IBM Data Facility Storage Management System

Summary
The IBM DFSMS automates the management of data and storage for IBM and plug-compatible mainframes. Other software products provide some of the same functions as DFSMS, but none provide as complete a set of capabilities:

Note
In a Statement of Direction IBM has indicated that z/OS v.1.6 will be the last release in which the Indexed Sequential Access Method (ISAM) will be available in DFSMS. IBM has provided the ISAM interface that allows users to run an ISAM program against a Virtual Storage Access Method (VSAM) Key-sequenced data set (KSDS).

Table of Contents
- Overview
- Analysis
- Pricing
- Competitors
- Strengths
- Limitations
- Insight
Overview

DFSMS is a suite of automated tools for centrally managing data in a mainframe environment that runs z/OS, z/OS.e or S/390. DFSMS implements the specific policies of a data center to manage data placement and retrieval throughout the storage hierarchy of disks, tapes (including those off-site) and optical devices. Each installation must define performance, availability and migration objectives for its different classes of data. The storage administrator also needs to define the physical attributes of the site’s storage configuration and policies for determining where data is initially placed. Once these activities have been completed, DFSMS will automatically manage the data from its creation to the time it can be removed from the system and all archives.

DFSMS consists of five modules. Only one module, the Data Facility Product (DFSMSdfp), is included with the operating system license. The other four modules are distributed with the operating system but require add-on licenses that also incur additional charges. The other modules in the suite are Removable Media Manager (DFSMSrmm), Data Set Services (DFSMSdss), Hierarchical Storage Manager (DFSMShsm) and, the newest module, Transactional VSAM Services (DFSMStvs). DFSMSrmm, DFSMSdss and DFSMSStvs can be purchased individually, but DFSMSHsm requires DFSMSdss, and the two come bundled together.

DFSMSdfp is the heart of the storage management system. It keeps track of all data and programs within the z/OS environment and provides access methods to read and write data. An access method defines the organization of the data in a data set and the technique by which the data is stored and retrieved. Each access method has its own data structure to organize the data, macro instructions to process its data sets and utility programs to manipulate its data sets. The two most common access methods are the VSAM and the Object Access Method (OAM). OAM processes named byte streams that have no record boundaries or internal organization that the system maintains. They are processed in their entirety. VSAM data sets have an internal organization, records. VSAM arranges records by an index key, relative byte address or relative record number.

DFSMSrmm is the tape management system that is used to track the location of all tapes, including those at remote sites or in transit.

DFSMSdss is the component of DFSMS that is used for high-performance data movement. DFSMSdss provides backup and recovery functions at the data set/object or volume levels (for example, at the logical or physical level). It also includes a Stand-alone Restore Program for disaster recovery that allows system packs to be restored without requiring an operating system environment. The SnapShot function produces copies of data sets or volumes without actual data movement. SnapShot can be called from DFSMSdss, DFSMSHsm or DB2. Concurrent Copy allows data to be copied or dumped while other applications
continue to update the data. Both Information Management System/Enterprise Systems Architecture (IMS/ESA) and DB2 can use Concurrent Copy for their backups.

DFSMShsm provides two main functions: availability management and space management. The availability management capabilities ensure that a recent copy of disk data exists. Data sets or objects can be backed up (a logical copy is made) automatically or by command. Entire disk volumes can also be dumped to tape. DFSMShsm has a disaster-recovery feature called Aggregate Backup and Recovery Support (ABARS) which can consistently capture all data necessary to rebuild an application.

DFSMShsm’s space management functions are used to manage disk space by freeing unused space in data sets, compressing inactive data sets or moving data sets off fast-access storage devices, thus creating free space for new allocations. These actions can be performed on a schedule or to meet emergency space requirements.

