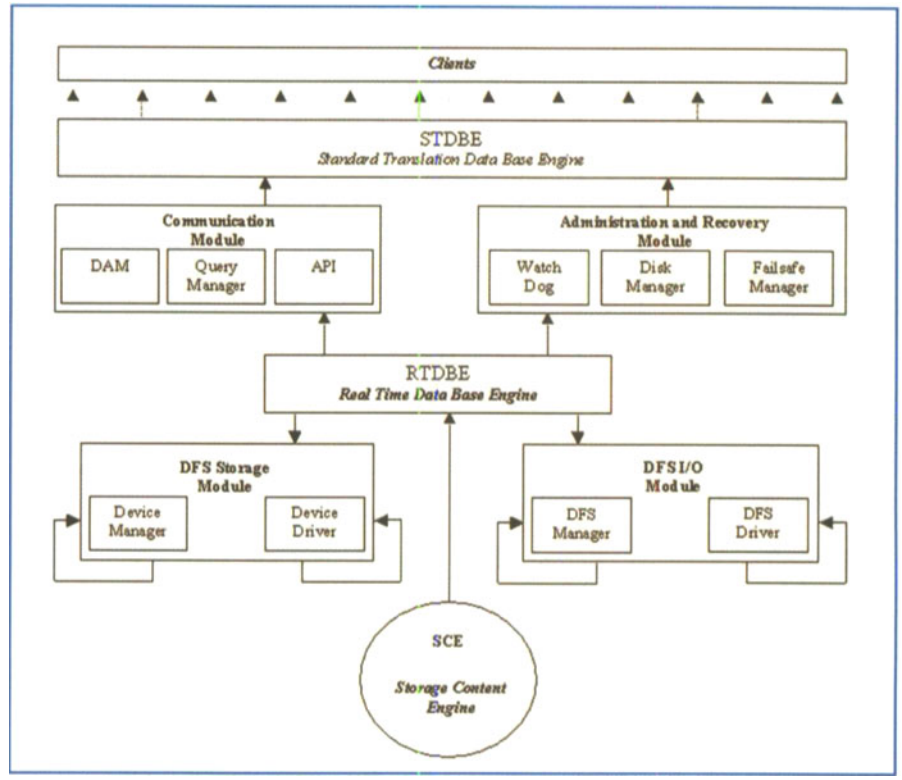

Figure 3. i-DHSM block diagram.

It is worthwhile to note a particular module that manages all the data flow of the application and translates it into database format. The Real Time Data Base Engine (RTDBE) is designed to be linked simultaneously to all the modules comprising the application, thus maintaining the available data even if a fault occurs. This is achieved by maintaining access to the data with a very low delay, using direct access to the device that contains it (typically a system disk) and an optimized structure based on dynamic page reallocation algorithm (a page contains a specific piece of data group).

A simple feature that was added to the RTDBE module is the capacity to import/export data from/to the application and external database format (Standard Translation Data Base Engine [STDBE]). Using this module, it becomes simple to interface the entire application with the existing user data environment.

Figure 3 is a block diagram of the main i-DHSM module's structure and linking, with some modules grouped by their functions. In the figure, the "Storage Content Engine" is the kernel module, and it monitors the i-DHSM flow by controlling the following modules:

- Device Manager. This module interfaces all devices connected to an i-DHSM system and controls the drivers developed according to specifications allocated by each device.
- DFS Manager and relevant drivers for ingestion/retrieval devices. Manages system data during I/O procedures, such as file consistency and hierarchical structure. If requested, data can be accessed and changed according to specific algorithms.
- Distributed Access Module

(DAM) used for data sharing over distributed geographical areas by connecting two separate i-DHSM systems, each having a specific HW configuration and installed in different sites.

- Query Manager module for the management of incoming queries.
- Application Programming Interface (API). Users can access and change any system event occurring during daily operational flow.
- WatchDog Module monitors the status of each process and, in case of fatal errors due to hardware failure, adjusts/changes settings, in order to avoid system shutdown.
- Disk Manager, controls the management of data files on local storage disks and preserves the system from any overloading by analyzing runtime statistical data, such as data access, profile of users, latency, etc., previously sampled.
- FailSafe Manager monitors the status of all elements and devices; in case of failure, this module re-allocates resources over the whole system, thus preventing overloading.

One example of real i-DHSM installation is the "TecaFast" provided for RAI (Radio Televisione Italiana) installed in Rome (Italy). The goal of this project is to build a high-quality audio/video digital archive of all the material that RAI produced in nearly 45 years (~400,000 hr) plus the actual trasmissions. All the multimedia materials, MPEG-2 4:2:2 files digitized at 10 Mbits/sec, must be accessed from different locations connected by a Fibre Channel network. Every location has access to the data using some encoding/decoding machine such as DFS I/O devices and PC workstations for database information browsing. The main system is composed by two SMP servers each with 8 CPU and 2 GBytes of memory connected to a 2.5 TB RAID disk.

The DFS Storage is an automated library containing 6000 cartridges and 10 tape drives. Currently, there are 40 client workstations, distributed in the main RAI offices within the city, that can access the data at the same time. In the near future it is expected that there will be 100 distributed over the entire country with distant locations accessible by satellite link or small regional installations.

Conclusion

The i-DHSM's modules are based on today's HSM user requirements, but it is anticipated that different customer needs may require adding more modules to the main application. The i-DHSM API will be a good choice for accomplishing this because of its ability to introduce new functions at different levels, starting from the high-level interface down to the kernel. The system will become more complex but will maintain good flexibility, and for this reason, could be one possible solution to the HSM system approach.

THE AUTHORS

Angelo D'Alessio is a Board Member at SHS Multimedia, a technology and systems provider of broadcasting and new media applications.

Before joining SHS, he worked in many aspects of television engineering and on a range of projects including those related to broadcast color systems. He was a project leader in HDTV systems and technical director of some key high-definition electronic movies, digital television systems, and 3-D systems for the medical field. In recent years he has worked in the areas of special integration and convergence of information technology, telecommunication technology, and broadcasting.

D'Alessio, a SMPTE Fellow, has served as an International Governor, and is currently Director of International Sections.

Giulia Ferrari graduated in electronic engineering from Politecnico of Milano in 1993. Since 1995 she has been a software engineer at SHS Multimedia. She has been involved in interactive CD-ROM applications and in MPEG projects, focusing on system layer aspects and developed the software of transport stream multiplexing for the first Italian NVOD project and libraries for the transcoding of MPEG streams.

Ferrarri has also been responsible for SHS participation in ESPRIT projects.

Following graduation, **Adriano Bertini** focused his attention on the logic analyzer environment. Early in his career, he worked mostly in the field of realtime graphics on dedicated machines, developing architectural reconstruction applications. From 1997 to 1999, he co-developed a magnetic-field compensation algorithm for realtime data sampling and realtime motion capture applications.

Bertini has worked at SHS Multimedia since 1999, designing and developing the i-DHSM project, with special attention to SMP and cluster implementation.

Filippo Ciferri earned a degree in physics in 1993, with a thesis regarding optimization algorithms on parallel machines. From 1993 to 1997, he worked mostly in the field of realtime graphics on dedicated machines developing architectural reconstruction applications. With his colleague, Adriano Bertini, he co-developed a magnetic-field compensation algorithm for realtime data sampling and realtime motion capture applications.

The design and development of the i-DHSM is a project Ciferri has been working on at SHS Multimedia since 1999. The focus of his attention is SMP and cluster implementation.