

Implementing Virtual Provisioning on EMC Symmetrix DMX with Sybase ASE

Applied Technology

Abstract

This white paper details the benefits of deploying Sybase ASE databases on EMC[®] Symmetrix[®] devices using Virtual Provisioning[™] technology.

April 2009

Copyright © 2008, 2009 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

All other trademarks used herein are the property of their respective owners.

Part Number h4288.3

Table of Contents

Executive summary	4
Introduction.....	4
Audience	4
Terminology	4
Overview.....	5
Requirements.....	6
Host allocation considerations.....	7
Considerations for ASE components on thin devices	7
Thin device planning and mapping	7
Creating the Sybase environment using thin devices.....	7
Creating, adding, and binding devices	8
Setting up the Sybase environment	9
Virtual Provisioning and automatic database expansion	11
Implementing sp_dbextend with thin devices.....	12
ASE temporary database (tempdb)	13
Approaches for replication	14
Thin pool management.....	14
Thin pool monitoring	14
Exhaustion of oversubscribed pools	15
Conclusion	15
References	15

Executive summary

EMC® Symmetrix® DMX-3 and DMX-4 storage arrays continue to provide increased storage utilization and optimization, enhanced capabilities, and greater interoperability and security, as well as multiple ease-of-use improvements.

Virtual Provisioning™, generally known in the industry as “thin provisioning,” enables organizations to improve ease of use, enhance performance, and increase capacity utilization for certain applications and workloads. The implementation of Virtual Provisioning for DMX-3 and DMX-4 storage arrays directly addresses improvements in storage infrastructure utilization, as well as associated operational requirements and efficiencies.

This white paper addresses the considerations for deploying Sybase ASE on thinly provisioned devices. An understanding of the principles discussed in this paper will allow users to effectively implement Sybase ASE databases with Virtual Provisioning.

Introduction

One of the biggest challenges facing storage administrators is storage provisioning for new applications. Administrators typically allocate space based on anticipated future growth of applications. This is done to mitigate recurring operational functions, such as incrementally increasing storage allocations or adding discrete blocks of storage as existing space is consumed. Using this approach results in more physical storage being allocated to the application than is needed for a significant amount of time and at a higher cost than is necessary. This type of provisioning of physical storage also leads to increased power, cooling, and floor space requirements. Even with the most careful planning, it may be necessary to provision additional storage in the future, which could potentially require an application outage.

A second layer of provisioning efficiency happens when a database administrator over allocates storage for a segment. The operating system sees the space as completely allocated but internally only a fraction of the allocated space might be used.

EMC Virtual Provisioning can address both issues. Virtual Provisioning allows more storage to be presented to an application than is physically available. More importantly, it will allocate physical storage only when the storage is actually written to. This allows more flexibility in planning for growth and reduces the initial costs of provisioning storage to an application. Lastly, Virtual Provisioning can help prevent the inherent waste in overallocation of space and administrative management of storage allocations.

Audience

This white paper is intended for storage architects and administrators, and database administrators responsible for deploying Sybase ASE databases on Symmetrix DMX-3 and DMX-4 using 5773 Engenuity™.

Terminology

Table 1. Basic Symmetrix array terms

Term	Description
Device	A logical unit of storage defined within a Symmetrix array.
Device Capacity	The actual storage capacity of a device
Device Extent	The size of the smallest contiguous region of a device for which an extent mapping can occur
Host Accessible Device	A device that is exported for host use
Internal Device	A device used for internal function of the array

Metavolume	An aggregation of host accessible devices, seen from the host as a single device
Storage Pool	A collection of internal devices for some specific purpose

