IBM® DB2® Universal Database

Hierarchical Storage: Using DB2 Universal Database and Tivoli® Space Manager

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 35.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2000. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

DB2 Universal Database Support of Tivoli® Space Manager

Executive Summary

This paper discusses how Hierarchical Storage Management (HSM) and DB2 databases can interoperate to automatically and transparently migrate rarely accessed data to Tivoli Storage Manager (TSM) storage. Using the cheaper alternative storage that is available through HSM, optical storage can retrieve small files rapidly, while tape storage is better suited for retrieval of larger files.

The results of a trial using DB2 Universal Database (DB2 UDB) and Tivoli Storage Manager in a large partitioned database environment where data is rarely accessed show:

- It is possible to configure TSM with Tivoli Space Manager and DB2 UDB so that they work together.
- Once the data is migrated, DB2 UDB is able to access only the required data from the alternative storage under TSM with Tivoli Space Manager.
- The performance of the alternative storage devices is slower than disk (DASD), but is acceptable given the characteristics of those devices.

If you are looking for a storage solution that assists you in managing increasing volumes of data, this paper will provide you with a dynamic, viable, and easy-to-use solution that uses DB2 UDB and Tivoli Storage Manager. If you are looking to keep the cost of DASD in your environment down while still maintaining the ability to access increasing volumes of historical business data, then an effective choice is to use DB2 UDB and Tivoli Storage Manager.

Introduction

Recent years have seen a tremendous growth in data volumes. This growth has been fueled by distributed applications and data-intensive technologies, such as business intelligence applications and the Internet. Historical data may provide excellent insight into business trends and directions. The ever-increasing amounts of data are stored in databases which need to be accessible all the time and available to business users on demand.

Several goals will be of interest to you in this environment:

- Minimize the hardware costs (since DASD is expensive relative to other forms of storage)
- Maximize access to the data
- Implement a solution that has the least effect on overall system performance
- Minimize the administration costs

Managing the location of historical data is important since intervention is required when moving data from an expensive high performance disk storage location to a less expensive alternative storage location. A system is available that can manage the location of the data for you based on guidelines that you establish, making it possible to achieve the goal of minimizing administration costs.

This white paper presents how the IBM product Tivoli Storage Manager (TSM), and in particular its optional component, Tivoli Space Manager, can be configured

© Copyright IBM Corp. 2000

to work together with IBM’s powerful DB2 Universal Database (DB2 UDB). The ability of these products to work together allows alternative storage options and enables IBM customers to handle the demands of growing databases and a growing number of database applications. These products also allow access to only the required data from that which has been stored on the less expensive alternative storage devices used by Tivoli Space Manager. This white paper documents how DB2 UDB with Tivoli Space Manager can take advantage of nontraditional database storage, such as tape and optical drives, to store infrequently used or historical data yet keep the data transparently available to users.

The testing documented in this paper was conducted by a team of researchers and specialists from IBM’s Toronto Laboratory, from Tivoli Storage Manager (TSM), and from IBM’s RS/6000 Teraplex Integration Center, a unique facility which studies very large database applications.

This paper will be useful if you are interested in alternative storage solutions for your DB2 databases. The first section of this paper introduces the products and testing facilities. Following this, the interoperability of the products is described, the test scenario and results are presented, and finally, valuable tips and techniques that will help you take advantage of the storage options available from the interaction of these products is discussed. Here is a list of the sections in which you may be interested:

- “The Products and Environment”
- “How DB2 Universal Database and Tivoli Space Manager Work Together” on page 7
- “Test Scenario and Results” on page 17
- “Tips and Techniques” on page 24
- “Summary and Conclusion” on page 28

The Products and Environment

This section introduces a Teraplex environment, DB2 Universal Database (DB2 UDB), and Tivoli Storage Manager (TSM) with Tivoli Space Manager. The following list of products allows you to go directly to the section you are interested in:

- “Teraplex Integration Centers Overview”
- “DB2 Universal Database Overview” on page 3
- “Tivoli Storage Manager with Tivoli Space Manager Overview” on page 5

If you are already familiar with these topics, then you should proceed to the next major section of this paper, “How DB2 Universal Database and Tivoli Space Manager Work Together” on page 7.

Teraplex Integration Centers Overview

The Teraplex Integration Centers are facilities to investigate, research, and prove very large data warehouse implementations. Since 1996, IBM has invested USD 75 million to create Teraplex Centers for AS/400, Netfinity, NUMA-Q, RS/6000, and S/390. The Teraplex Centers are used to integrate and test IBM hardware and software, along with products developed by IBM Business Partners, for very large, end-to-end customer solutions. The execution of large-scale proofs of concept use real data on very large-scale configurations.

Since their inception, IBM's Teraplex Centers have brought value to IBM customers and Business Partners with their proofs of concept to:

- Minimize risk. After the Teraplex created a data warehouse for a very large international bank, using the bank's data, the bank was confident in their choice of an IBM solution for their 10 TB data warehouse.
- Prove the scalability of IBM systems. A large U.S. insurance company took advantage of Teraplex expertise to verify that DB2 UDB Enterprise - Extended Edition runs on a very large system – 332 nodes – which can accommodate their expected growth.
- Share skills to improve performance. A large retail chain participated in a proof-of-concept test of their data warehouse on new hardware and software. They were able to improve performance dramatically by testing the new Automated Summary Table feature in DB2 UDB (most queries ran ten times faster; several queries ran 48 times faster).
- Stress test IBM Business Partner software to resolve scalability issues before systems are shipped. Premier Business Intelligence software developers such as BMC, Coglin Mill, Hyperion, MicroStrategy, and Visions Solutions have used the Teraplex Centers to help ensure that their solutions function when pushed to a very large scale.

Each Teraplex Center is outfitted with several of the most powerful systems of their type. For example, the RS/6000 Teraplex Integration Center can put together configurations of hundreds of nodes and many terabytes of storage. The physical assets of each Teraplex Center are continually updated to stay at the crest of the technology curve.

IBM's Teraplex Integration Centers represent the largest, most powerful dedicated integration and testing facilities in the world.

Further information about IBM Teraplex Integration Centers is available at:
http://www.ibm.com/solutions/businessintelligence/teraplex/index.htm

DB2 Universal Database Overview

IBM's DB2 Universal Database (DB2 UDB) for Unix, Windows, and OS/2 is a relational database management system that is fully Web-enabled and scalable from single processors to symmetric multiprocessors to massively parallel clusters. DB2 UDB features full multimedia capabilities with image, audio, video, text, and other advanced object relational support, making it capable of handling leading-edge applications.

DB2 UDB demonstrates its strengths through support of business intelligence applications such as data warehousing and online analytical processing (OLAP). DB2 UDB's parallel database technology and query optimization can help you and your customers achieve competitive advantage, better customer service, and reduce costs by analyzing available data for the knowledge required to make better decisions. DB2 UDB can be used across your enterprise for all data management requirements.

Other important aspects of DB2 UDB include:

- Ease of use and management. DB2 UDB is one of the easiest databases in the industry to set up, use, and manage. It includes a suite of GUI administration tools for easy installation, administration, and remote operations. It includes programmer-friendly tools to get applications created and in use quickly and user-friendly tools to make users immediately productive. All you need to get started and working comes in the box, ready to go.

- Bullet-proof reliability. DB2 UDB is setting the standard for quality and reliability in the client/server database industry. As more mission-critical applications are being developed, IBM's ability to bring mainframe-level reliability to this environment has become a major selection factor for customers in choosing DB2 UDB. Better reliability and availability can reduce your costs, while scalability both within and across operating systems can reduce the risk of dead-end projects.
- Multimedia extensibility. DB2 UDB allows you to extend the capabilities of the database to meet your specific organizational requirements. This includes the ability to support more advanced applications involving multimedia data such as documents, images, audio, and video. You can now develop applications that use technology to gain competitive advantage in a way not previously possible or practical and you can save costs by extending DB2 UDB to support these new applications.
- Complete Web enablement. One of the key new areas for application development is in e-business. DB2 UDB is fully integrated with web technology so that data can be easily accessed from the Internet or from your business intranet with complete security. This allows you to quickly build e-business or Internet applications that can provide competitive advantage, greater customer service, and reduced costs.
- A universe of partner solutions. DB2 UDB is capable of supporting a broad range of data management applications with excellent performance and reliability. As a result, a large number of industry solution providers, including Siebel, SAP, Baan, PeopleSoft, and thousands more, have adopted DB2 UDB to support their applications or adapted their tools to support DB2 UDB. This allows you to quickly satisfy your requirements by buying rather than developing solutions.
- Universal access. DB2 UDB can be accessed from almost any client workstation over almost any network. Through built-in data replication and distributed transactions, DB2 UDB provides you with the flexibility of placing data anywhere in your network to meet your requirements for optimum service and productivity. Further, DB2 UDB provides the most efficient and seamless integration of data on mainframe and midrange data servers in the industry, allowing you to reduce costs and improve cycle times by leveraging your current investments in data, hardware, software, and skills.
- Multi-platform support. DB2 UDB is one of the most open database platforms available. DB2 UDB runs on the most popular Unix, Windows, and OS/2 operating systems including AIX, HP-UX, Solaris Operating Environment, OS/2, Windows NT, Windows 2000, and Linux. DB2 UDB supports all major industry standards relevant to distributed data, so that it can be accessed using thousands of existing tools and applications, and can be managed easily within an open, network computing environment. IBM is committed to continuing to provide a superior, open-architecture, full-function database, with consistency at the application programming level between all DB2 Family members running on all major industry operating systems. (These operating systems include: OS/390, VM/VSE, AS400, AIX, Solaris Operating Environment, HP-UX, Windows NT, Windows 2000, OS/2, and SCO.) These capabilities allow you to reduce costs and improve cycle times by leveraging your current investments in data, hardware, software, and skills.

Further information about DB2 UDB is available at: **http://www-4.ibm.com/software/data/**

Tivoli Storage Manager with Tivoli Space Manager Overview

Tivoli Storage Manager (TSM) is the base product of an enterprise-wide network storage management solution, IBM's Tivoli Storage Management product set. TSM Version 3.7 is the successor of Tivoli ADSTAR Distributed Storage Manager (ADSM) Version 3. Just as ADSM did, TSM focuses on delivering a complete business solution rather than on solving only the data backup and recovery function. It provides first level data protection (backup and restore), file archiving, and file retrieving for long-term data retention and central administration.

