White Paper

z/OS AND VIRTUAL PROVISIONING™ BEST PRACTICES

Abstract

This white paper describes EMC® Virtual Provisioning for count key data (CKD) volumes in a z/OS Operating System environment and covers some best practices and considerations for implementing virtual provisioned storage.

June 2013

EMC²

Copyright © 2102, 2013 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate 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.

Part Number h10688.2

Table of Contents

Executive summary

VMAX® 40K is the newest member of the Symmetrix® VMAX Family arrays. It is built on the foundation of the industry-leading Virtual Matrix architecture, and provides higher levels of scalability, performance, and consolidation for the most demanding virtual data-center environments. The Symmetrix VMAX Family consists of VMAX® 40K, VMAX® 20K, and VMAX® 10K.

The latest release of the Enginuity™ operating environment for Symmetrix, Enginuity 5876, supports Symmetrix VMAX 10K, VMAX 20K, and VMAX 40K. The capabilities of Enginuity 5876 to network, share, and tier storage resources allows data centers to consolidate applications and deliver new levels of efficiency with increased utilization rates, improved mobility, reduced power and footprint requirements, and simplified storage management.

Enginuity 5876 includes significant enhancements for mainframe users of the Symmetrix VMAX array that rival in importance to the original introduction of the first Symmetrix Integrated Cached Disk Array in the early 1990s. After successful deployment in open systems (FBA) environments, mainframe VMAX users now have the opportunity to deploy Virtual Provisioning™ and Fully Automated Storage Tiering for Virtual Pools (FAST™ VP) for count key data (CKD) volumes.

EMC® Symmetrix Virtual Provisioning for CKD devices adds a new dimension to z/OS tiered storage by improving capacity utilization and simplifying storage management. Symmetrix Virtual Provisioning integrates with existing device management, replication, and management tools, enabling customers to easily build Virtual Provisioning into their existing z/OS storage management processes.

This white paper examines Virtual Provisioning for count key data (CKD) volumes in a z/OS Operating Systems environment and covers some best practices and considerations for deploying a virtually provisioned storage strategy.

Audience

This white paper is intended for z/OS system programmers, performance analysts, capacity planners, storage administrators, SAN administrators, IT engineers, and others who are responsible implementing virtually provisioned storage on Symmetrix VMAX arrays with Enginuity 5876.

Terminology

Virtual Provisioning for z/OS brings with it some new terms that may be unfamiliar to mainframe practitioners. The following table describes these new terms that are used extensively throughout this white paper.

Term	Description
Device	A logical unit of storage defined within a Symmetrix array.
Device capacity	The actual storage capacity of a device.
Track group (Thin 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 presented on a FICON channel for host use.
Internal device	A device used for internal function of the array.
Storage pool	A collection of internal devices for some specific purpose.
Thin device (TDEV)	A host-accessible device that has no storage directly associated with it.
Data device (TDAT)	An internal device that provides storage capacity to be used by a thin device.
Extent mapping	Specifies the relationship between a 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 (Virtual pool)	A collection of data devices that provide storage capacity for thin devices.
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 disassociated 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, a DRAIN operation needs to be executed against it, so that the extents are 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 device User pre-allocated capacity	The minimum amount of capacity that is pre-allocated to a thin device when it is bound to a thin pool. This property is not under user control.
Thin device Allocated capacity	The capacity that has been allocated from the thin pool enabled capacity for the exclusive use of a thin device.
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.

Table 1. Virtual Provisioning terms

Virtual Provisioning: Implementation overview

Symmetrix Virtual Provisioning introduces a new type of host-accessible device called a thin device that can be used in the same ways that regular, host-accessible Symmetrix devices have traditionally been used. However, unlike regular Symmetrix devices, thin devices do not need to have physical storage completely allocated at the time the devices are 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 (also known as a virtual pool). The thin pool is comprised of devices called data devices that provide the actual physical storage to support the thin device allocations. When they are first created, thin devices are not associated with any particular thin pool. An operation referred to as binding must be performed to associate a thin device with a thin pool before the thin device can have a VTOC or indexed VTOC (IXVTOC) placed upon it or is available for any data operations. Thin pools support specific disk geometry (CKD3390 or FBA), drive technology (EFD, FC, or SATA), drive speed, and RAID protection type.

When a write is performed to a portion of the thin device, the Symmetrix subsystem allocates a minimum allotment of physical storage from the thin pool and maps that storage to a region of the thin device, including the area targeted by the write. The storage allocation operations are performed in small units of storage called track groups (also known as device extents). A round-robin mechanism is used to disperse the track group across all of the data devices in the pool that are enabled and that have remaining unused capacity. The track group size is 12 tracks. This means that the initial bind of a thin device to a pool causes some number of track groups, depending on device size, to be allocated to the thin device.

When a read is performed on a thin device, the data being read is retrieved from the appropriate data device in the storage pool to which the thin device is bound. Reads directed to an area of a thin device that has not been mapped do not trigger allocation operations. The result of reading an unmapped track is that an empty track (no user records, only standard record zero) is returned. When more storage is required to service existing or future thin devices, data devices can be added to existing thin storage pools. New thin devices can also be created and associated with existing thin pools.