Table 2. Virtual Provisioning terms

Term	Description
Thin Device	A host accessible device that has no storage directly associated with it
Data Device	An internal device that provides storage capacity to be used by thin devices
Extent Mapping	Specifies the relationship between the thin device and data device extents. The extent sizes between a thin device and a data device do not need to be the same.
Thin Pool Capacity	The sum of the capacities of the member data devices
Bind	The process by which one or more thin devices are associated to a thin pool
Unbind	The process by which a thin device is diassociated from a given thin pool. When unbound, all previous extent allocations from the data devices are erased and returned for reuse
Enabled Data Device	A data device belonging to a thin pool on which extents can be allocated for thin devices bound to that thin pool
Disabled Data Device	A data device belonging to a thin pool from which capacity cannot be allocated for thin devices. This state is under user control. If a data device has existing extent allocations when a disable operation is executed against it, the extents will be relocated to other enabled data devices with available free space within the thin pool
Thin Pool Enabled Capacity	The sum of the capacities of enabled data devices belonging to a thin pool
Thin Pool Allocated Capacity	A subset of thin pool enabled capacity that has been allocated for the exclusive use of all thin devices bound to that thin pool
Thin Pool Pre-allocated Capacity	The initial amount of capacity that is allocated when a thin device is bound to a thin pool. This property is under user control
Thin Device Written Capacity	The capacity on a thin device that was written to by a host. In most implementations this is a subset of the thin device allocated capacity
Thin Device Subscribed Capacity	The total capacity that a thin device is entitled to withdraw from a thin pool. This may be equal to or less than the thin device capacity. In the current implementation they are equal
Thin Device Allocation Limit	The capacity limit that a thin device is entitled to withdraw from a thin pool, which may be equal to or less than the thin device subscribed capacity

Overview

Symmetrix thin devices are logical devices that can be used in many of the same ways that Symmetrix devices have traditionally been used. However, unlike traditional Symmetrix devices, thin devices do not need to have physical storage completely allocated at the time the device is created and presented to a host. The physical storage that is used to supply disk space to thin devices comes from a shared storage pool called a thin pool. The thin pool is comprised of devices called data devices that provide the actual physical storage to support the thin device allocations.

When a write is performed to a part of the thin device for which physical storage has not yet been allocated, the Symmetrix allocates physical storage from the thin pool for that portion of the thin device only. The Symmetrix operating environment, Enginuity, satisfies the requirement by providing a block of storage from the thin pool called a thin device extent. This approach reduces the amount of storage that is actually consumed.

The minimum amount of physical storage that can be reserved at a time for the dedicated use of a thin device is referred to as the data device extent. The entire data device extent is physically allocated to the thin device at the time the thin storage allocation operation is performed. The data device extent is allocated from any one of the data devices in the associated thin pool.

The minimum amount of physical storage that must be mapped to a thin device at a time is called a thin device extent. For Symmetrix, the thin device extent size is the same as the data device extent size.

When a read is performed on a thin device, the data being read is retrieved from the appropriate data device in the thin pool to which the thin device is associated. If for some reason a read is performed against an unallocated portion of the thin device, zeroes are returned to the reading process.

When more physical data storage is required to service existing or future thin devices, for example, when a thin pool is approaching full storage allocations, data devices can be added to existing thin pools dynamically without needing a system outage. New thin devices can also be created and associated with existing thin pools.

When data devices are added to a thin pool they can be in an enabled or disabled state. In order for the data device to be used for thin extent allocation it needs to be in the enabled state. For it to be removed from the thin pool, it needs to be in a disabled state. A data device can be disabled only if it does not have any thin extents allocated. The following figure depicts the relationships between thin devices and their associated thin pools. There are nine devices associated with thin Pool A and three thin devices associated with thin pool B.