In addition to TSM, the Tivoli Storage Management solution product suite includes complementary products and Data Protection products. One of the complementary products, Tivoli Space Manager, is the focus of this study. Tivoli Space Manager is the former ADSM client product Hierarchical Storage Manager (HSM). Like its HSM predecessor, Tivoli Space Manager is used for hierarchical space management and efficient use of online storage; it is this attribute that attracted the attention of the IBM DB2 UDB/Teraplex team.

Note: When this paper refers to *HSM*, it means either Hierarchical Storage Manager for those who are familiar with ADSM; or, Tivoli Space Manager for those who are familiar with Tivoli Storage Manager (TSM). When you read *HSM* in this paper, it refers to both of these products.

Tivoli Space Manager uses hierarchical storage management to transparently migrate rarely accessed files to the TSM storage repository. This delivers the following benefits:

- Frees administrators and users from manually pruning a file system
- Lets user applications recall migrated files by simply referencing them
- Ensures that sufficient free storage is available at the workstation or server system
- Allows administrators to take advantage of less expensive storage alternatives

Tivoli Space Manager file migration is not a backup and restore process. When Tivoli Space Manager migrates a file from online storage to offline storage, it leaves a place-holder in the online storage called the *stub file*. This file is relatively small (the minimum stub file size is 511 bytes) and it contains information necessary to locate and recall the migrated file, including some leading bytes of the migrated file. In addition, the stub file contains information that Tivoli Space Manager can use to respond to specific operating system commands such as *ls*, *find*, *df*, and *du*, without recalling the file. Similarly, DB2 UDB also can use information found in the stub file that will assist in the retrieval of data without recalling the entire file.

The advantage of all this is that, to the database systems, files appear to reside on the online file system. To users, the recall is also transparent: users simply access the file the same way they do an online file.

Tivoli Space Manager migrates files to *storage pools*, which are a named set of storage volumes that TSM uses to store data. The *volume* is the basic unit of storage and can be a tape cartridge, optical cartridge, or disk device. Storage pools can be defined for these different device types, thus allowing great flexibility in designing storage alternatives. See Figure 1 on page 6 for the relationship between volumes and storage pools.

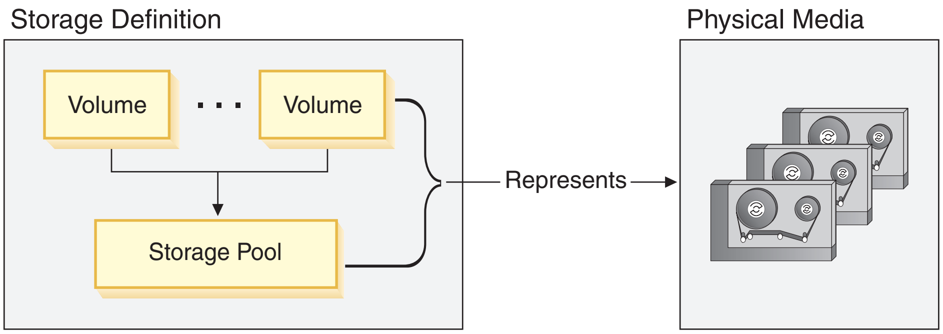

Figure 1. Volumes and Storage Pools

Tivoli Space Manager Migration and Recall

Tivoli Space Manager provides options in the way files are migrated and recalled, giving database administrators added control.

You can select one of two ways to have Tivoli Space Manager migrate files: automatically or selectively.

If you choose automatic migration, Tivoli Space Manager monitors space usage under its control and automatically migrates files when it detects the need to free up space on the local file system. There are two options with automatic migration: *threshold migration*, which maintains a predefined level of free space on a local file system using high and low thresholds; and *demand migration*, which responds to an out-of-space condition on a local file system.

With automatic migration, migration is not under your direct control; whereas with selective migration, migration is under your direct control. That is, if you choose selective migration, you can specifically select one or more files to be migrated by name.

Tivoli Space Manager also allows you to select how files are recalled: either transparently or selectively. The transparent option automatically recalls a migrated file back to a local file system when the file is accessed; it remains in online storage for as long as needed. With the selective option, the administrator can request that a specific file or group of files be recalled.

Note: Tivoli Space Manager migration is not a replacement for backup. Migration frees space on your local file system. Unlike file backup, migration does not protect against file deletion, file corruption, or disk failure. You should continue to back up your important files whether they reside on your local file system or are migrated to TSM storage.

Criteria for Migrating Files: Tivoli Space Manager migrates only regular files and does not migrate character special files, block special files, FIFO special files (named pipe files), or directories. For more general criteria for automatically migrating files, see “Hints and Tips” on page 16.

Further information about Tivoli products is available at: **http://www.tivoli.com/**

How DB2 Universal Database and Tivoli Space Manager Work Together

A DB2 instance owner can use selective migration to migrate historical data files and the associated index files to TSM storage, thereby making space for new tables and indexes.

In a DB2 environment, selective migration, or the option to determine the files Tivoli Space Manager moves to storage pools, allows the instance owner to migrate data files of tables and indexes that are infrequently accessed such as historical data. This can be done by specifying file names or container path names to the **dsmmigrate** command.

To enable space management for any file systems used by DB2, you configure Tivoli Space Manager without having to change any DB2 settings. DB2 itself does not directly interact with Tivoli Space Manager. When DB2 attempts to access a data file, if that file is resident on the local file system, DB2 has direct access to the file. If that file is migrated, HSM automatically recalls the file from TSM storage.

Depending on the recall mode in effect when DB2 closes the file, the file remains in the local file system which will be eligible for future migration, but it does take up space. If the migrate-on-close mode is enabled, HSM changes the file back to the migrated state (replacing it with a stub file), thereby freeing up space immediately. In the latter case, the file is recalled only for the duration it is needed.

The stub file kept on disk can be of variable size. DB2 can keep some important file control information in the stub file. Then, instead of recalling the entire file, the stub can be used to access the needed information.

The migrate-on-close mode is effective only if a query reads, but does not update, any rows. Tables in a data warehouse are typically read-only.

Basically, selective migration gives the database administrator fine control, and transparent recall gives the users of the database easy access.

DB2 allows tables to span multiple files. Only the files required by an application are recalled when necessary.

Frequently with historical data, you may want to process only selected periods of time over many years. For example, you may want to process all months in a previous year or only the first quarter of the past five years. If you have requirements like this, you should consider how you divide your data into the tables needed for your applications. DB2 supports the selective processing of large logical tables (made of multiple smaller tables) that are presented as a single logical entity to the end user or application. By defining constraints on each of the smaller tables, the DB2 optimizer can eliminate processing of those files that are not required for a query. By storing tables in groups of files, and by selecting appropriate table space naming conventions, you can easily set rules that protect the most active data from migration. Similarly, it then becomes easy to set the rules to enforce migration.

A single DB2 UDB Enterprise - Extended Edition instance can work with many TSM nodes. Each of the database partitions interfaces with a single TSM, but not all data partitions need to interface with the same TSM. Given that there is a migration and recall rate for a TSM server, you can scale to the desired rate by implementing multiple TSM servers.

Set Up and Configure Tivoli Space Manager

To set up space management, perform the following tasks:

1. Install the HSM client code and TSM server code
2. Configure the client system options file (dsm.sys) and the default user options file (`dsm.opt`)
3. Define management class on server
4. Enable space management to each file system

You may also optionally configure the include-exclude options file to include or exclude specific files from space management and assign specific management classes to files.

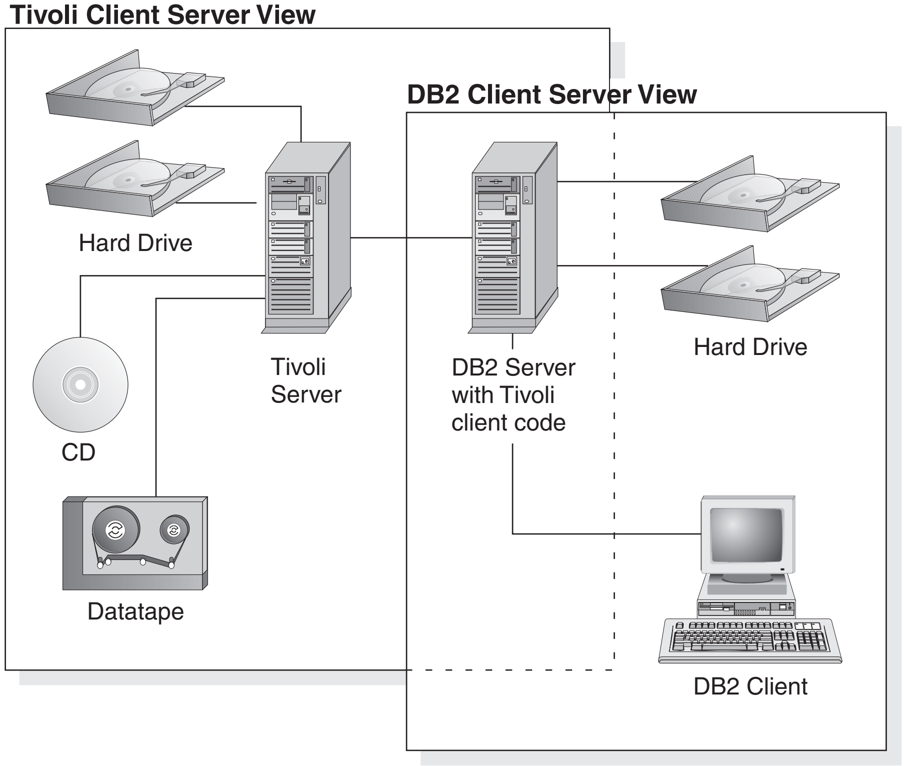

Figure 2. Tivoli Server and Clients; DB2 Server and Clients

As shown in Figure 2, DB2 is a server to DB2 clients but is itself a client of the Tivoli server. This is important when performing the following tasks related to the installation and configuration of the environment shown in the previous figure.

Client Installation (on DB2)

Install the following components in the ADSTAR Distributed Storage Manager (ADSM) Version 3.1 client set:

- Hierarchical Storage Management Client
- Application Programming Interface
- Backup-Archive Client

Note: The ADSM Version 3.1 client set is compatible with the TSM Version 3.7 server. We chose to use the earlier version because at the time of project, the new HSM client was not generally available.

If you only want to migrate or recall DB2 files, only the HSM client is required. The ADSM API is used by DB2 UDB to request backup/restore services. The Backup-Archive Client is used to back up or archive files not used by DB2.