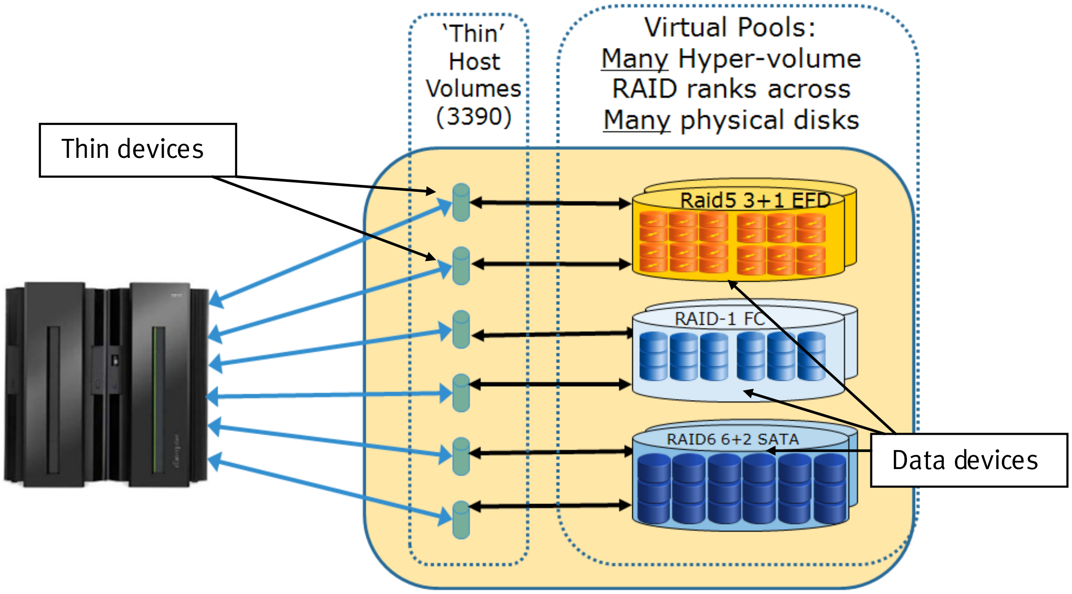

Figure 1. Virtual Provisioning in a Symmetrix VMAX array

If the thin device has the attribute of PREALLOCATE, then the Symmetrix subsystem ensures that the physical track space is pre-assigned to the device. If the thin device has the attribute of PERSIST (which implies and requires PREALLOCATE), then the Symmetrix subsystem ensures that the physical track space is not reclaimed.

As noted, it is possible for a thin device to be presented for host-use before all of the reported capacity of the device has been assigned. It is also possible for the sum of the reported capacities of the thin devices using a given pool to exceed the available storage capacity of the pool. Such a thin device configuration is said to be oversubscribed or over-provisioned. There are options available to the user to control whether or not an over-provisioned thin pool will be allowed.

Virtual Provisioning provides these important benefits:

1. The data is effectively wide striped across all the disks in the pool, thereby greatly reducing or eliminating hot spots and improving overall performance of the array.
2. The array is positioned for active performance management at both the sub-volume and sub-dataset level using FAST VP.

Performance

The performance implications of using Virtual Provisioning depend on the nature of the workload and the state of the thin device. In any thin device implementation, there are response time and throughput overheads that are incurred the first time a write is performed on an unallocated region of a thin device. In the Symmetrix implementation, these overheads are quite modest and, in most cases, are undetectable. The overhead applies primarily to the first write to a thin device extent and disappear altogether once the working set of a thin device has been written to.

The most important aspect to understand about thin device performance is how thin devices are spread across the back end. As already discussed, the track groups for thin devices are dispersed (widely striped) across the back end, typically spanning a much greater number of drives than a regular device. Because of this, thin devices make it easier to avoid skews in the back-end workload. The result can be improved performance for most I/O workloads.

Virtual Provisioning: Configuration guidelines

Creating a Virtual Provisioning environment that results in high performance and availability requires adherence to specific configuration standards.

When planning a configuration using thin devices, the first step involves determining how many separate thin pools are needed, and the required composition of each thin pool. Typically, this involves conceptually organizing drive storage into separate classes, with further subdivision, as needed, to allow the back-end resources (drives and DAs) used by the pools to be isolated from one another. Depending on the mix of applications to be placed on thin devices, it is often necessary to create multiple thin pools. However, the most efficient use of resources is achieved by using a minimal number of pools.

A thin pool should be designed for use by a given application, or set of related applications, aligned with a given business group. The applications sharing a thin pool compete for back-end resources, including thin pool storage capacity, so applications should not share the same pool if this is not acceptable. The devices comprising a thin pool have the same performance and protection properties, so the applications sharing a thin pool should have the same performance and protection requirements.

Once a set of thin pools has been designed, the back-end layout of future storage provisioning requests may be planned by simply considering the class of storage that is required and which business group is requesting the storage (if that class has been further sub-divided).

Sizing resources for thin devices

The initial storage capacity of a thin pool must be large enough to accommodate the initial storage requirements of the applications using the pools. The initial storage requirements must be well understood, and there must always be enough physical space available to the thin devices to accommodate this initial requirement.

Besides storage capacity for thin devices, there are other processes that reserve capacity in thin pools. For example, if a thin pool is enabled for compression of bound thin devices, storage space is reserved in the pool that will be used to temporarily decompress data.

If a thin pool is later defined to be part of a Fully Automated Storage Tiering (FAST) storage tier, FAST ensures that a user-defined percentage of the pool's capacity is unallocated (pool reserved capacity) before FAST proceeds with any data movement into the pool.

Within each data device of a thin pool there are tracks reserved for allocation metadata. There is a minimum of 12 tracks reserved on each data device. The 12 tracks of metadata hold allocation maps for up to 76,800 cylinders of a data device. Data devices larger than 76,800 cylinders require more metadata space. Currently, the largest data device that can be configured for a thin pool is 256,000 cylinders. The following table shows the metadata tracks reserved for data device sizes by cylinder ranges.