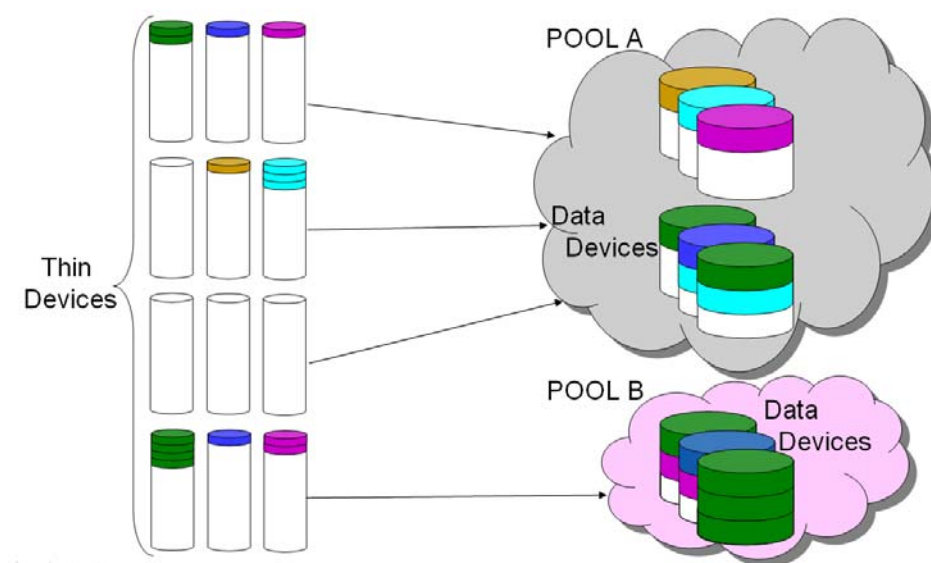


Figure 1. Thin devices and thin pools containing data devices

The way data extents are allocated across the data devices results in a form of striping in the thin pool. The more data devices that are in the thin pool, the wider the striping is and the greater the number of devices that can participate in application I/O. The thin extent size is currently 768 KB in size. This may change in future versions of the Enginuity operating environment.

The maximum size of a thin device is 64 GB. If a larger size is needed then a metavolume comprised of thin devices can be created. It is recommended that the metavolume be concatenated rather than striped since the thin pool is already striped using thin extents.

The maximum size of a single ASE device is 32 GB, so a single thin device, not comprised of any metavolumes, is big enough to accommodate an entire ASE device.

Requirements

Virtual Provisioning requires an Enginuity code level of 5773.XX or later. In order to create and manage thin pool devices, EMC Solutions Enabler 6.5.xx is required. If the Symmetrix Management Console (SMC) is being used to manage Virtual Provisioning components, version 6.1 is needed. Thin pools can

only be created by the customer and cannot be created during the bin file (DMX configuration) creation process.

To document the implementation guidelines for this paper, tests were performed on SUN Solaris 5.10 using Sybase ASE 15.0.2.

The screenshots and command syntax displayed in this paper were created using a pre-GA release of Solutions Enabler and Symmetrix Management Console. The options displayed in the screen captures may be different from the ones available in the GA release of the code.

Host allocation considerations

Host file system and volume management routines behave in subtly different ways. These behaviors are usually invisible to end users. The introduction of thin devices can put these activities in a new perspective. For example, a UNIX file system might show as empty when a “`df -k`” command is executed, but many blocks may have been written to the thin device in support of file system metadata like inodes and so on. From a Virtual Provisioning point of view this thin device could be using a substantial number of thin extents even though the space is showing as available to the operating system and logical volume manager. While this kind of preformatting activity diminishes the value of Virtual Provisioning in one area, it does not negate the other values that Virtual Provisioning brings.

In addition, ASE database segments, partitions, and other components each can have different kinds of behavior when laid down on virtually provisioned devices. When Sybase initializes new files, such as log, data and temp files, it writes metadata to each initialized block based on the page size selected for the Sybase server (2, 4, 8, or 16K). This will cause the thin pool to allocate the amount of space that is being initialized by the database. As database files are added, more space will be allocated in the pool. In order to get the most benefit from a Virtual Provisioning infrastructure, a strategy for sizing files, pools, and devices should be developed in accordance with application and storage management needs.

Considerations for ASE components on thin devices

There are certain considerations that a DBA should be aware of when deploying Sybase ASE using Virtual Provisioning. How the various components work in this kind of configuration depends on the component itself, its usage, and sometimes the underlying host data structure supporting the component. This section describes how the various ASE components interact with virtually provisioned devices.

Thin device planning and mapping

When planning the thin device sizes and quantities, the best approach is to map as many thin devices to the server that will possibly be needed for the lifetime of the database application. Also, their size should be as large as the long-term capacity growth will require. These guidelines will help avoid future application impact due to the need for additional LUN provisioning. In other words, it is common for the thin pool to be sized for *near-term* database capacity growth, and for the thin LUNs to be sized for *long-term* LUN capacity growth. Since the thin LUNs do not take space in the pool until data is written to them, this method optimizes storage utilization and reduces the database and application impact. Periodically, as database space requirements are reassessed and changes become necessary, thin pool capacity and utilization changes can easily be accommodated.