Tivoli Server Installation

Install the Tivoli Storage Manager (TSM) Server Version 3.7. This server provides backup and restore, archive and retrieve, and migrate and recall services to registered clients.

Install the Tivoli Storage Manager Administrative Client Version 3.7. You can install it on any workstation from which administration or activity monitoring is performed.

Configuring Options Files

The ADSM Backup-Archive Client and the HSM Client use the same set of options files. Most of the configuration parameters are common to both client sets; however, a small set of parameters is specific to each client. For a complete list of options grouped by categories, see *ADSTAR Distributed Storage Manager Using the UNIX Hierarchical Storage Management Clients*, SH26-4083. Or, you can see *Tivoli Space Manager for UNIX Using the Hierarchical Storage Management Clients Version 3 Release 7*, SH26-4098.

Client System Options File: Use the client system options file to specify the following options:

- One or more servers to contact for services
- The communications options for each server
- Authorization options
- Other options

The options defined in this file apply to all users on a client workstation.

The following is a sample system options file (`dsm.sys`) with options for the HSM client:

```
DEFAULTSERVER server_a
MIGRATESERVER server_a
CHECKTHRESHOLDS 2
RECONCILEINTERVAL 12
MAXRECONCILEPROC 5
MAXTHRESHOLDPROC 5
MIGFILEEXPIRATION 10
MINRECALLDAEMONS 5
MAXRECALLDAEMONS 15

SERVERNAME  server_a
COMMMETHOD tcpip
TCPPORT 1500
TCPSERVERADDRESS aServerName.aCompany.COM
⋮
INCLEXCL /usr/lpp/adsm/bin/dsm.inclexcl
```

All of the following space management options have default values. If different values are preferred, they must be placed before the first server stanza in the system options file. Once this file is updated, the space monitor daemon and recall daemon that are currently running must be refreshed to make the changes effective.

MIGRATESERVER: specifies the single TSM server to which each HSM client can migrate files. However, backup and archive can still go to multiple servers.

CHECKTHRESHOLDS: specifies the frequency for the space monitor daemon to check space usage on HSM-controlled file systems.

RECONCILEINTERVAL: specifies the frequency for HSM to reconcile files in local file systems and the associated copies in TSM storage.

MAXRECONCILEPROC: controls the maximum number of reconcilication processes that the HSM program can start at one time.

MAXTHRESHOLDPROC: controls the maximum number of threshold migration processes that the HSM program can start at one time.

MIGFILEEXPIRATION: specifies the number of days copies of the migrated or premigrated files are kept on the TSM server after they are modified or deleted from local file sytems.

MINRECALLDAEMONS and MAXRECALLDAEMONS: control the minimum and maximum concurrent recall daemons to perform recalls for the client node.

For detailed information on client system options, see *ADSTAR Distributed Storage Manager Installing the Clients*, SH26-4080.

Client User Options File: A client user options file contains the following types of options:

- Node and server options
- Backup and archive options
- Restore and retrieve options
- Format and language options
- Command and error processing options
- Information availability option

The following is a sample default user options file (`dsm.opt`):

```
SERVERNAME server_a
TAPEPROMPT no
SUBDIR yes
OPTIONFORMAT standard
```

If an option is not specified in the user options file, its default value is assumed. Most of the options in the user options file affect the Backup-Archive Client only. Some affect both the Backup-Archive Client and the HSM client.

TAPEPROMPT: controls if a user will be prompted to wait for the media to be mounted if the migration destination is a storage pool consisting of removable media.

OPTIONFORMAT: determines the format in which the HSM commands are entered in the command line.

Options specified in the user options file can be customized for each individual user. A system-wide default user options file should be configured by the system administrator. Customized user options files can be stored anywhere that a user can access.

For detailed information about client user options, see *ADSTAR Distributed Storage Manager Installing the Clients*, SH26-4080.

Client Include-Exclude Options File: An include-exclude file is used to exclude a specific file or groups of files from backup and space management services. By default, all files in the file systems that are managed by the Tivoli Space Manager are implicitly included for migration, subject to other migration eligibility criteria such as age and size. However, some files, such as DB2 tag files and files created by HSM for its own use, should not be migrated from local file systems. Files that are accessed frequently should also be kept locally to allow efficient access.

Defining an include-exclude file is optional. When defined for backup, you can define a separate include-exclude options file for each TSM server. When defined for migration, only the include-exclude file associated with the destination server is referenced, since the client can only migrate files to one server. In any case, an include-exclude file must be readable by all users.

For detailed information on how to create an include-exclude options file, see *ADSTAR Distributed Storage Manager Installing the Clients*, SH26-4080.

For detailed information on how the include-exclude option may affect the behavior of HSM, see *ADSTAR Distributed Storage Manager Using the UNIX Hierarchical Storage Management Clients*, SH26-4083.

An include-exclude options file is processed by any TSM client commands (including HSM commands) from the bottom up.

The following is a sample include-exclude file (`dsm.inclexcl` specified in `dsm.sys`) for the purpose of migrating DB2 data files and index files:

```
exclude /fs[123]/.../*
include /fs[123]/comps/li1_9[2345]/SQL*.DAT
include /fs[123]/comps/li1_9[2345]/SQL*.INX
```

In this example, there are three file systems that are under space management: `/fs1`, `/fs2`, and `/fs3`. The DB2 storage containers are defined in these file systems. The DB2 tables are divided by month. For example, in the data shown in more detail later in this paper, `/fs1/comps/li1_92` is one of the containers that stores the January 1992 table. This options file specifies that only the DB2 data files with the file extensions .DAT or .INX are eligible for migration.

Different mangement classes can be assigned to specific files or groups of files. This is done by creating multiple include options in the include-exclude file. In the example above, it uses the default management class registered to the client on the TSM server. For detailed information about using the include option to assign management classes to files, see *ADSTAR Distributed Storage Manager Using the UNIX Backup-Archive Clients*, SH26-4075.

Assigning Management Classes to Files

Space management attributes are set in the definitions of TSM management classes. A management class is bound to each file migrated to a TSM server. No special options are needed in the server options file, `dsmserv.opt` to be able to use HSM.

Here is a list of space management attributes in a management class:

- Space Management Technique: which determines if a file is eligible for automatic and selective migration, selective migration only, or neither
- Auto Migrate on Non-usage: which specifies the number of days that must elapse since the file was last accessed before it is eligible for migration

- Backup Required Before Migration: which determines if a backup version of the file must exist on the TSM server before it is eligible for automatic or selective migration
- Destination for Migrated Files: which specifies the TSM storage pool in which the migrated file is stored

For a detailed description of these attributes, see *Tivoli Storage Manager for AIX Administrator's Reference*, GC35-0369.

Here is a TSM command that defines a management class that, in turn, defines space management attributes:

```
define mgmtclass dwdomain db2policy optmclass spacemgtechnique=automatic
migrequiresbkup=no migdestination=opticalpool
```

This command defines a management class named `optmclass` in the policy set `db2policy` in the domain `dwdomain`. This command allows files to be migrated automatically or selectively. Also having a current backup version is optional. The migrated files go directly to a storage pool which uses removable optical media. For the relationship between the device class, the library, and the drives, see Figure 3.

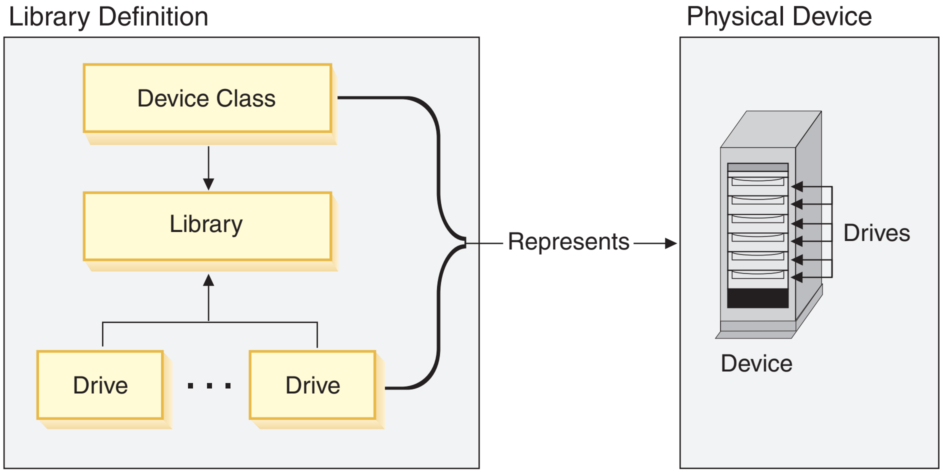

Figure 3. Device Class, Drives, and the Library

Figure 4 on page 13 summarizes the relationships among the physical device environment, TSM storage and policy objects, and clients.