Data device (TDAT) size by cylinder range		Reserved metadata tracks
Minimum	Maximum	
48	76,800	12
76,801	153,600	24
153,601	230,400	36
230,401	256,000	48

Table 2. Metadata tracks reserved by data device size

When a fully pre-allocated strategy is being employed, the data device sizes must be sized to account for the aggregate thin device capacity, as well as the potential reserved and metadata overheads in the pool.

If an oversubscribed pool is being used to permit inexpensive pre-provisioning, then there should be enough additional capacity to ensure that there is time for the storage administrator to add capacity to the pool when it is determined that the risk of the pool filling is imminent.

The data devices comprising a thin pool should be spread across a back-end hardware configuration that has enough available performance capacity to handle the I/O workload for both the near- and long-term requirements. After determining the DA and drive configuration of the initial thin pool layout, adding storage capacity to the pool at a later time can be done in a straightforward manner. If, however, the initial pool was not spread over enough drives and DAs to accommodate the workload on the initially mapped regions of thin devices, adding data devices to the pool, even if they are on separate physical drives, does not automatically alleviate the problem.

Additional cache requirements

Because thin devices are cache devices, the use of thin devices introduces some additional cache requirements. Though minimal, these requirements should be considered when designing a thin device configuration. The EMC sales tool Direct Express should be used to determine the additional cache requirements.

Thin pool layout considerations

Many of the same considerations that apply to the design of other types of Symmetrix pools also apply to the design of thin pools. The devices comprising a given thin pool should satisfy all of the following requirements:

- Only data devices may be placed in a thin pool.
- The data devices must all have the same emulation.
- The data devices must all have the same protection type, but cannot be RAID 10.

  Recommended protection by class of drive:

Drive class	Protection type
EFD	RAID5 (3+1)
FC/SAS	RAID1
SATA	RAID6 (6+2)

If there are SRDF® relationships for all bound thin devices, RAID5 (3+1) is viable for the thin pool.

- It is recommended that data devices in a pool all reside on drives that have the same rotational speed. FAST VP enforces this restriction if the pool is added to a FAST VP tier.
- The data devices in the pool should generally be spread across as many DAs and drives of a given speed as possible. The wide striping provided by Virtual Provisioning disperses thin device track groups evenly across the data devices. However, when adding data devices to a pool, they should be evenly spread across the back end.
- It is recommended that all data devices in a pool are of the same size. Using different size devices could result in uneven data distribution.

- The data device sizes should be as large as possible to minimize the number of devices required to encompass the desired overall pool capacity.
- For most drive sizes and protection types, size data devices so that there are eight hyper volumes per physical drive to maintain adequate disk queue depth and to prevent excessive seeks.

Adding capacity to a thin pool running out of space

When an oversubscribed thin pool begins to run out of space, data devices should be added to the pool before the pool completely fills. Otherwise there is a risk that an application that performs a write to a previously unmapped portion of a thin device may encounter an out-of-space error. Data devices can be added to a thin pool non-disruptively. The set of data devices to be added to an existing thin pool should satisfy the following requirements:

- They must have the same protection type as the devices already in the target thin pool.
- They must have the same emulation as the devices already in the target thin pool.
- They should all reside on drives that have the same rotational speed.
- When considered on their own, the data devices should be evenly spread across a set of DAs and drives.
- It is recommended that all data devices in a pool are of the same size. Using different size devices could result in uneven data distribution.

Thin pool rebalancing

Thin pool rebalancing allows the user to rebalance workloads non-disruptively in order to extend thin pool capacity in small increments as needed. This maximizes performance and minimizes Total Cost of Ownership (TCO). Users can run a command against a thin pool that rebalances the assigned track groups across all enabled data devices in the pool. This allows a small number of data devices to be used to expand a pool without causing wide striping to be compromised.

Thin device compression

With Enginuity 5876 (Q4-2012 Service Release) and Mainframe Enablers (MFE) V7.5, thin device data can be compressed to save space and decompressed when desired. Support was added to MFE that allows users to manually compress or decompress thin device extents and provide configuration management and reporting on the compression state for thin devices.

Compression should only be used with thin devices with idle data. It is best not to run medium or greater workloads against compressed devices. Although compression does support reads and writes against the device, it can affect the performance of the entire subsystem. FAST VP automates virtual pool compression for thin devices that are managed by FAST VP. Only manual compression is supported by MFE.

Suggested use of thin device manual compression (not FAST):

- Archiving old user files
- Decompress-use-recompress end-of-quarter activity

Not a suggested use for manual compression:

- Low-cost/low-performance active data

Enginuity allows data that is compressed to be both read and written. To allow this, Enginuity decompresses the data, when needed, into space that is reserved in the thin pool containing the compressed thin device allocations. To reserve this space, and to enable allocations in a pool to be compressed, a pool must be enabled for compression. When compression is no longer desired for allocations in a pool, the compression capability of a pool can be disabled.

A read of a compressed track temporarily decompresses the track into temporary storage maintained in the pool. The space in this temporary storage is controlled by a least-recently-used (LRU) algorithm ensuring that a track can always be decompressed and that the most recently utilized tracks remain available in a decompressed form.

Writes always write the decompressed allocations on the thin device. The device can be manually compressed at any time. Decompression happens upon the user's request or when the data is written. There is no automatic recompression after a write.

When a user chooses to compress a device, only allocations that have been written are compressed. Any allocations that were created during the bind or allocate processing that have not been written will be reclaimed during the compression process. Note, the zero reclaim that runs as part of the compression process deallocates standard record-zero tracks but does not scratch deleted datasets that do not yet contain standard record-zero tracks. In order to reclaim these tracks and avoid compressing deleted datasets, the Thin Reclaim Utility (TRU) should be run against a thin device before compressing it.