Creating the Sybase environment using thin devices

Disk devices created from Symmetrix thin devices need to be managed in the same manner as traditional devices. All recommendations apply, including correctly defining data and log segments, and other database object placement guidelines. Symmetrix DMX™ thin devices are configured using either the Symmetrix Command Line Interface (SYMCLI) or Symmetrix Management Console (SMC). Thin pools are created, and then thin devices are added and bound to a pool.

When devices are presented to a host for Sybase initialization, they look identical to other host operating system devices that users are already familiar with. The procedures and/or commands to initialize and create ASE databases and objects remain the same for thin devices, as they would for any other host or storage-based device.

Symmetrix ID: 000190300703

Device Name	Directors	Device	Cap	Attribute	Sts	(MB)
Physical	Sym SA :P DA :IT	Config				
/dev/rdisk/emcpower0c	002B 15C:0 15A:C6	2-Way Mir	N/Grp'd	RW	3	
/dev/rdisk/emcpower1c	0217 15C:0 NA:NA	TDEV	N/Grp'd	RW	30720	
/dev/rdisk/emcpower3c	0219 15C:0 NA:NA	TDEV	N/Grp'd	RW	30720	
/dev/rdisk/emcpower9c	0038 15C:0 01A:D9	2-Way Mir	N/Grp'd	RW	18433	
/dev/rdisk/emcpower21c	0043 15C:0 01B:DC	2-Way Mir	Grp'd	RW	18433	
/dev/rdisk/emcpower22c	020F 15C:0 NA:NA	TDEV	N/Grp'd	RW	30720	
/dev/rdisk/emcpower23c	0044 15C:0 16A:DC	2-Way Mir	Grp'd	RW	18433	
/dev/rdisk/emcpower26c	0047 15C:0 16B:DD	2-Way Mir	Grp'd	RW	18433	
/dev/rdisk/emcpower27c	0048 15C:0 01A:D5	2-Way Mir	Grp'd	RW	18433	
/dev/rdisk/emcpower28c	0210 15C:0 NA:NA	TDEV	N/Grp'd	RW	30720	
/dev/rdisk/emcpower31c	0213 15C:0 NA:NA	TDEV	N/Grp'd	RW	30720	
/dev/rdisk/emcpower32c	0214 15C:0 NA:NA	TDEV	N/Grp'd	RW	30720	
/dev/rdisk/emcpower33c	0023 01C:0 15A:C8	2-Way Mir	N/Grp'd	RW	3	
/dev/rdisk/emcpower34c	0024 01C:0 15B:D4	2-Way Mir	N/Grp'd	RW	3	

ICO-IMG-000626

Figure 2. Symmetrix devices available to the host

Figure 2 displays a Symmetrix physical device listing showing all available LUNs for the host. The devices labeled “TDEV” are the thin devices, “2-Way Mir” are non-thin (also called “fat”) devices, and the devices that are 3 MB in size are EMC gatekeepers.

Creating, adding, and binding devices

Creating thin pools, adding devices to pools, and binding those devices are required steps for creating and configuring a virtually provisioned environment with EMC Symmetrix.

Data devices are created with the Symmetrix Command Line Interface `symconfigure` command syntax:

1. **`symconfigure -sid 232 -cmd "create pool sybase type=thin;" commit`**
 - a. A thin pool named sybase is created.

Note: Thin pools can be created without adding data devices so that they can be populated at a later time.

2. **`symconfigure -sid 232 -cmd "create dev count=16, size=4602, emulation=fba, in pool=sybase member_state=enable config=2-Way-Mir, attribute=datadev;" commit`**
 - a. This command creates 16 data devices from standard RAID 1 (2-Way Mir) devices. The devices must be of emulation type FBA for open systems hosts. The sizing designation is at the user's discretion. For this example, the devices were created of size 4602 MB (or roughly 4 GB each).

Note: The output listing of the above command includes the new device IDs which will be needed to create the thin devices.

3. **symconfigure -sid 232 -cmd "create dev count=16, size=4602, emulation=fba, config=tdev binding to pool=sybase;" commit**

- a. Create 16 thin devices and bind them to the sybase pool.

Note: the output of the command displays the thin device IDs 10F4:10F7.