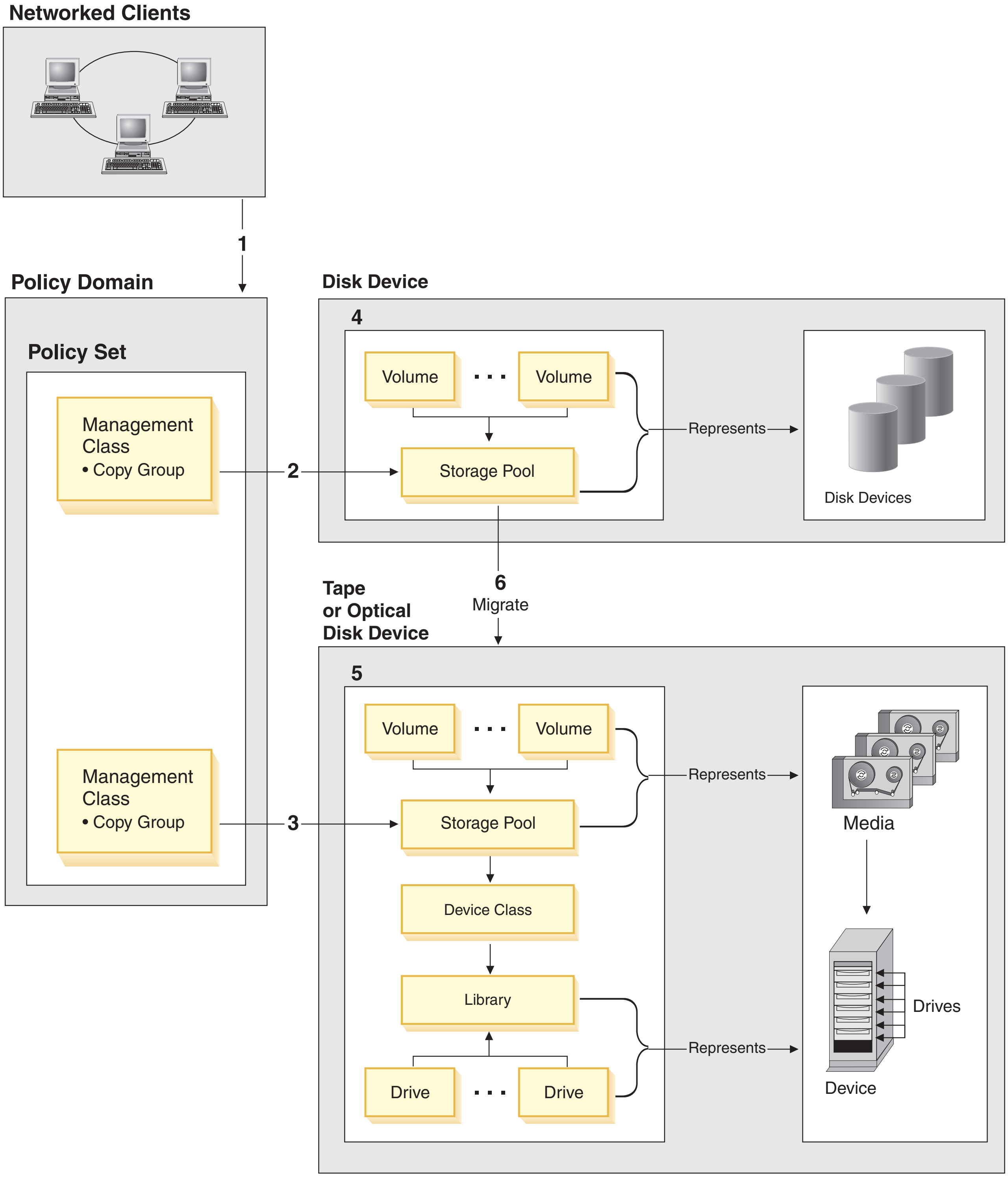

Figure 4. Overall Picture of the TSM Environment

The numbers in the following list correspond to the numbers in the figure.

1. When clients are registered, they are associated with a policy domain. Within the policy domain are the policy set, management class, and copy groups.
2. When a client backs up, archives, or migrates a file, it is bound to a management class. A management class and the backup and archive copy

groups within it specify where files are stored and how they are managed when they are backed up, archived, or migrated (for space-managed files).

   In this case, the files are placed on a disk device.

3. At this point, the files are placed directly on alternate storage devices.
4. Storage pools are the destinations for backed-up, archived, or space-managed files. Copy groups specify storage pools for backed-up or archived files. Management classes specify storage pools for space-managed files.

   Storage pools are mapped to device classes, which represent devices. The storage pool contains volumes as indicated by the device type associated with the device class. For example, a storage pool that is mapped to a device class with a device type of 8mm contains only 8mm tapes.

   All devices require a device class that specifies at least a device type. No library and drive information is required for disk devices.
5. The storage pool information is the same as for disk devices except that tape and optical devices also require a library and drive for management of media, including the mounting of that media.
6. If the pools are set up in a storage hierarchy, then HSM migrates files initially stored on disk storage pools to tape or optical disk storage pools.

Enabling Space Management to File Systems

A file system must be mounted before space management can be added to it. In AIX, the file system migrator (FSM) must be mounted over the AIX file system (JFS) in order for HSM to provide space management services. When space management is added to an AIX file system (JFS), FSM is mounted automatically. The FSM is a kernel extension that intercepts any file system I/O operations and provides support for space management as needed. If a file is migrated, HSM automatically recalls it back to the local file system using the recall mode that is in effect. If the file is not migrated, it is accessed locally by AIX.

For details in enabling and disabling space management, see *ADSTAR Distributed Storage Manager Using the UNIX Hierarchical Storage Management Clients*, SH26-4083. You can enable space management with the HSM graphical user interface (**dsmhsm**) or command line (**dsmmigfs**). The command-line approach is used in the examples.

When adding space management to a file system, consider the following basic space management settings:

- HSM stub file size
- Minimum file size for migration
- The order in which files are migrated
- Quota for migration and premigration
- Migration thresholds

HSM uses stub files to maintain information necessary to locate and recall a migrated file, and to respond to operating system commands such as **ls**, **find**, **df**, and **du**, without recalling the file. Depending on the size allowed for stub files, a stub file can contain leading bytes of data from the original file. This is useful if the data found at the beginning of the file is read frequently and not modified.

HSM does not migrate a file unless doing so actually saves space on the local file system. For a file to be considered for migration, its size must be greater than the stub file size plus one byte, and greater than the file system block size.

The order in which eligible files are migrated during automatic migration is determined by the age and size factors. The values associated with the age and size factors are set by you to reflect the priority you place on age relative to size. These factors are used to determine the priority score for each file. HSM uses the concept of priority score, which is the sum of (i) the product of the age of a file and the age factor, and (ii) the product of the size of a file and the size factor. HSM prioritizes files for migration in decending order by priority score.

A *quota* in HSM context is the amount of data that can be migrated and premigrated from the space managed file system to TSM storage. By default, this is the size (in megabytes) of the file system itself. Consideration should be given to the projected growth of the file system and restrictions on TSM storage.

The values established for the *migration thresholds* control when migration of files starts and stops. The high threshold determines when HSM starts threshold migration. It continues migrating files until the space utilization drops to the low threshold. These thresholds are used to initiate automatic migration. In a DB2 UDB environment, it is suggested that you use the selective migration technique to precisely control what files are moved to TSM storage. It is a very good idea to engage automatic migration and to set thresholds to relatively high values (for example, a low threshold of 90 percent and a high threshold of 95 percent), so that the migration does not start very often. This can be used as a safety measure in the unlikely event of a system crash due to infrequent space shortage. See Figure 5 for how thresholds are used to migrate files.

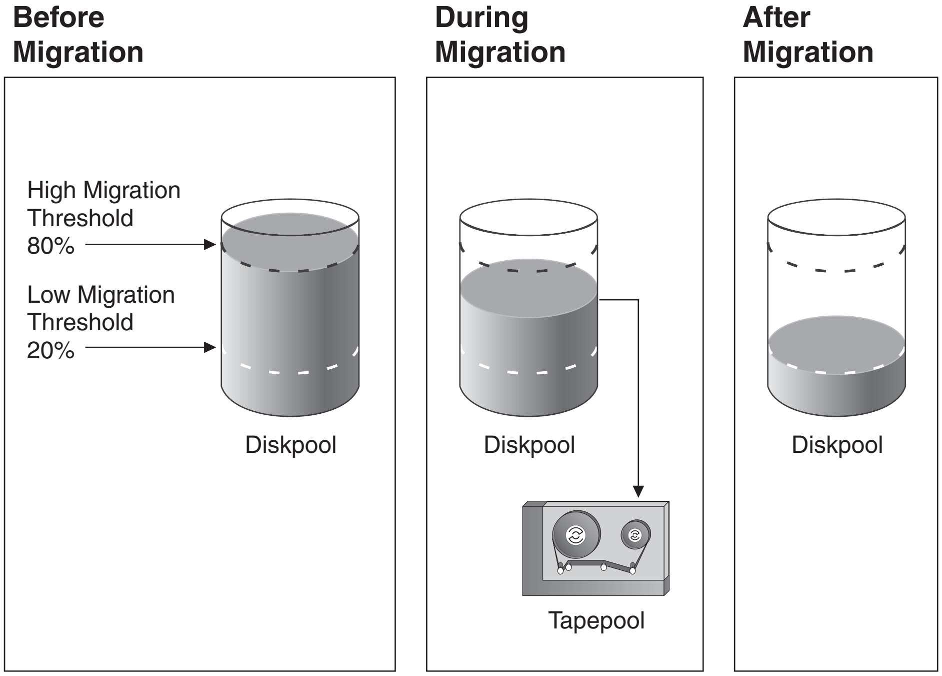

Figure 5. Threshold Migration

For example, use the following command to enable space management for a file system named `/scratch` in which the migration eligibility is determined solely based on the age factor:

```
dsmmigfs add -agefactor=1 -sizefactor=0 /scratch
```

Note: In the test scenario shown later in this paper, threshold migration is not used because it is not needed in a testing situation. Selective migration is the only migration method used for the purposes of the test scenario. Selective migration provides the control needed to carry out migration testing found in the scenario shown later. In your business environment, you may want to use threshold migration and so this method is presented here to show you how this could be done.

When space management is added to a file system, HSM performs the following tasks:

- Creates a hidden directory named `.SpaceMan` in the file system to store a list of space management objects.
- Adds an entry for the file system in the `dsmmigfstab` file in the `/etc/adsm/SpaceMan/config` directory.
- Changes the mount attribute for the file system in `/etc/filesystem` to false. Adds an `adsmfsm` attribute with a value of *true* to indicate that the file system is to be managed by HSM.
- Mounts the FSM over the native file system, the JFS.
- Activates space management for the file system.

Update space management attributes by activating or deactivating them at any time.

For more details on using the **dsmmigfs** command to manipulate space management options, see *ADSTAR Distributed Storage Manager Using the UNIX Hierarchical Storage Management Clients*, SH26-4083.

Hints and Tips

To be migrated either selectively or automatically, a file must be:

- A regular file.
- A resident (not migrated or modified) file, or pre-migrated file (migrated but not replaced by a stub file).
- Not excluded from space management with an **exclude** or **exclude.spacemgmt** option in the include-exclude options file.
- Greater in size than both the size specified for stub files plus one byte, and the file system block size.
- Meet the following management class requirements:
  - A current backup version of the file exists if the management class requires it.
  - For automatic migration only, the associated management class assigned (which is set up by TSM) must allow automatic migration.
  - For automatic migration only, the file must not have been accessed for at least the number of days specified in the management class.
  - For selective migration only, the associated management class assigned (which is set up by TSM) must allow either selective or automatic migration.

Set Up and Configure DB2 Universal Database

There are several key recommendations when setting up and configuring DB2 UDB:

- Only the use of SMS containers is supported when setting up and configuring this environment.
- The division of data is important. In the scenario used in this paper, data was divided by month and year so that all of the data for one month of a particular

year was kept together in a table. You could also group the data by quarter. How you divide the data is determined by how you are most likely to retrieve the data in the future.