The fact that unwritten allocations can be reclaimed imposes a restriction that persistent allocations cannot be compressed, since they are not allowed to be reclaimed automatically. In order to compress a thin device that has persistent allocations, which implies the possible reclamation of unwritten allocations, the persistent attribute must first be removed, allowing compression and reclamation to occur.

Compression of allocations for a thin device is a background task. Once the compression request is accepted, the thin device gets a background task associated with it that performs the compression for that device.

If a thin device containing allocations in multiple pools is compressed (which is possible in a FAST VP environment), the MFE compression function first moves all allocations for the device to the pool that the device is currently bound to before compressing those allocations.

Therefore, after compressing a thin device, all allocations for that device reside in the bound pool.

Thick-to-thin migration considerations

Early planning for any data migration is a key factor that contributes to success, and that practice holds true for thick-to-thin migrations as well. Once you decide to use thin devices, you may choose either host-based copy methods or array-based replication technologies. Ultimately, the choice in copy methods may come down to which method is disruptive or not to the application that accesses the data.

- Data can be migrated from existing devices to larger, oversubscribed thin devices, using either array-based or host-based migration methods.
- A host-level copy mechanism can be used to copy data from regular source volumes to thin target volumes.

The following table shows examples of migration choices with further considerations noted.

Host-based copy	**Disruptive to application (access must be interrupted)**	**Notes**
DFDSS, FDRDSF, other utilities	YES	Offers dataset extent consolidation
DFSMS re-allocation	YES	Redefinition of batch datasets results in migration to a new SMS group of thin devices with volume selection changes in ACS routines
EMC z/OS Migrator	NO	Volume and dataset level in single product; smaller to larger volume REFVTOC performed
TDMF (ZDMF)	NO	TDMF=Volume level product; ZDMF=dataset level product; smaller-to-larger volume REFVTOC performed
FDRPAS (FDRMOVE)	NO	FDRPAS=Volume level product; FDRMOVE=dataset level product; smaller to larger volume REFVTOC performed
VMAX array-based replication	**Disruptive to application (access must be interrupted)**	**Notes**
EMC SRDF/Data Mobility (Adaptive Copy)	YES	*Check EMC Support matrix for supported Enginuity code levels.* See "SRDF thick R1 reclaim feature" section.
EMC TimeFinder®/Clone (SNAP VOLUME)	YES	See example that follows
EMC TimeFinder/Clone	YES	

(SNAP DATASET)		

Table 3. Thick-to-thin sample migration matrix

Local and remote replication and Virtual Provisioning

Both EMC TimeFinder and EMC SRDF replication products are supported with CKD thin devices. For local replication, a thin device can be the source or the target or both of a TimeFinder/Clone action and a source of a TimeFinder/Snap action (a target snap Virtual Device (VDEV) must map to the SAVE pool).

For remote replication, a thin device is supported in SRDF/S, SRDF/A, and SRDF/DM modes, in all SRDF topologies (concurrent and cascaded). Reclaim processing for thick R1 devices with thin-related R2 devices is supported with the Mainframe Enablers (MFE) V7.5. The Thin Reclaim Utility (TRU) of Symmetrix Control Facility (SCF) in that release is described later in the "Thin pool space reclamation" section.

Data devices are not host addressable and may not be used as the source or target of any TimeFinder or SRDF operation.

Virtual Provisioning with Mainframe Enablers (MFE)

Mainframe Enablers(MFE) V7.4 and later contains an enhanced pool-management function that provides support for managing virtual pools, as well as save pools for TimeFinder/Snap virtual devices, and DSE pools for SRDF/A Delta Set Extension.

The support is implemented by way of a new Symmetrix Control Facility (SCF) environment called General Pool Maintenance (GPM) and has a z/OS MODIFY command interface, a batch utility interface, and a z/OS MODIFY command interface that can also use the SCF command Prefix facility (refer to the initialization parameter SCF.INI.CPFX).

Using GPM, the customer can perform both virtual-pool-oriented functions and thin-device-oriented functions as listed below:

- Pool-oriented commands:
    - CREATE—Create a thin, DSE, or snap device pool
    - DELETE—Delete an empty device pool
    - DISPLAY—Display thin pool information
    - POOLATTR—Specify pool-level attributes
    - REBALANCE—Initiate leveling of allocated tracks within a pool
    - RENAME—Change the name of a thin pool
- Device-oriented commands:
    - ADD—Add one or more back-end devices to a thin pool
    - ALLOCATE—Causes assignment of all thin device tracks to data devices
    - BIND—Bind one or more thin devices to a thin pool

- COMPRESS—Compress data for thin devices (MFE V7.5 and Enginuity 5876 Q4-2012 Service Release)
- DECOMPRESS—Decompress data for thin devices (MFE V7.5 and Enginuity 5876 Q4-2012 Service Release)
- DISABLE—Change the status of one or more pool devices from active to inactive
- DRAIN—Initiate reassignment of allocated tracks to other pool devices
- ENABLE—Change the status of one or more pool devices from inactive to active
- HDRAIN—Halt draining activity of pool devices
- QUERY—Display information about thin provisioning device entities
- MOVE—Causes movement of assigned tracks from prior bound pool to current pool
- REBIND—Change the pool to which a thin device is bound
- REMOVE—Remove an inactive device from a device pool
- UNBIND—Unbind one or more thin devices from a thin device pool
- USR_NRDY—Change device's control unit status to user-not-ready
- USR_RDY—Change device's control unit status to user-ready

General Pool Maintenance (GPM) usage examples

The following are examples of how to configure, update, and monitor a Symmetrix Virtual Provisioning environment using GPM commands and the thin pool monitor implemented in the Symmetrix Control Facility. The examples used in this white paper have been created for illustrative purposes and do not represent a Virtual Provisioning environment configured for production workloads.