4. **symconfigure -sid 232 -cmd "map dev 10F4:10F7 to dir 10A:1 starting lun=32;" commit**

symconfigure -sid 232 -cmd "map dev 10F4:10F7 to dir 7A:0 starting lun=32;" commit

- a. Map the thin devices to the two front-end adapter (FA) ports.

5. **symmask -sid 232 -wwn 10000000c9563968 add devs 10F4:10F7 -dir 10A -p 1**
symmask -sid 232 -wwn 10000000c9563969 add devs 10F4:10F7 -dir 7A -p 0

- a. Mask the devices to the host.

Note: It will be necessary to rescan the host buses or reboot, in order to make the devices visible to the host.

6. **symcfg discover**

- a. After rescan or reboot, issue this command to rediscover available devices.

Setting up the Sybase environment

Using the UNIX link command, create files that identify the devices, in this case raw partitions, needed by Sybase for initialization.

```
licoc035# ln -s /dev/rdisk/emcpower1c master-VP
licoc035# ln -s /dev/rdisk/emcpower3c sybprocs-VP
licoc035# ln -s /dev/rdisk/emcpower22c systemdb-VP
licoc035# ln -s /dev/rdisk/emcpower28c VP-datadev1
licoc035# ln -s /dev/rdisk/emcpower31c VP-datadev2
licoc035# ln -s /dev/rdisk/emcpower32c VP-logdev1
```

For an ASE instance, use transact-SQL commands or procedures to initialize the disk devices necessary to create Sybase databases and objects. Standard "disk init" syntax was used to create the Sybase devices using the UNIX links that point to thin devices.

```
1> sp_helpdevice
2> go
device name physical name
description
status cntrltype vdevno   vpn low  vpn high
-----
VP_datadev1 /usr/sybaseASE/tpcc/devs/VP-datadev1
special, dsync off, directio off, physical disk, 1000.00 MB
2      0      3      1  7864320
VP_datadev2 /usr/sybaseASE/tpcc/devs/VP-datadev2
special, dsync off, directio off, physical disk, 1000.00 MB
2      0      4      1  7864320
VP_logdev1  /usr/sybaseASE/tpcc/devs/VP-logdev1
special, dsync off, directio off, physical disk, 500.00 MB
2      0      7      1  3670016
```

ICO-IMG-000627

Figure 3. Disk devices initialized for an ASE instance

Notice that “dsync” and “directio” are set to “off” when the devices are created. These are the default settings and not applicable when using raw partitions. Only raw partitions were used during this testing.

When dsync is off, writes to the device file may be buffered by the UNIX and Windows NT file systems, and the recovery of data on the device cannot be ensured. The dsync setting should be turned off only when data integrity is not required, or when the system administrator requires performance and behavior similar to earlier Adaptive Server versions.

As for directio, this parameter allows users to configure Adaptive Server to transfer data directly to disk, bypassing the operating system buffer cache. directio performs I/O in the same manner as raw devices and provides the same performance benefit as raw devices, but has the ease of use and manageability of file system devices. directio is a static parameter that requires a restart of Adaptive Server to take effect.

Next, create the database on the previously defined thin devices using the following Sybase command:

```
Isql> CREATE DATABASE TEST ON
Isql>VP_datadev1="1000M", VP_datadev2="1000M"
Isql>LOG ON VP_logdev1="500M"
```

```
Create database output that looks like this:
CREATE DATABASE: allocating 256000 logical pages (1000.0 megabytes) on disk 'VP_datadev1'.
CREATE DATABASE: allocating 256000 logical pages (1000.0 megabytes) on disk 'VP_datadev2'.
CREATE DATABASE: allocating 128000 logical pages (500.0 megabytes) on disk 'VP_logdev1'.
Msg 3225, Level 21, State 1:
Server 'VP', Line 1:
```

It's good practice to review the status of the thin device pools after database creation. As depicted in the next figure, the thin pools that have been created on symmetrix id 703 are displayed. The intent of this example is to show that space is consumed within a thin device pool, on an as-needed basis.

```
Licoc035# symcfg -sid 703 list -pool -tp data
```