- The size of the stub file must allow the first extent to be local.

  DB2 Universal Database needs to know some information that is kept on page zero of the .DAT file. This is not a problem where, for performance reasons, you have enabled DB2 UDB to carry out "big-block reads."

  Big-block reads allow DB2 UDB to move data one extent at a time.

  In the scenario used, the largest extent was 32 pages. To ensure that "migrate-on-close" works, the first 32 pages of each migrated file must be kept together. With 4 KB per page and 32 pages, this meant that 128 KB was needed. However, the stub file also requires some space. The stub size must be a power of 2; therefore, since 128 KB was not sufficient, 256 KB was the size used.
- Determining which files should or should not be migrated is very important. DB2 tag files (that is, the SQLTAG.NAM files in each container directory) should not be migrated because they are used by the database manager to verify that the database is complete and consistent when a connection is made. These files are 512 bytes in size, which is smaller than the typical file system block size (4096 bytes). As a result, these files are not elligible for migration.

  Some DB2 files should be considered for migration. These include data files (with the .DAT extension) and index files (with the .INX extension).
- Volatile data files (such as the catalog files or others that are updated frequently) should not be considered for migration. You should consider moving these files by placing their table spaces in their own file system or set of file systems.

Test Scenario and Results

Here are the sections that have the details you might be interested in:

- Testing Environment table found below
- "Test Scenario" on page 19
- "Time-to-Migrate Test Data and Test Results" on page 19
- "Query Times on Migrated Data" on page 22

The test environment was constructed using the following hardware and software configurations.

Table 1. Testing Environment

Machines by Function	Number of Nodes	Hardware	Software
Tivoli Storage Manager Server	1	4-way 332MHz Silver wide node 3 GB RAM, 4 x 9.1 GB internal disks 4 3590-B11 tape drives 1 3995-C66 optical library	AIX 4.3.3.+ Maintenance Level 2 TSM Server 3.7.2 TSM Client 3.7.1 (admin client) Atape 5.0.2

Table 1. Testing Environment (continued)

Machines by Function	Number of Nodes	Hardware	Software
DB2 Database Servers	12	4-way 332MHz Silver thin node 3 GB RAM, 2 x 9.1 GB internal disks 16 4.5 GB SSA external disks spread over 2 SSA-80 adapters, each adapter with 1 loop of 8 disks	AIX 4.3.3.+ Maintenance Level 2 ADSM Client 3.1.20.7 (see Note) DB2 UDB EEE Version 7.1
Note: The HSM Client is not included in the TSM Client Version 3.7 product set. We used the ADSM 3.1 clients, which are compatible with the TSM 3.7 server.			

Figure 6 shows the different hardware devices used in the testing environment described in the table above. The different types of hardware are shown but the number of devices used in the test configuration was not the same as presented in the table above. For example, the Tivoli Storage Manager Server described in the table requires one 3995-C66 optical library. This hardware device is shown accurately; only one is required and only one is shown. Also, the DB2 Database Server requires two SSA-80 disk adapters. However, only one is shown in the following diagram.

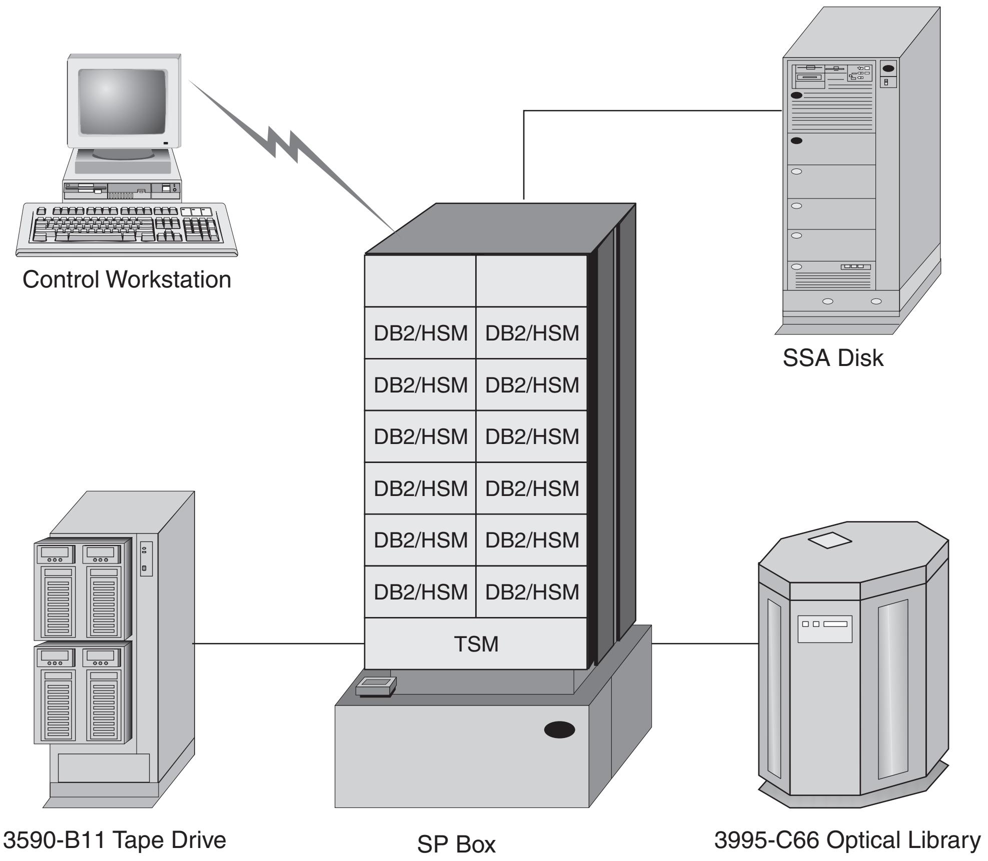

Figure 6. Configuration Used to Test

Test Scenario

A database was created over 12 nodes. The database was made up of business data covering seven years. Each month of those seven years was placed into a separate table for a total of 84 tables. Indexes were created on the tables. Each table was created with a constraint (enforced by DB2) to ensure that it only had data for the valid month. Views were also created on the tables; for example, a view of the union of all monthly tables and a view that is union of all yearly views.

Note: Details on the creation of the tables, indexes, constraints, and views is shown in an example found in "Appendix: A Simple Example" on page 29.

Each table was placed in its own SMS table space. Each table space was created with access to three containers.

The actual business data was placed into .DAT files that were each approximately 28 MB in size. The index was placed into .INX files that were each approximately 1.1 MB in size.

Testing was done using two different types of storage devices under TSM. In the first case, six 3995 rewriteable optical drives were used so that two nodes were mapped to each device. In the second case, four 3590 tape drives were used so that three nodes were mapped to each device. The difference in number of the two types of storage devices does not allow for direct comparison of the retrieval speed of data that has been migrated.

How Tivoli Space Manager recalls a migrated file is controlled by changing the *recall mode*. The mode used for testing was "migrate-on-close". This mode temporarily recalls a migrated file back to its original file system. The recalled file remains on your local file system only for as long as the file remains open. When the file is closed, if it has not been modified, Tivoli Space Manager replaces it with a stub file on the local file system, and it once again becomes a migrated file. There is no need to send the file to TSM through Tivoli Space Manager because the file has not been modified and the copy that is stored in TSM is still valid.

Another mode is "normal". This mode recalls a migrated file back to its original file system.

The only other choice is a mode called "read-without-recall". The migrated file is not recalled but only copied, caching the file information in a memory buffer at your workstation. No modification of this file is allowed.

Use the **dsmattr** command to set or display the recall mode for one or more migrated files. You cannot set a recall mode for a resident or premigrated file. A migrated file retains its recall mode setting only as long as the file remains migrated.

Time-to-Migrate Test Data and Test Results

For migration testing done for this paper, examples of the location of the migrated data are shown in Table 2.

Table 2. Paths of the Migrated Table Spaces

File System 1 (/fs1)	File System 2 (/fs2)	File System 3 (/fs3)
/fs1/comps/li1_92	/fs2/comps/li1_92	/fs3/comps/li1_92
/fs1/comps/li1_93	/fs2/comps/li1_93	/fs3/comps/li1_93

Table 2. Paths of the Migrated Table Spaces (continued)

File System 1 (/fs1)	File System 2 (/fs2)	File System 3 (/fs3)
/fs1/comps/li1_94	/fs2/comps/li1_94	/fs3/comps/li1_94
Note: The paths show migrated table spaces.		

There are three file systems (/fs1, /fs2, and /fs3) in which are created the table spaces. For example, the system containers /fs1/comps/li1_92, /fs2/comps/li1_92, and /fs3/comps/li1_92 are used to create table space li1_92_ts. The following SQL statement is used to create this table space:

```
CREATE TABLESPACE li1_92_ts MANAGED BY SYSTEM USING ('/fs1/comps/li1_92',
   '/fs2/comps/li1_92', '/fs3/comps/li1_92')
```

After the creation of the table spaces, create the tables to be placed into the table spaces. Using a table layout previously defined (and called monthlydata), a table can be created in the table space using a SQL statement such as:

```
CREATE TABLE li1_92 LIKE monthlydata IN li1_92_ts NOT LOGGED INITIALLY
```

The amount of data (in MB) migrated by node and file system is shown in the following table. The table heading numbers 00 to 11 are the partition numbers.

Table 3. Partitioned (Node) Data Migrated

Partition Numbers	04	05	06	07	08	09	10	11	00	01	02	03	Total (MB)
The partition numbers identify SP node numbers.													
No client compression.													
Total/ node (MB)	176.24	175.94	176.18	176.17	176.38	176.23	176.20	175.97	176.15	176.63	176.57	176.43	2115.09
Client compression.													
Total/ node (MB)	86.40	86.25	86.35	86.37	86.46	86.40	86.38	86.26	86.35	86.59	86.55	86.48	1036.84

The results of the migration testing are shown in the following tables.

Note: The time-to-migrate testing only considers the time required to move data out to the new alternative storage devices under TSM. It does not show the time to recall a subset of the data from the different devices. The testing of the time-to-run-queries on migrated data is presented in the next section.