Creating and viewing thin pools

Creating thin pools and associating thin devices with thin pools can be summarized in the following steps:

1. CREATE a pool
2. QUERY DATADEV to find available devices
3. ADD data devices to a pool
4. QUERY THINDEV for available thin devices
5. BIND thin devices to the pool
6. DISPLAY pool
7. QUERY thin and data devices

The thin devices are now ready for ICKDSF INITialization and general use. Details on each step are provided below.

Thin pools are created using the CREATE command. They can be created without adding data devices, so that they can be populated at a later time. In this example, a pool called ZOS_FC_2M is created:

```
CREATE LCL(UNIT(1000) POOL(ZOS_FC_2M) TYPE(THINPOOL)
```

Sample response:

```
EMCU500I  CREATE LCL(UNIT(1000)) POOL(ZOS_FC_2M) TYPE(THINPOOL)
EMCU002I GPM command successful
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
EMCU008I END OF COMMANDS FILE REACHED.
```

After a pool is created, data devices can be added to the pool and enabled. You can perform a query of data devices to find available devices (from the default pool DF_DDEV_POOL or .NOPOOL.) to add to pools using the following command.

```
QUERY DATADEV LCL(UNIT(1000))
```

Sample response:

```
EMCU500I  QUERY DATADEV LCL(UNIT(1000))
EMCU184I Data Devices on 0001957-00079                        API Ver: 7.40
EMCU061I Device#  Emul      Used      Free  Pool Name    Type  Class    Speed    Prot
EMCU063I 00000270  3390         0     16680  MFCKD3       Thin  FIBRE    10K      RAID1
EMCU063I 00000271  3390         0     16680  MFCKD3       Thin  FIBRE    10K      RAID1
EMCU063I 00000272  3390     13920      2760  MFCKD3       Thin  FIBRE    10K      RAID1
EMCU063I 0000028A  3390                      DF_DDEV_POOL       FIBRE    10K      RAID1
EMCU063I 0000028B  3390                      DF_DDEV_POOL       FIBRE    10K      RAID1
EMCU063I 0000028C  3390                      DF_DDEV_POOL       FIBRE    15K      RAID1
EMCU063I 0000028D  3390                      DF_DDEV_POOL       FIBRE    15K      RAID1
EMCU063I 0000028E  3390                      DF_DDEV_POOL       FIBRE    15K      RAID1
EMCU063I 0000028F  3390                      DF_DDEV_POOL       FIBRE    15K      RAID1
EMCU063I 00000290  3390                      DF_DDEV_POOL       FLASH    FLASH    RAID1
EMCU063I 00000291  3390                      DF_DDEV_POOL       FLASH    FLASH    RAID1
```

Devices can exist in a thin pool and be in a disabled state, but disabled devices are not available for track group assignments. The ADD command can also be used to add data devices to an existing pool and activate (enable) them in a single step:

```
ADD LCL(UNIT(1000) POOL(FC_2M) DEV(28C-28D) ACTIVE
```

Sample response:

```
EMCU500I  ADD  LCL(UNIT(1000)) DEV(28C-28D) POOL(FC_2M)ACTIVE
EMCU009I Requested devices
EMCU009I  028C-028D
EMCU00AI Eligible devices
EMCU00AI  028C-028D
EMCU00BI Completed devices
EMCU00BI  028C-028D
EMCU002I GPM command successful
```

After the thin pool has been populated with active data devices, thin devices can be bound to the thin pool. The QUERY command can be use to find available devices (unbound) for binding to the thin pools. Use the following command format to run the query.

```
QUERY THINDEV LCL(UNIT(1000))
```

Sample query response:

```
EMCU500I  QUERY THINDEV LCL(UNIT(1000))
EMCU184I Thin Devices on 0001957-00079                          API Ver: 7.40
EMCU108I Device#   CUU   Emul   Volser  Bound To       Rdy  S/E       Cyls
EMCU110I 00000160  3E70  3390   ******  *Unbound        N    N        3339
EMCU110I 00000161  3E71  3390   ******  *Unbound        N    N        3339
EMCU110I 00000162  3E72  3390   ******  *Unbound        N    N        3339
EMCU110I 00000163  3E73  3390   ******  *Unbound        N    N        3339
EMCU110I 00000164  3E74  3390   ******  *Unbound        N    N        3339
EMCU110I 00000165  3E75  3390   ******  *Unbound        N    N        3339
EMCU110I 00000168  3E78  3390   ******  *Unbound        N    N        3339
```

Now, to bind the thin devices to a thin pool, use the following GPM command.