DFSMStvs enables batch jobs and Customer Information Control System (CICS) online transactions to concurrently update shared VSAM data sets in an IBM Parallel Sysplex. Multiple batch jobs and online transactions can be run against the same VSAM data sets. DFSMStvs ensures data integrity for the concurrent batch updates, while CICS ensures it for the online updates. Using DFSMStvs it is no longer necessary to quiesce online CICS processing to perform batch updates. DFSMStvs provides transactional recovery directly within VSAM. It gives CICS/VSAM database-like recovery properties similar to those that have been available for DB2 and IMS databases. DFSMStvs is an extension to the functions provided by VSAM record-level sharing (RLS). It builds on the locking, data caching and buffer coherency functions provided by VSAM RLS, which is a prerequisite. It adds logging, two-phase commit and backout protocols to provide full transactional capability and sharing for non-CICS applications.

The data that DFSMS needs to perform its tasks are recorded in three structures called “classes,” and one structure called a “group.” (Object data is handled using an object storage group structure and an object class structure.) The data class structure holds information about the type of data for which space must be allocated. Each class or type of data that is defined in the data class structure contains the following types of information:

- Record size
- Record organization
- The operations that can be performed on this data, for example, compaction
- The amount of unused space that must be allocated
- Who can access the data.
- How long the data must be retained.
- Will the data be simultaneously accessed by multiple systems in a Parallel Sysplex (RLS)?

The storage class structure is used to store information about performance goals and device-availability requirements. Each storage class entry contains information about performance objectives, such as the amount of time it takes to first access the data, how quickly data must be returned following a data request and the sustained transfer rate that must be achieved. If the data set being defined provides record-level sharing (RLS), its performance requirements are also specified in this structure by defining cache characteristics for the Coupling Facility (CF).

Each entry in the management class structure contains the following type of information:
The method used to determine the data’s expiration date
The criteria for migrating data to slower, less expensive storage
Backup information, such as whether backup occurs automatically or requires operator intervention, how frequently it occurs (if automated), retention period for backup data, number of versions of the backed up data that should be retained

Each record in storage group structure is used to specify the physical volumes that should be processed by the DFSMShsm and the system on which processing should occur.

The storage administrator writes automatic class selection (ACS) routines for each of the three types of classes and another routine to assign storage groups. When a new data set is allocated, SMS first runs the ACS data class routine to determine whether the new data set should be placed in any of the established data set classes. The storage class ACS routine is then run, and if a storage class is assigned, then the data set is placed under SMS control. If the data set is SMS-managed, then the management class routine is run to assign a management class. Finally, the storage group routine is run to find possible storage groups for the data set. The data that is needed to determine the specific class or group in which a data set should be placed is based on information from Job Control Language (JCL) or other allocation parameters. The ACS routines, combined with the class structure definitions and the storage group definitions, define a site’s SMS configuration.

Other members of the DFSMS family include Data Facility Sort (DFSORT), for fast sorting and DFSMS Optimizer to analyze performance and make recommendations for changes. These are separately licensed and distributed programs that are not included with the operating system media.

Analysis
System Managed Storage, the concept behind DFSMS, grew out of customer interactions at IBM user group conferences in the early 1980s. The first version of DFSMS made its appearance in 1988. Since that time, the need for system-managed storage has only increased. There has been an exponential growth in electronically stored information, made possible and affordable by the dramatic drop in the cost of storage hardware. Data growth itself simply requires more administration. In addition as the amount of data grows, storage administrators require increasingly greater expertise to simultaneously manage performance, availability, and archival and data retention requirements. The move to system-managed storage represents an attempt to reduce the complexity and cost of storage management through automation.

To simplify the ability of the user to manage data as well as the administrative procedures for defining an installation’s data management policies, IBM provides the Interactive Storage Management Facility (ISMF). ISMF allows users to perform functions against data sets, and disk, tape, and optical volumes. Users can also obtain information about the current SMS class definitions. For the storage administrator, ISMF provides dialogs for accomplishing many storage management tasks including defining and changing system-managed storage (SMS) class and group definitions. ISMF can also be used to display storage usage information. ISMF can be run in batch mode using the NaviQuest Tool. Both ISMF and NaviQuest are included with DFSMSdfp.