The “Total” column shows the elapsed time executing the “dsh” command on the SP control workstation. The “dsh” command sends the HSM migration command to all the partitions (nodes) in parallel. The “dsh” command used was:

```
dsh "dsmmigrate /fs[123]/comps/li1_9[234]"
```

which is equivalent to

```
dsh "dsmmigrate /fs1/comps/li1_92 /fs1/comps/li1_93 /fs1/comps/li1_94
   /fs2/comps/li1_92 /fs2/comps/li1_93 /fs2/comps/li1_94
   /fs3/comps/li1_92 /fs3/comps/li1_93 /fs3/comps/li1_94"
```

Note: The pathnames shown in the command are separated by spaces.

Table 4. Time-to-Migrate Test Results Using Optical Devices

Partition	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**00**	**01**	**02**	**03**	**Total**
All times are in the mm:ss format.													
Optical 1	03:33	03:32	03:36	07:08	07:16	07:03	03:40	03:47	07:04	03:54	06:56	06:56	07:17
Optical 2	03:33	**12:39**	07:53	**12:42**	03:39	03:59	07:30	03:42	07:52	03:48	03:52	07:35	**12:44**
Optical 2A	03:33	07:39	07:53	07:42	03:39	03:59	07:30	03:42	07:52	03:48	03:52	07:35	07:44
The numbers in **bold** reflect a five minute retention timeout that was experienced during testing. The retention timeout period is configurable and is under your control. Mounting an optical platter takes about two or three seconds.													
Note: In each of these cases, optical storage devices were used and six drives were available. The only difference was that collocation and compression were turned on and off. • Optical 1. Used with collocation turned off, compression turned on. • Optical 2. Used with collocation turned on, compression turned on. • Optical 2A. Used with collocation turned on, compression turned on. The values have been **adjusted** to account for the five minute retention timeouts.													

Table 5. Time-to-Migrate Test Results Using Tape Devices

Partition	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**00**	**01**	**02**	**03**	**Total**
All times are in the mm:ss format.													
Tape 3	02:02	01:23	02:42	02:47	01:27	02:08	01:53	01:11	02:43	02:51	02:09	01:22	02:52
Tape 4	03:15	01:23	05:09	02:57	04:39	01:12	01:10	04:28	02:52	01:18	05:31	03:25	05:32
Mounting a 3590 tape device takes about twenty to thirty seconds.													
Note: In each case, 3590 tape storage devices were used with four devices available. The only difference was that collocation and compression were turned on and off. • Tape 3. With collocation turned off, compression turned off. • Tape 4. With collocation turned on, compression turned off.													

The times shown in the two tables are elapsed times for all migrated sessions. Recall that the scenario has a many-to-one client-device ratio. All of the migration sessions were started at approximately the same time, but because of the many-to-one client-device ratio, only one client could access a device at a time. The elapsed time for a waiting client reflects the duration the client waited in the queue. The times are shown in minutes and seconds (MM:SS).

The conclusions that may be derived from these results are based on comparing a 3995-C66 Optical Library to a drive in a 3590-B11 tape storage device. This type of comparison allows for some generalizations may be made but not definitive statements.

The transfer thoughput of an optical drive is slower than that of a tape drive. This means that if you want to migrate many large files, going directly to tape is faster than going to an optical disk.

Collocation causes marginal degradation in the performance of an optical drive, but significant degradation in the performance of a tape drive. The difference occurs because the mounting or unmounting of an optical disk requires very short mechanical movement, whereas the mounting or unmounting of a tape requires forward and rewind motions that take longer to finish. This means that if you want to further reduce the migration time, turn off collocation of a storage pool so that the number of media changes is minimized.

These two types of storage devices can be connected to form a storage hierarchy. You can first migrate files to storage pools that use 3590-B11 tape storage devices, then move the data from tape storage pools to storage pools that use 3995-C66 optical storage devices during off-peak hours. The migration in this environment is a function of Tivoli Storage Manager. You describe to Tivoli Storage Manager how you want this migration environment to work and then it manages the actual movement of the data. This approach to migration of the data can shorten the time it takes to migrate files, and yet produce an efficient way to access the files. When the data is on the optical drives, the optical drives exhibit random I/O characteristics which provide greatly reduced seek times for data when compared to sequential access on tape drives.

Query Times on Migrated Data

The results of the time taken to run queries on migrated data testing are shown in Table 6 on page 23.

Table 6. Testing Results

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Storage: optical Collocation: off	97.870 seconds	37.706 seconds	99.101 seconds	39.379 seconds	52.374 seconds	94.278 seconds	41.564 seconds	43.952 seconds	54.951 seconds
Storage: optical Collocation: on	474.763 seconds	37.455 seconds	428.882 seconds	39.995 seconds	449.928 seconds	419.336 seconds	43.578 seconds	44.240 seconds	51.791 seconds
Storage: tape Collocation: off	385.883 seconds	103.873 seconds	393.124 seconds	104.298 seconds	363.586 seconds	390.413 seconds	102.320 seconds	101.000 seconds	120.451 seconds
Storage: tape Collocation: on	1179.689 seconds	89.712 seconds	1219.245 seconds	87.329 seconds	1188.914 seconds	1208.922 seconds	122.425 seconds	119.629 seconds	161.354 seconds
Number of nodes accessed	12	1	12	1	12	12	1	1	for maximum performance, access one node per optical device
Number and sizes of files recalled per node	3 files of ≈28M each 1 file of ≈1.1M	3 files of ≈28M each 1 file of ≈1.1M	3 files of ≈1.1M each 1 file of ≈28M	3 files of ≈28M each 1 file of ≈1.1M	3 files of ≈1.1M each 1 file of ≈28M	3 files of ≈28M each 1 file of ≈1.1M	3 files of ≈28M each 1 file of ≈1.1M	3 files of ≈28M each 1 file of ≈1.1M	

Note:

The descriptions of the queries are shown here:

- Q1:

```
select l_receiptdate, sum(l_quantity) from li1_93
   where (day(l_receiptdate) between 5 and 20)
   group by l_receiptdate
   order by l_receiptdate
```

- Q2:

```
select l_receiptdate, sum(l_quantity) from li1_93
   where (day(l_receiptdate) between 5 and 20)
   and (nodenumber(l_receiptdate)=0)
   group by l_receiptdate
   order by l_receiptdate
```

- Q3:

```
select l_shipdate, sum(l_quantity) from li1_93
   where (day(l_shipdate) between 5 and 20)
   group by l_shipdate
   order by l_shipdate
```

- Q4:

```
select l_shipdate, sum(l_quantity) from li1_93
   where (day(l_shipdate) between 5 and 20)
   and (nodenumber(l_shipdate)=0)
   group by l_shipdate
   order by l_shipdate
```

- Q5:

```
select count(*) from li1_93
   where (day(l_shipdate> between 5 and 20)
```

Note:

The descriptions of the rest of the queries are shown here:

- Q6:

```
select count(*) from li1_93
   where (day(l_receiptdate> between 5 and 20)
```

- Q7:

```
select l_receiptdate, sum(l_quantity) from li1_93
   where (day(l_receiptdate) between 5 and 20)
   and (nodenumber(l_receiptdate)=1)
   group by l_receiptdate
   order by l_receiptdate
```

- Q8:

```
select l_receiptdate from li1_93
   where (day(l_receiptdate) between 5 and 20)
   and (nodenumber(l_shipdate)=1)
   group by l_receiptdate
   order by l_receiptdate
```

- Q9:

  This query accesses one node per optical device by using “nodenumber” and temporary tables for individual nodes.

Note: The query testing used on migrated data considers the time required to query previously migrated data and recall a subset of that data from one of the new alternative storage devices under TSM. It does not show the time to migrate the data to the different devices. The testing results of the time-to-migrate data are presented in the previous section.

The time shown in this table includes the time to run the query plus the time to recall the selected subset of migrated data as defined by the query. In general, the time to run each query is approximately the same: around 10 seconds. As a result, valid conclusions about the recall time can still be made.

Note: The query run time might be different in your work environment due to the additional workload and the available resources of the database engine. The time taken to ship the result set to the coordinator node may also cause differences in the overall time. The testing results reported here were done in a controlled environment without any additional workload requirements.

When considering the results contained in this table, you should know that when using optical devices, six were used for twelve partitions; when using tape devices, four were used for twelve partitions. Therefore, any comparison between the two types of devices cannot be stated unequivically. It is possible to make some general comments that do seem to be borne out despite these differences.

The seek time for optical devices is less than the time required by tape devices. The mounting and positioning of the tape devices is considerably longer than the time required by optical devices.

With larger files, there is a throughput advantage when using tape devices. This needs to be balanced against the time required to mount, dismount, or rewind the tape device.

When considering the configuration used for testing in this white paper, for both optical and tape devices, collocation “off” is the better choice. However, if a 1:1 client-to-device ratio is used, collocation “on” is the better choice. This is true because each client has exclusive use of its drive and can access the volumes containing data belonging to that client.

Tips and Techniques

Several collections of tips and techniques are gathered here into separate sections that you might be interested in:

- “Network Optimization”
- “TSM and HSM Tuning” on page 25
- “Using Data Compression on the Client System” on page 25
- “Collocation on Sequential Access Storage Pools” on page 27
- “Application Performance” on page 27
- “DB2 UDB Tips” on page 27

Network Optimization

Our test environment consisted of a mixed workload – database and TSM backup – that exhibits different network characteristics. Sometimes tuning one environment may work against another environment. Therefore, it is necessary to tune the environment for the more important workload. In the example, since the database environment carries more weight than the TSM environment, the network options for DB2 are optimized, and TSM does its own socket tuning.

The following settings are recommended for a typical database environment on the RS/6000 SP platform running AIX 4.3.3. Place the following settings in the /etc/rc.net file:

```
sb_max=1310720
subnetsarelocal=1
ipforwarding=1
tcp_sendspace=262144
tcp_recvspace=262144
rfc1323=1
tcp_mssdflt=1448
tcp_pmtu_discover=1
ipqmaxlen=512
```

In an RS/6000 SP environment, use the SP switch for node-to-node communications, which carries database traffic as well as TSM traffic. The following settings are recommended for the switch adapter device:

```
sendpoolsize=16777216
recvpoolsize=16777216
```

TSM and HSM Tuning

Both the Tivoli Space Manager or HSM clients and the TSM server have to be tuned to take advantage of the network tuning done earlier.