Symmetrix ID: 000190300703

SYMMETRIX POOLS

Pool Name	Type	Device	Device Emulation	Configuration	Total Tracks	Enabled Tracks	Used Tracks	Free Tracks	Full (%)	State
sybase	TD	FBA	2-Way	Mir	2293788	1966104	1368180	597924	69	Ena
kjb	TD	FBA	2-Way	Mir	327684	327684	163860	163824	50	Ena
test	TD	FBA	2-Way	Mir	65544	65544	65544	0	100	Ena
Total Tracks					2687016	2359332	1597584	761748	67	

ICO-IMG-000628

Figure 4. Thin pool capacity and utilization

Query the Symmetrix to review thin pool capacity and utilization. Three thin storage pools were defined for the testing. They were created as small (1 GB), medium (200 GB), and large (900 GB). When a Sybase ASE database is created, every page is written with zeros and as a result, the necessary storage is preallocated at the time of database creation. The ASE databases were created using a 4K page size. The pools in Figure 4 are utilized at 69%, 50%, and 100% capacity. The next section will discuss automatic database extension features that can be used to increase the size of a device or database segment as it reaches a predefined limit (or threshold).

Virtual Provisioning and automatic database expansion

Sybase ASE provides automatic expansion of databases and devices. This feature allows users to configure databases to expand automatically when they run out of space. The automatic database expansion stored procedure "sp_dbextend" allows users to install thresholds that identify those devices with available space, and then appropriately alter the database(s) onto these devices.

First, the database must be set up for automatic expansion (refer to the *Sybase ASE System Administration Guide Volume 2* for details). When a database grows to its free space threshold, internal mechanisms fire, increasing the size of the database by the amount of space each expansion policy specifies. The automatic expansion process measures the amount of room left on all devices bound to the database. If there is sufficient room on the devices, the database continues to grow. If any devices are configured for expansion, those devices will be expanded as threshold procedures dictate.

Symmetrix DMX Virtual Provisioning technology provides a viable integration opportunity with ASE's automatic database expansion feature. Thin devices presented to Sybase should be configured to address the storage requirement for the lifetime of the infrastructure. This approach makes sense because thin devices do not require equivalent physical storage initially. Physical capacity can be added to the thin pool as the environment grows.

As an example, if users create small devices and let them grow over time, they make better use of Virtual Provisioning and increase storage utilization. Conversely, if large database files are created, and they remain mostly empty, all of the storage is allocated up front. Therefore, little to no savings in storage utilization is achieved.

Implementing sp_dbextend with thin devices

Database administrators can configure the size or percentage by which each database and segment pair should be expanded, and the size or percentage by which each device should be expanded.

By executing the “sp_dbextend” stored procedure, database device thresholds are defined by growth allocation (growby) and maximum size (maxsize) values as shown in Figure 5.

```
1> exec sp_dbextend 'set', 'device', VP_datadev1, "10M", "1100M"

1> exec sp_dbextend 'set', 'device', VP_datadev2, "10M", "1100M"

1> exec sp_dbextend 'set', 'device', VP_logdev1, "10M", "600M"

1> sp_dbextend list
2> go
type      name      segment  item  value status
-----
feature server-wide (n/a)  (n/a) (n/a) enabled
database VP      logsegment growby 3M  enabled
database VP      system    growby 1M  enabled
database default (all)  growby 10% enabled
device  VP_datadev1 (n/a)  growby 10M enabled
device  VP_datadev1 (n/a)  maxsize 1100M enabled
device  VP_datadev2 (n/a)  growby 10M enabled
device  VP_datadev2 (n/a)  maxsize 1100M enabled
device  VP_logdev1 (n/a)   growby 10M enabled
device  VP_logdev1 (n/a)   maxsize 600M enabled
device  default (n/a)    growby 10% enabled
```

ICO-IMG-000629

Figure 5. Define database device thresholds for automatic expansion

In Figure 5 the database data and log devices reside on DMX thin devices. The automatic database expansion threshold has been set to allocate device growth by 10 MB for both the data and log devices. The maxsize of the data devices is set to 1100 MB and the maxsize for the log device is set to 600 MB. As the database is automatically extended, the thin pool allocation size increases as shown in the following table.