To improve application availability, DFSMS provides emergency storage capacity through extended storage groups and overflow storage groups to ensure that lack of storage space does not result in premature termination of the application. To improve performance, DFSMS provides:

- The ability to spread data sets across different physical control units (PCU) to reduce the impact of a PCU failure.
IBM Data Facility Storage Management System

• Providing a single, common queue for DFSMS requests in a Parallel Sysplex that allows any host to process requests.

IBM plans to introduce new capabilities for DFSMS in z/OS v.1.5, scheduled for release in the first quarter of 2004, that reduce the occurrence of system outages, enhance disaster recovery, improve storage administrator productivity, and enhance data security. Also planned for z/OS v.1.5 are improvements to application backup by enabling DFSMSHsm to use volume-level fast replication that exploits the features of the IBM Enterprise Storage Server (ESS) and RAMAC Virtual Array (RVA). DFSMSHsm fast replication will offer a fast, easy-to-use backup and recovery solution specifically designed to work with DB2 Universal Database (UDB) v.8.

Pricing

The cost of DFSMSdfp is included in the license fee for OS/390, z/OS and z/OS.e. The other modules are priced individually and are subject to the Variable Workload License Charge (VWLC) rates.

GSA Pricing

No.

Competitors

There are a few software products on the market that provide some of the same functions as the DFSMS family, but none provide as complete a set of capabilities:

• Innovation Data Processing with Fast Dump Restore (FDR) family of products
• Computer Associates with CA-Disk and CA-ASM2.

As data centers become more heterogeneous, the need to centrally manage and share data that is created on or resides on the different system environments becomes increasingly critical. Shared storage allows:

• Applications to be more easily moved between servers for better resource utilization and
• Eliminates the overhead associated with moving the data between platforms that use different data formats.

Storage subsystems from competitors such as EMC Corp. with their Symmetrix line of Enterprise Storage Servers, HDS with their Lightning 9000 Storage Server and IBM itself, with the Enterprise Storage Servers offer advanced functions like copy services that extend the capabilities of DFSMS. IBM also supports its own Enterprise Storage Servers on mainframe servers by directly using ESS facilities for DFSMS functions.

Strengths

Efficient Use of Storage

Efficient space management reduces wasted or unused space, thus reducing operating costs by postponing new purchases. DFSMS automatically deletes obsolete data, removes wasted space and migrates infrequently used data sets to lower-cost storage. DFSMS can enhance application performance through data placement.

Consistent Storage Policies
Centralized data storage management allows companies to set consistent policies for data backup and retention. Data recovery requirements can vary by the type of data. DFSMS provides options ranging from online duplexing of data and continuous backup to more conventional strategies.

**Limitations**

*Complexity*

DFSMS allows the user tremendous flexibility in placing files, migrating files and designing effective backup and restore policies. This tremendous flexibility makes it necessary to specify many attributes to efficiently configure the storage environment.

*Processing Overhead*

Using the full scope of features that the DFSMS suite of software provides can incur considerable processor and storage overhead.

**Insight**

There are also software products that can be substituted for comparable DFSMS components that consume fewer processor resources. Innovation Data Processing’s FDR family of products frequently outperforms the corresponding DFSMS component in backup, restore, recovery and migration functions, using fewer resources than DFSMS. The drawback in using these products is that they have to be interfaced with DFSMS and are subject to revision whenever IBM introduces changes. Using only IBM components reduces the chance of incompatibilities occurring, since the IBM components have gone through extensive integration testing.

High-end storage servers provide similar capabilities to the combination of IBM directly attached storage devices and DFSMS albeit with somewhat less flexibility. These systems offer large storage capacity in a small footprint and centralized storage accessible by mainframes, and Unix, Linux and Windows servers using high-performance interconnects native to the respective environment with no additional processor overhead on the host server system.

The choice of approach to centralized mainframe storage management for operational sites will be dependent on the size of the installation and the requirements for data and storage management. For new installations, users will obviously want to explore the cost, performance, capacity and complexity trade-offs between using IBM Storage with DFSMS and the storage server approach.