BIND LCL(UNIT(1000) POOL(ZOS_FC_2M) DEV(160)

Sample response:

```
EMCU500I  BIND   LCL(UNIT(1000)) POOL(ZOS_FC_2M) DEV(160)
EMCU009I Requested devices
EMCU009I  0160
EMCU00AI Eligible devices
EMCU00AI  0160
EMCU00BI Completed devices
EMCU00BI  0160
EMCU002I GPM command successful
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

You can use the DISPLAY and QUERY commands to interrogate the Virtual Provisioning environment as follows:

DISPLAY LCL(UNIT(1000) POOL(FC_2M)

```
EMCU500I  DISPLAY  LCL(UNIT(1000)) POOL(FC_2M)
EMCU013I Devices in Thin Pool FC_2M     on 0001957-00079      API Ver: 7.40
EMCU014I Device#    Emul    State       Used        Free
EMCU015I 0000028C   3390    Act          60        16620
EMCU015I 0000028D   3390    Act          60        16620
EMCU001I GPM command complete
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
EMCU008I END OF COMMANDS FILE REACHED.
```

QUERY THINDEV LCL(UNIT(1000)) DEV(160)

```
EMCU500I QUERY THINDEV LCL(UNIT(1000)) DEV(160)
EMCU184I Thin Devices on 0001957-00079                         API Ver: 7.40
EMCU108I Device#   CUU   Emul   Volser  Bound To      Rdy S/E      Cyls
EMCU110I 00000160  3E80  3390   ******  FC_2M          Y   N       3339
EMCU001I GPM command complete
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

Growing and rebalancing a thin pool

An EMC virtually provisioned environment provides enhanced management features, like thin pool rebalancing after growing capacity in a thin pool.

Expanding and initiating a rebalancing process on thin pools can be summarized in two steps:

1. Add data devices to thin pool and enable them
2. Initiate the rebalance process on a thin pool

This use case sample is based on a scenario where a given thin pool has reached about 29 percent of used capacity and more data devices are non-disruptively added to the thin pool. A detailed display of the specific thin pool can be obtained using the following command:

DISPLAY LCL(UNIT(1000) POOL(ZOS_FC_2M)

Sample response:

```
EMCU500I  DISPLAY  LCL(UNIT(1000)) POOL(ZOS_FC_2M)
EMCU013I Devices in Thin Pool ZOS_FC_2M  on 0001957-00079 API Ver:7.40
EMCU014I Device#   Emul   State       Used         Free
EMCU015I 0000028C  3390   Act         4032        12648
EMCU015I 0000028D  3390   Act         4032        12648
EMCU015I 0000028E  3390   Act         4032        12648
EMCU015I 0000028F  3390   Act         4032        12648
EMCU001I GPM command complete
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

Expanding a thin pool is accomplished by adding new data devices to the pool and enabling them.

In this example, only two new devices are added, but, in fact, a device count that is a multiple of the number of DAs is strongly recommended:

To add data devices to the pool and enable them, use the following command.

ADD LCL(UNIT(1000) POOL(ZOS_FC_2M) DEV(290-291) ACTIVE

Sample response:

```
EMCU500I  ADD  LCL(UNIT(1000)) DEV(290-291) POOL(ZOS_FC_2M) ACTIVE
EMCU009I Requested devices
EMCU009I  0290-0291
EMCU00AI Eligible devices
EMCU00AI  0290-0291
EMCU00BI Completed devices
EMCU00BI  0290-0291
EMCU002I GPM command successful
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
EMCU008I END OF COMMANDS FILE REACHED.
```

The addition of the new devices makes the pool unbalanced. Figure 2 below shows an example with free space (shown in white) in a pool of six drives. The free space is not equally distributed across all drives.

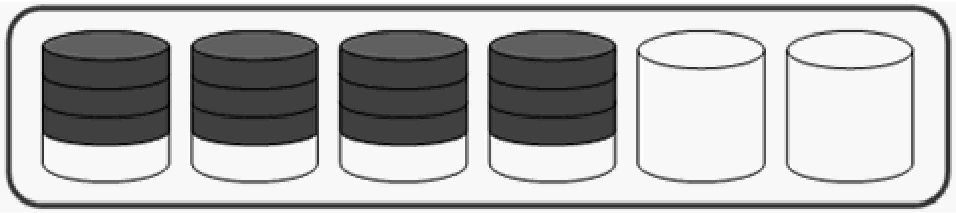

Figure 2. An unbalanced thin pool

A detailed display of the specific thin pool is obtained using the following command:

DISPLAY LCL(UNIT(1000) POOL(ZOS_FC_2M)

Sample response:

```
EMCU500I  DISPLAY  LCL(UNIT(1000)) POOL(ZOS_FC_2M)
EMCU013I Devices in Thin Pool ZOS_FC_2M      on 0001957-00379
API Ver: 7.40
EMCU014I Device#   Emul   State      Used        Free
EMCU015I 0000028C  3390   Act        4032       12648
EMCU015I 0000028D  3390   Act        4032       12648
EMCU015I 0000028E  3390   Act        4032       12648
EMCU015I 0000028F  3390   Act        4032       12648
EMCU015I 00000290  3390   Act           0       16680
EMCU015I 00000291  3390   Act           0       16680
EMCU001I GPM command complete
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

To rebalance the used tracks across all of the data devices in thin pool, use the following GPM command format:

REBALANCE LCL(UNIT(1000)) POOL(ZOS_FC_2M)

Sample response:

```
EMCU500I  REBALANCE  LCL(UNIT(1000)) POOL(ZOS_FC_2M)
EMCU002I GPM command successful
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
EMCU008I END OF COMMANDS FILE REACHED.
```

The rebalancing operation redistributes data across the enabled data devices in the thin pool.

The balancing algorithm measures the minimum, maximum, and mean used capacity values of the data devices in the thin pool. The Symmetrix subsystem then moves thin device extents from the data devices with the highest used capacity to those with the lowest until the pool device capacities are balanced.

Enginuity gives the highest priority to host I/O operations over rebalancing activities. However, rebalancing can be stopped for any reason, such as an upcoming period of very high host I/O or an operation that causes a large amount of internal copy tasks. The rebalancing can be resumed at a later time, continuing from where it left off.