On the client side, the following settings in the system options file (dsm.sys) are suggested:

```
TCPWINDOWSIZE 256
TCPBUFFSIZE 32
TCPNODELAY yes
TXNBYTELIMIT 25600
LARGECOMMBUFFERS yes
```

On the server side, the following settings in the server options file (dsmserv.opt) are suggested:

```
TCPWINDOWSIZE 256
TXNBYTELIMIT 25600
TCPBUFFSIZE 32
TCPNODELAY yes
BUFPOOLSIZE 32768
LOGPOOLSIZE 1024
TXNGROUPMAX 256
```

Using Data Compression on the Client System

An HSM client could be set up such that it is allowed to compress the migrating files in order to decrease the amount of data sent over networks and to reduce the space requirement in TSM storage. Data can be compressed by the HSM client (software) before it is sent to the TSM server, or by the storage device (hardware) on which the data is stored. Use either client compression or device compression, but not both.

Using client compression has the advantage of reducing load on the network, but it utilizes more CPU time and memory. In general, it prolongs the elapsed time associated with client operations.

By using device compression (if supported) it is possible to achieve a higher compression ratio than client compression. However, compressing data that has already been compressed may actually explode the size of the data.

In order to decide which type of compression to use, consider the following factors:

- The final storage device where the migrated files are stored.

  Different storage devices have different I/O characteristics. Disks are fast but cost more. Tapes are cheap, but access pattern is sequential. Optical disks are cheap, can perform random I/O, but have a lower data transfer rate.
- Whether compression is supported by the device.

  Not all storage devices support compression in their hardware assemblies. The 3590 tape drives support device compression which can be enabled or disabled by a system administrator. The 3995 optical library, on the other hand, does not support device compression.
- Network capacity.

  Is the data being sent over a high-bandwidth, low-latency network? If so, reducing network load may not be a pressing priority. If not, client compression may help relieve network congession.
- Processor speed and available memory.

  For client compression, does the client have the extra processing power and the memory buffers needed to compress data without significantly degrading overall system performance?

  For the server, if no compression is done by the clients, does it have enough buffers in memory to stage the data?
- Amount of space available in TSM storage.

  Is the TSM storage under tight constraints? Is saving space more important than saving time?

Putting it all together, find the slowest compoment and pick the compression strategy that is best suited for that environment. Can the client compress data fast enough to keep the network fully utilized? Can the network transmit the data fast enough to the TSM server? Can the TSM server receive from the network fast enough? Can the storage device write to media quick enough?

For example, in an RS/6000 SP environment using an SP switch to communicate between nodes, if the files are migrated directly to 3590 tapes, device compression produces the better performance as oppose to client compression. If the files are migrated directly to 3995 optical cartridges, client compression produces better performance as opposed to not doing compression at all – 3995 optical drives do not support device compression.

To enable client compression, add the following line to the server section in the system options file (`dsm.sys`):

```
COMRESSION yes
```

To disable client compression, change it to:

```
COMRESSION no
```

The TSM administrator has ultimate control over whether a client is allowed to use compression no matter what is set in the client system options file. This can be done through client registration.

For more information on compression issues, see *Tivoli Storage Manager for AIX Administrator's Guide*, GC35-0368.

Collocation on Sequential Access Storage Pools

Collocation is a concept applicable to sequential access storage pools in TSM. With collocation enabled, a TSM server makes its best attempt to keep files belonging to a single client or files belonging to a single file space of a client on a minimal number of volumes. A storage pool consists of volumes. A sequential volume could be a tape cartridge or an optical disk cartridge. The collocation attribute of a storage pool can be enabled or disabled by updating the storage pool definition in the TSM server.

By using collocation, there is a reduction in the number of volume mount operations required to restore, retrieve, or recall multiple files belonging to a client. Therefore, collocation has the benefit of reducing access time for these operations. However, in general, it uses more volumes because not all available space is fully utilized.

By disabling collocation, the server attempts to utilize all available space on each volume before using a new volume. This provides better utilization of space of each volume, but a client's files may scatter across multiple volumes, resulting in more volume mount operations and also longer access times.

It is a good idea to use a 1:1 client-to-drive ratio when using collocation. This reduces contention for access to drives so that all database clients can be served simultaneously.

For information on collocation, see *Tivoli Storage Manager for AIX Administrator's Guide*, GC35-0368.

Application Performance

The performance of database applications can be affected if the database files requested have been placed in alternative storage under Tivoli Storage Manager. Typically, it takes longer to access files stored on tape drives or on optical disk devices than those stored on high performance disk devices. Take this delay while accessing the files into account when sequencing and scheduling applications.

DB2 UDB Tips

The maximum database files open per application (*maxfilop*) database configuration parameter may need to be adjusted lower. When using SMS table spaces this parameter value is usually increased (as compared to the value for DMS table spaces). However, when using alternative storage with Tivoli Space Manager and TSM, consider using a lower value. This is particularly true when you have implemented threshold migration.

"Migrate-on-close" requires two environment variable changes. DB2_MMAP_READ and DB2_MMAP_WRITE must be set to "OFF" using the **db2set** command. This allows AIX caching of DB2 data read from JFS file systems into memory (outside the buffer pool).

For maximum performance, only one TSM storage device should be accessed by one TSM client session at a time. (The TSM client can also be a DB2 node.)

Note: Consider the use of the Query Patroller product if you want to maximize the number of client applications accessing the migrated data. For more information on Query Patroller, see *DB2 Query Patroller Administration Guide* and *DB2 Query Patroller User's Guide*.

Be selective in choosing the files that are migrated to the TSM storage devices. Use two key considerations to determine which files to migrate:

- The frequency of access of the file after it has been migrated. (There should be a low frequency.)
- The size of the file. (The size of the file should be large.)

Summary and Conclusion

This paper discusses how Hierarchical Storage Management (HSM) and DB2 databases work together to migrate rarely accessed data to Tivoli Storage Manager storage automatically and transparently. When using the cheaper alternative storage that is available through HSM, optical storage can retrieve small files rapidly, while tape storage is better suited for larger files.

If you are looking for a storage solution that assists you in managing increasing volumes of data, this paper shows that you can configure DB2 UDB and Tivoli Storage Manager to work together to provide such a solution. Migration of data to TSM is fast, depending on the number of devices used as alternative storage and the number of partitions in the database. Recalling a subset of the stored information is easy once the data is migrated to TSM. The speed of retrieval depends on the type of storage used and the amount of data being recalled. If you want to keep the cost of DASD in your environment down while maintaining the ability to access increasing volumes of historical business data, then DB2 UDB and Tivoli Storage Manager is a choice you should consider.

Appendix: A Simple Example

This appendix shows how to integrate DB2 Universal Database Enterprise Extended Edition and Tivoli Space Manager using a simple configuration. In this scenario, a standalone RS/6000 workstation is used to run both the DB2 server and the Tivoli Storage Manager (TSM) with Tivoli Space Manager server. The HSM client communicates with the TSM server using the loopback interface (127.0.0.1). The TSM storage is an 8mm tape drive.

The sample database consists of two partitions on a single physical machine. Partiton 0 has its table containers in filesystem `/fs0`, and partition 1 has its table containers in filesystem `/fs1`. Data is divided by month and is stored in monthly group tables. Each monthly table is created in its own table space in the IBMDEFAULTGROUP nodegroup, which spans both partitions. All tables have the same structure and are partitioned by the *l_orderkey* column, which is the first column. Each table has an index and a constraint on the *l_shipdate* column. The index is sorted in ascending order; the constraint ensures that only rows belonging to a particular month are actually stored in that table. From the monthly tables, yearly views are created; each yearly view is the union of the 12 tables of that year. From the yearly views, an overall view is created which is the union of all the yearly views.

The migration destination is a TSM tape storage pool. All DB2 files are migrated directly to this storage device.

The following subsections list the scripts that create the sample environment of this scenario. These scripts are written in Korn shell programming language.

To create and test this scenario, you need system administrator privilege to set up the HSM client, and database privilege to create the tables and views. The scenario assumes that the database instance is *db2inst*.

Before developing the scripts, create a data file named `months.dat` in the directory where you will place your scripts. This file contains the month information in the following format:

```
# Syntax: month<tab>firstday<tab>lastday
#         month is the name of a month in a year
#         firstday and lastday are in the DB2 DATE format
1_92   1992-1-1   1992-1-31
2_92   1992-2-1   1992-2-29
3_92   1992-3-1   1992-3-31
4_92   1992-4-1   1992-4-30

⋮
10_94  1994-10-1  1994-10-31
11_94  1994-11-1  1994-11-30
12_94  1994-12-1  1994-12-31
```

This data file facilitates the table creation scripts. You can use a format best suited for your needs and develop your scripts differently.

Creating Tables

This script reads the month data in the `months.dat` file, creates a table space for each monthly table first, then creates the table itself.

```
#!/bin/ksh
db2 connect to testdb
grep "^[^#]" months.data | cut -f1 | while read month
```

```
do
  tsname=li${month}_ts
  tabname=li${month}
  db2 "create tablespace ${tsname} managed by system \
       using ('/fs0/${tabname}') on node (0) \
       using ('/fs1/${tabname}') on node (1)"
  db2 "create table ${tabname} ( l_orderkey     integer not null,        \
                                 l_partkey      integer not null,        \
                                 l_suppkey      integer not null,        \
                                 l_linenumber   integer not null,        \
                                 l_quantity     decimal(15,2) not null,  \
                                 l_extendedprice decimal(15,2) not null, \
                                 l_discount     decimal(15,2) not null,  \
                                 l_tax          decimal(15,2) not null,  \
                                 l_returnflag   char(1) not null,        \
                                 l_linestatus   char(1) not null,        \
                                 l_shipdate     date not null,           \
                                 l_commitdate   date not null,           \
                                 l_receiptdate  date not null,           \
                                 l_shipinstruct char(25) not null,       \
                                 l_shipmode     char(10) not null,       \
                                 l_comment      varchar(44) not null,    \
        in ${tsname} not logged initially"
done
db2 commit work
db2 connect reset
```

Creating Indexes and Constraints

Since the data is divided by month, ensure that only the data belonging to a particular month goes into the table. This is done by adding a table constraint on the *l_shipdate* column. For example, the rows in the January 1992 table must satisfy the condition that *l_shipdate* is between 1992-1-1 and 1992-1-31. Also, create an index on the *l_shipdate* column sorted in ascending order.

```
#!/bin/ksh
db2 connect to testdb
grep "^[^#]" months.data | while read month firstday lastday
do
  db2 "create index ix${month} on li${month} (l_shipdate asc)"
  db2 "alter table li${month} add constraint c1 \
       check ( l_shipdate between '${firstday}' and '${lastday}')"
done
db2 commit work
db2 connect reset
```