Table 3. Thin pool consumption

Test description	Used tracks	Free full tracks	Approx. usage size	Pool util %
create thin pool & bind devices	84	1146831	5.5 MB	0
create Sybase server	46980	1099935	3.0 GB	4
Initialize Sybase devices	47028	1099887	3.0 GB	4
create Sybase database	111060	1035855	7.2 GB	9
create tables & indexes	111060	1035855	7.2 GB	9
load data	111060	1035855	7.2 GB	9
auto extend 10MB	99161	1023956	6.4 GB	10
auto extend 10MB	87262	1012057	5.7 GB	11
auto extend 10MB	75363	1000158	4.9 GB	12

Table 3 represents the amount of thin pool consumption based on the type of database action taken. Used tracks and percentage of thin pool utilization for various database actions are depicted to show how the thin device pool is utilized during everyday database administrative operations. When the ASE database is first created, zeros are written to every page allocated to the database (in this case, a 4K page size). Therefore, we see a significant increase in pool utilization. However, with various database operations, such as transaction processing and data loads, the pool size is not impacted because Sybase has preallocated the necessary space during the database creation process. Only when the database devices are created or altered do we see another increase in the thin device pool utilization. File management of the thin device pools can also be done manually by extending or adding devices.

Note that the “used tracks” and “free full tracks” appear to go down when the database is extended. The number of “used tracks” depicts tracks consumed per operation (for example, “auto extend”). The number of “free full tracks” continues to decrease as space is utilized in the pool. Each operation depicted in Table 3 represents an individual action taken on the database, which in turn affects the devices associated with the pool. Keep in mind that the Sybase instance and the database devices are configured on different thin devices.

The best practice recommendation would be to implement threshold procedures for automatic database expansion. This is an optimal approach to Virtual Provisioning, seamless to end users, and will reduce, if not eliminate, the ongoing storage management tasks associated with increasing storage capacity utilization.

ASE temporary database (tempdb)

Sybase ASE has a temporary database, named tempdb, that provides a storage area for temporary tables and other temporary working storage needs. The space in tempdb is shared among all users of all databases on the server. The default size of tempdb depends on the logical page size for the server: 2, 4, 8, or 16K. Certain activities make it necessary for users to increase the size of tempdb. The most common of these are:

- The need for large temporary tables.
- A lot of activity on temporary tables, which fills up the tempdb logs.
- Large sorts or many simultaneous sorts. Subqueries and aggregates with “group by” also cause activity in tempdb.

Users can increase the size of tempdb with alter database. The tempdb is initially created on the master device. Space can be added to tempdb from the master device or from any other database device, including a Symmetrix DMX thin device. The temporary database is dropped and re-created automatically when the Sybase server is rebooted. Therefore, the amount of space that tempdb consumes continually changes.

The usage cycle of virtually provisioned storage is important to follow and understand. Keep in mind that the storage needs for tempdb are essentially growing and shrinking as temporary tables are created and then dropped. When space is allocated using a virtual device for tempdb, the device continues to grow as it is written to. When the space is released by the database, the thin extents will still remain allocated. This is because there is not yet a command set from the host that can indicate back to the storage system that the device is no longer in use. Therefore, the space is not truly freed up, making use of a thin device for tempdb not optimal for increasing storage capacity utilization.

However, another usage scenario for tempdb involves creating multiple tempdbs. This can be done on-the-fly, depending on workload needs. A user may bind a login or application to a user-created tempdb, or by adding a tempdb to a named group of tempdbs. Then the user can provision tempdb space dynamically for an application, or for a specific workload. Once the tempdb space is no longer needed, the DBA can drop the dynamically created tempdb, and the ASE device(s) associated with it, thereby returning the space to the virtual (thin device) pool.

Requirements for satisfying tempdb needs will vary from application to application. Understanding applications' specific needs for tempdb will help utilize the capabilities of Virtual Provisioning where tempdb is concerned.

Approaches for replication

Organizations will be able to perform "thin to thin" replication with Symmetrix thin devices by using standard TimeFinder®, SRDF®, and Open Replicator operations. This includes TimeFinder/Snap, TimeFinder/Clone, SRDF/Synchronous, and SRDF/Asynchronous. Not all replication modes are available with the initial release. Refer to the specific product's Release Notes for more details.