Creating Views

Again, since the data is grouped by month, when it comes to selecting data from several months, there may be a need to specify a long list of table names in a query. This can be simplified by creating a view that is the union of all the monthly tables. This scenario uses a set of yearly views over the monthly tables, and then an overall view that is the union of all the yearly views.

```
#!/bin/ksh
function buildselect
{
  selectclause=""
  (( $# == 0 )) && return
  selectclause="select * from $1"
  shift 1
  while (( $# > 0 ))
  do
    selectclause="${selectclause} union all select * from $1"
    shift 1
  done
  return
}
db2 connect to testdb
for yr in $(grep "^[^#]" months.data | cut -f1 | cut -d_ -f2 | uniq)
do
  # Create a yearly view li<xx>.
  buildselect $(grep "_${yr}" months | awk '{print "li"$1}')
  db2 "create view li${yr} as ${selectclause}"
  yearlyviews="${yearlyviews} li${yr}"
done
# Create an overview which is the union of all yearly views.
buildselect ${yearlyviews}
db2 "create view lineitem as ${selectclause}"
db2 commit work
db2 connect reset
```

Populating the Tables

When you are finished creating the tables and views, create and use a suitable program or script to generate the data to be inserted into the monthly tables. In this scenario, approximately 282 000 rows were generated with the ship dates in 1992, 1993, and 1994. Use the DB2 AutoLoader to split and load the data into the tables. After completing the load of the data, issue the **runstats** command in DB2 to collect statistics. As an alternative, you could use the following script:

```
#!/bin/ksh
db2 connect to testdb
grep "^[^#]" months.data | cut -f1 | while read month
do
  db2 "runstats on table db2inst.li${month} and indexes all"
done
db2 commit work
db2 connect reset
```

Configuring TSM Server and the HSM Client

The server and the client communicate through the loopback interface. The detailed steps are not presented here, only those that are relevant to the scenario. For additional details, please consult the appropriate Tivoli reference manuals.

To migrate data directly to an 8mm tape drive, a storage pool and a management class must be defined, and the test machine must be registered to use the management class. A TSM administrator can run the following TSM macro to accomplish these tasks:

```
define libraray manlib libtype=manual
define drive manlib 8mmdrv device=/dev/mt0
define devclass 8mmdev devtype=8mm library=manlib format=8900 estcap=20g
define stgpool 8mmpool 8mmdev maxscratch=0
define domain dwdomain
define policyset dwdomain db2policy
define mgmtclass dwdomain db2policy hsmmgmt spacemgtechnique=automatic \
       migrequiresbkup=no migdestination=8mmpool
assign defmgmtclass dwdomain db2policy hsmmgmt
validate policyset dwdomain db2policy
activate policyset dwdomain db2policy
register node <host_name> <pass_word> domain=dwdomain
/* <host_name> is the name returned by the 'hostname' command */
/* <pass_word> is the password the client machine will use    */
```

Note: You must label and define a tape volume to the storage pool before you can store data on it.

Next, you must configure the HSM client. To complete the following tasks, you must be the root user on the test machine.

The first task is to set up the three Tivoli client options files. Place the HSM-specific parameters at the beginning of the system options file (dsm.sys) before any of the server definitions. Parameters not explicitly specified have default values. For our example, the contents of this file are:

```
DEFAULTSERVER                 server_a
MIGRATESERVER                 server_a
SERVERNAME   SERVER_A
   COMMETHOD                  tcpip
   TCPPORT                    1500
   TCPSERVERADDRESS           localhost
   PASSWORDACCESS             generate
   INCLEXCL                   /usr/lpp/adsm/bin/dsm.inclexcl
```

In our example, the include-exclude file (dsm.inclexcl) has contents as shown below. The contents show that we declare only data files and index files as eligible for migration.

```
exclude         /fs[01]/.../*
include         /fs[01]/li*9[234]/SQL*.DAT
include         /fs[01]/li*9[234]/SQL*.INX
```

A sample of the default user options file (dsm.opt) looks like this:

```
SERVERNAME       server_a
TAPEPROMPT       no
SUBDIR           yes
OPTIONFORMAT     standard
```

The second task is to enable space management on the two filesystems in which the table containers are created. You should ensure that the JFS filesystems are mounted before you proceed with this task. Then enter this command:

```
dsmmigfs add -agefactor=1 -sizefactor=0 -stubsize=256K /fs0 /fs1
```

The stub file size of 256 kilobytes was chosen in order to keep the first extent of any DB2 file local. The first 511 bytes in a stub file is used by HSM. The first extent of a DB2 file in the test database takes up 128 KB (made up of 32 pages each 4 KB in size). Since the stub size can only take values that are powers of 2, the next largest size is 256 KB. Files smaller than this size will not be migrated.

Migrating Tables

Three years of data were created: 1992, 1993, and 1994. The next command migrates the data for 1992 and 1993 to TSM storage. (The data for 1994 resides locally.)

```
dsmmigrate -recursive /fs[01]/li*9[23]
```

This command may take several minutes to complete. When completed, check the migration results by entering the next command to list the results for 1992:

```
dsmls /fs[01]/li*92
```

The output from this command appears similar to:

```
ADSTAR Distributed Storage Manager
space management Interface - Version 3, Release 1, Level 0.7
(C) Copyright IBM Corporation, 1990, 1999. All Rights Reserved.
      Actual     Resident     Resident  File   File
        Size         Size     Blk (KB)  State  Name
⋮
```

```
/fs0/li2_92:
      204800       204800          200  r      SQL00002.DAT
       20480        20480           20  r      SQL00002.INX
         512          512            4  r      SQLTAG.NAM
/fs0/li3_92:
      372736       262143          256  m      SQL00002.DAT
       28672        28672           28  r      SQL00002.INX
         512          512            4  r      SQLTAG.NAM
⋮
```

The File State column indicates the current migration status of a file. An “r” indicates a file is resident; an “m” indicates a file is migrated; a “p” indicates a file is pre-migrated. A pre-migrated file is one where a copy already exists on the server and it is the same as the local copy.

Sample Queries

This section presents some SQL statements that query migrated tables. In general, any SQL can be written without the knowledge that the tables have been migrated. The DB2 optimizer decides what files are needed to answer a query and only those files are recalled.

To view a list of files recalled by a query, you do not need to enable the migrate-on-close attribute in HSM. If you have already enabled this attribute, the recalled files are changed back to the migrated state when you disconnect from the DB2 session. With the migrate-on-close flag off, if a file is recalled it is indicated by a “p” in the File State column in the `dsmls` output.

Query 1

This query demonstrates querying a migrated monthly table:

```
select l_receiptdate, sum(l_quantity) from li1_93
   where (day(l_receiptdate) between 5 and 20)
   group by l_receiptdate order by l_receiptdate
```

When the query returns, check the migration status of the data files, and you will find the files for January 1993 are recalled. The output for the other files is not shown.

```
      Actual     Resident     Resident  File   File
        Size         Size     Blk (KB)  State  Name
/fs0/li1_93:
      557056       557056          544  p      SQL00002.DAT
       36864        36864           36  r      SQL00002.INX
         512          512            4  r      SQLTAG.NAM
/fs1/li1_93:
      573440       573440          560  p      SQL00002.DAT
       36864        36864           36  r      SQL00002.INX
         512          512            4  r      SQLTAG.NAM
```

Query 2

This query demonstrates querying a view for data that belongs to the first quarter of 1993:

```
select l_receiptdate, sum(l_quantity) from lineitem
   where (l_shipdate between '1993-1-1' and '1993-3-31') and
         (day(l_receiptdate) between 5 and 20)
   group by l_receiptdate order by l_receiptdate
```

This query is better written as shown above rather than the way shown below:

```
select l_receiptdate, sum(l_quantity) from (
   select l_receiptdate, l_quantity from li1_93 union all
   select l_receiptdate, l_quantity from li2_93 union all
   select l_receiptdate, l_quantity from li3_93 ) as firstquarter
   where day(l_receiptdate) between 5 and 20
   group by l_receiptdate order by l_receiptdate
```

Using either query, the data files for the tables li1_93, li2_93, and li3_93 are recalled.

Query 3

This query does not cause any data files to be recalled:

```
select count(*) from li1_93 where l_shipdate='1994-1-15'
```

No rows in li1_93 satisfy the predicate because the constraint on the `l_shipdate` column is between 1993-1-1 and 1993-1-31.

More Queries

Here are several other queries, some of which may be of interest to you:

```
select l_receiptdate, sum(l_quantity) from li1_93
   where (day(l_receiptdate) between 5 and 20) and
   (nodenumber(l_receiptdate)=0)
   group by l_receiptdate order by l_receiptdate

select l_shipdate, sum(l_quantity) from li1_93
   where day(l_shipdate) between 5 and 20)
   group by l_shipdate order by l_shipdate

select l_shipdate, sum(l_quantity) from li1_93
   where (day(l_shipdate) between 5 and 20) and
   (nodenumber(l_shipdate)=0)
   group by l_shipdate order by l_shipdate

select count(*) from li1_93
   where day(l_shipdate) between 5 and 20

select count(*) from li1_93
   where day(l_receiptdate) between 5 and 20

select l_receiptdate, sum(l_quantity) from li1_93
   where (day(l_receiptdate) between 5 and 20) and
   (nodenumber(l_receiptdate)=1)
   group by l_receiptdate order by l_receiptdate

select l_receiptdate from li1_93
   where (day(l_receiptdate) between 5 and 20) and
   (nodenumber(l_shipdate)=1)
   group by l_receiptdate order by l_receiptdate
```

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Limited

Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

The following terms, which may be denoted by an asterisk(*), are trademarks of International Business Machines Corporation in the United States, other countries, or both.

ACF/VTAM
AISPO
AIX
AIX/6000
AIXwindows
AnyNet
APPN
AS/400
BookManager
CICS
C Set++
C/370
DATABASE 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eNetwork
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
LAN DistanceMVS
MVS/ESA
MVS/XA
Net.Data
OS/2
OS/390
OS/400
PowerPC
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/DS
SQL/400
System/370
System/390
SystemView
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WIN-OS/2

The following terms are trademarks or registered trademarks of other companies:

Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

Java or all Java-based trademarks and logos, and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries or both and is licensed exclusively through X/Open Company Limited.

Other company, product, or service names, which may be denoted by a double asterisk(**) may be trademarks or service marks of others.