In addition, thin devices can be used as control devices for hot and cold pull and cold push Open Replicator copy operations. If a push operation is done using a thin device as the source, zeroes will be sent for any regions of the thin device that have not been allocated, or that have been allocated but have not been written to.

Open Replicator can also be used to copy data from a standard device to a thin device. If a pull or push operation is initiated from a standard device that targets a thin device, then a portion of the target thin device, equal in size to the reported size of the source volume, will become allocated.

Thin pool management

When storage is provisioned from a thin pool to support multiple thin devices there is usually more "virtual" storage provisioned to hosts than actually exists in the thin pool. This approach is called oversubscribing and is one of the main reasons for using Virtual Provisioning. However, there is a possibility that applications using a thin pool may grow rapidly and allocate more storage in the thin pool than is actually there. This condition is known as exhaustion of the thin pool. This is an undesirable situation. The next section discusses the steps necessary to avoid exhaustion of an oversubscribed pool.

Thin pool monitoring

Along with Virtual Provisioning comes several methodologies to monitor the capacity consumption of the thin pools. The Solutions Enabler 6.5 `symcfg monitor` command can be used to monitor pool utilization, as well as display the current allocations through the `symcfg show pool` option. There are also event thresholds that can be monitored through the SYMAPI event daemon, thresholds that can be set with Symmetrix Management Console, and SNMP traps that can be sent for monitoring by EMC ControlCenter® or any data center management product.

System administrators and storage administrators must put processes in effect to monitor the capacity for thin pools to make sure that they do not get filled. The pools can be dynamically expanded to include more data devices without application impact. These devices should be added in groups long before the thin pool approaches a full condition. If devices are added individually, hot spots on the disks can be created when

much of the write activity is suddenly directed to the newly added data devices because other data devices in the pool are full.

Exhaustion of oversubscribed pools

If a thin pool is oversubscribed it may fill beyond its capacity. When Sybase attempts to write a page that requires the allocation of a new extent in the pool, and the pool can't allocate it because it exhausted its actual capacity, an I/O error will be returned to the user. The database responds to that error by waiting or hanging the user process, and provides the following message: "The user process will sleep until write completes successfully." More storage must then be provisioned to the affected thin pool before database operations can resume.

However, in some cases this condition may result in a file system hang and/or database crash. Since Sybase is an ACID-compliant database, customers should not be concerned that a crash, caused by the exhaustion of space in the thin pool, could cause data corruption. This will not happen with Sybase since there will be no loss of committed transactions. During database restart the database will undo and redo transactions to achieve a transactionally consistent point in time.

The virtual device pool named "test" in Figure 4 on page 11 has reached its full capacity and must be expanded before any additional writes to the database will be allowed. In the Sybase log file, a disk I/O error (823) is reported when attempting to write to a thin device that has been oversubscribed. Users can avoid this situation by assigning appropriate thresholds to both the thin pool, and the Sybase database, which can alert threshold procedures to alter the database(s) onto virtual devices, as storage limits approach.

Conclusion

Virtual Provisioning provides a simple, noninvasive, and economical way to provide storage for a Sybase ASE database. The following recommendations and guidelines should be followed for successful implementations using Symmetrix DMX Virtual Provisioning with Sybase ASE:

- All of the standard best practice recommendations for configuring Sybase on EMC Symmetrix apply. This includes correctly defining data and log segments, and following all database object placement guidelines.
- Use "sp_dbxextend" to implement database thresholds that will alter the database(s) onto virtual devices, as storage limits are approached.
- Be aware of disk space utilization if implementing tempdb with Virtual Provisioning. The storage allocated to the thin device is not released as tempdb objects are dropped or deleted. Therefore, if thin devices are used for tempdb, it's critical to ensure that device pools are monitored, and threshold procedures are implemented to automatically extend the temporary database as it becomes full.

The following are situations where Virtual Provisioning may not be optimal:

- Systems where predictable and deterministic performance is necessary.
- Systems where data is deleted and re-created rapidly and the underlying data management structure does not allow for reuse of the deleted space.

References

- *Best Practices for Fast, Simple Capacity Allocation with EMC Symmetrix Virtual Provisioning Technical Note*
- *Symmetrix Virtual Provisioning Feature Specification*