After the operation has completed, the pool is balanced. Figure 3 depicts free space (shown in white) in a pool of six drives. The free space is now equally distributed across all drives, once again maximizing the efficiency of the wide striping set.

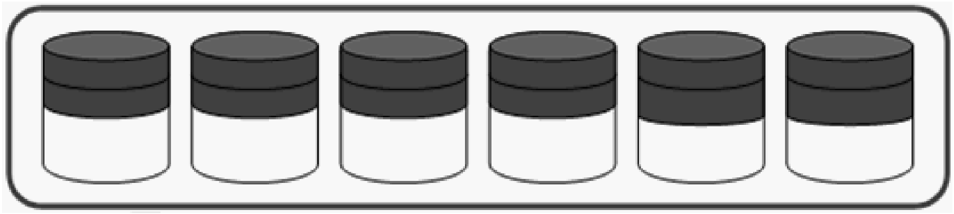

Figure 3. A rebalanced thin pool

A detailed display of the specific thin pool can be obtained using the following command:

DISPLAY LCL(UNIT(1000) POOL(ZOS_FC_2M)

Sample response:

```
EMCU500I  DISPLAY  LCL(UNIT(1000)) POOL(ZOS_FC_2M)
EMCU013I Devices in Thin Pool ZOS_FC_2M      on 0001957-00379
API Ver: 7.40
EMCU014I Device#   Emul   State       Used        Free
EMCU015I 0000028C  3390   Act         2688       13992
EMCU015I 0000028D  3390   Act         2666       14004
EMCU015I 0000028E  3390   Act         2666       14004
EMCU015I 0000028F  3390   Act         2688       13992
EMCU015I 00000290  3390   Act         2688       13992
EMCU015I 00000291  3390   Act         2666       14004
EMCU001I GPM command complete
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

Care should be used when migrating application devices whose back-end layout has already been carefully tuned. This is especially true if the tuning effort included isolating the back-end resources used by certain portions of the workload, and the application has stringent throughput or response time requirements. In such a case, simply migrating all devices to a single, large thin pool that is shared with other applications could cause performance degradation.

Compressing thin devices

With Enginuity 5876 (Q4-2012 Service Release) and Mainframe Enablers (MFE) V7.5, thin device data can be compressed to save space and decompressed when desired.

Compressing thin devices can be summarized in two steps:

1. Alter a thin pool's attribute to enable compression
2. Initiate the COMPRESS process on the desired thin devices

You can use the DISPLAY command to interrogate the attribute of a thin pool as follows:

DISPLAY LCL(UNIT(3A00)

Sample response:

```
EMCU500I DISPLAY LCL UNIT 3A00
EMCP001I GPM DISPLAY LCL UNIT 3A00
EMCU010I Pools on Controller 0001957-00086                     API Ver: 7.50
EMCU011I Pool name       Id  Typ Stat   Emul  Class  Speed  Alarms   MaxO   ActO  % Used  Compress
EMCU012I DEFAULT_POOL   0000  S
EMCU012I MFCKD1         0001  T  Avail  3390  FIBRE  10K    70  80             1    31    Disabled
EMCU012I MFTEST         0002  T  Avail  3390  FIBRE  10K    70  80             2     1    Disabled
EMCU012I DF_DDEV_POOL   0100  T
EMCU001I GPM command complete
```

Note that for pool MFCKD1, compression is disabled and the pool-used percentage is at 31 percent.

A column (Com) from the Query Thin Devices output shows that none of the devices in the pool are currently compressed. The Query Thin Devices command is shown below:

QUERY THIND LCL UNIT 3A00 POOL MFCKD1

Sample response:

```
EMCU500I QRY THIND LCL UNIT 3A00 POOL MFCKD1
EMCP001I GPM QRY THIND LCL UNIT 3A00 POOL MFCKD1
EMCU184I Thin Devices on 0001957-00086 Bound to Pool MFCKD1       API Ver: 7.50
EMCU108I Device#   CUU  Emul  Volser Rdy S/E      Cyls  Typ  Com  Task        Status
EMCU110I 00000160  3A70 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000161  3A71 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000162  3A72 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000163  3A73 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000164  3A74 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000165  3A75 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000166  3A76 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000167  3A77 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000187  3A97 3390  SYM187  Y   N       1113       N                Done
EMCU110I 00000E30  3B00 3390  ******  Y   N       1113  R1   N                Done
EMCU110I 00000E31  3B01 3390  MF3B01  Y   N       1113  R1   N                Done
EMCU110I 00000E32  3B02 3390  MF3B02  Y   N       1113  R1   N                Done
EMCU110I 00000E33  3B03 3390  MF3B03  Y   N       1113  R1   N                Done
EMCU110I 00000E34  3B04 3390  MF3B04  Y   N       1113  R1   N                Done
EMCU110I 00000E35  3B05 3390  MF3B05  Y   N       1113  R1   N                Done
EMCU110I 00000E36  3B06 3390  MF3B06  Y   N       1113  R1   N                Done
EMCU110I 00000E37  3B07 3390  MF3B07  Y   N       1113  R1   N                Done
EMCU110I 00000E38  3B08 3390  MF3B08  Y   N       1113  R1   N                Done
EMCU110I 00000E39  3B09 3390  MF3B09  Y   N       1113  R1   N                Done
EMCU110I 00000E3A  3B0A 3390  MF3B0A  Y   N       1113  R1   N                Done
EMCU110I 00000E3B  3B0B 3390  MF3B0B  Y   N       1113  R1   N                Done
EMCU110I 00000E3C  3B0C 3390  MF3B0C  Y   N       1113  R1   N                Done
EMCU110I 00000E3D  3B0D 3390  MF3B0D  Y   N       1113  R1   N                Done
EMCU110I 00000E3E  3B0E 3390  MF3B0E  Y   N       1113  R1   N                Done
EMCU110I 00000E3F  3B0F 3390  MF3B0F  Y   N       1113  R1   N                Done
EMCU001I GPM command complete
```

A query of the data devices in the pool shows used track counts per device and total percentage used. The Query Data Devices command is shown below:

QUERY DATADEV LCL UNIT 3A00 POOL MFCKD1

Sample response:

```
EMCU500I QRY DATAD LCL UNIT 3A00 POOL MFCKD1
EMCP001I GPM QRY DATAD LCL UNIT 3A00 POOL MFCKD1
EMCU184I Data Devices on 0001957-00086 in Pool MFCKD1              API Ver: 7.50
EMCU061I Device#   Emul  A/I     Used      Free  Class     Speed     Prot      Status
EMCU063I 00000281  3390  I          0     16680  FIBRE     10K       RD1
EMCU063I 00000282  3390  I          0     16680  FIBRE     10K       RD1
EMCU063I 00000283  3390  A       5424     11256  FIBRE     10K       RD1
EMCU063I 00000284  3390  A       5304     11376  FIBRE     10K       RD1
EMCU063I 00000285  3390  A       5400     11280  FIBRE     10K       RD1
EMCU063I 00000286  3390  A       5400     11280  FIBRE     10K       RD1
EMCU063I 00000287  3390  A       5304     11376  FIBRE     10K       RD1
EMCU063I 00000288  3390  A       5220     11460  FIBRE     10K       RD1
EMCU063I 00000289  3390  A       5388     11292  FIBRE     10K       RD1
EMCU063I 0000028A  3390  A       5304     11376  FIBRE     10K       RD1
EMCU063I 0000028B  3390  A       5496     11184  FIBRE     10K       RD1
EMCU063I 0000028C  3390  A       5376     11304  FIBRE     10K       RD1
EMCU063I 0000028D  3390  A       5292     11388  FIBRE     10K       RD1
EMCU063I 0000028E  3390  A       5412     11268  FIBRE     10K       RD1
EMCU063I 0000028F  3390  A       5376     11304  FIBRE     10K       RD1
EMCU063I 00000290  3390  A       5412     11268  FIBRE     10K       RD1
EMCU063I 00000291  3390  A       5412     11268  FIBRE     10K       RD1
EMCU063I 00000292  3390  A       5292     11388  FIBRE     10K       RD1
EMCU063I 00000293  3390  A       5328     11352  FIBRE     10K       RD1
EMCU063I 00000294  3390  A       5340     11340  FIBRE     10K       RD1
EMCU063I 00000295  3390  A       5400     11280  FIBRE     10K       RD1
EMCU063I 00000296  3390  A       5364     11316  FIBRE     10K       RD1
EMCU063I 00000297  3390  A       5448     11232  FIBRE     10K       RD1
EMCU064I Totals:
EMCU064I    3390:      112692 used tracks,    270948 free tracks,   29% used
EMCU064I    Act :      112692 used tracks,    237588 free tracks,   32% used
EMCU001I GPM command complete
```

Compression is disabled by default and must be enabled by issuing the POOLATTR command with COMPRESSION(ENABLE) parameter before thin devices bound to that pool can be compressed. The following command sequence enables compression and displays the Compression attribute:

POOLATTR LCL UNIT 3A00 COMPRESS(ENABLE) POOL MFCKD1

DISPLAY LCL 3A00

Sample response:

```
EMCU500I POOLATTR LCL UNIT 3A00 COMPRESS(ENABLE) POOL MFCKD1
EMCP001I GPM POOLATTR LCL UNIT 3A00 COMPRESS(ENABLE) POOL MFCKD1
EMCU002I GPM command successful



EMCU500I DISPLAY LCL UNIT 3A00
EMCP001I GPM DISPLAY LCL UNIT 3A00
EMCU010I Pools on Controller 0001957-00086                    API Ver: 7.50
EMCU011I Pool name      Id  Typ Stat   Emul  Class  Speed  Alarms   MaxO   ActO  % Used  Compress
EMCU012I DEFAULT_POOL  0000  S
EMCU012I MFCKD1        0001  T  Avail  3390  FIBRE  10K     70  80            1     32    Enabled
EMCU012I MFTEST        0002  T  Avail  3390  FIBRE  10K     70  80            2      1    Disabled
EMCU012I DF_DDEV_POOL  0100  T
EMCU001I GPM command complete
```

Sample command to initiate the compression of thin device track groups follows:

COMPRESS LCL UNIT 3A00 POOL MFCKD1 DEV E30 E3F

Sample response:

```
EMCU500I COMPRESS LCL UNIT 3A00 POOL MFCKD1 DEV E30-E3F
EMCP001I GPM COMPRESS LCL UNIT 3A00 POOL MFCKD1 DEV E30-E3F
EMCU009I Requested devices
EMCU009I  0E30-0E3F
EMCU00AI Eligible devices
EMCU00AI  0E30-0E3F
EMCU00BI Completed devices
EMCU00BI  0E30-0E3F
EMCU002I GPM command successful
```

The thin devices can now be identified as having compressed allocations by issuing a query on the thin devices as shown:

QUERY THIND UNIT 3A00 POOL MFCKD1

Sample response:

```
EMCU500I QRY THIND LCL UNIT 3A00 POOL MFCKD1
EMCP001I GPM QRY THIND LCL UNIT 3A00 POOL MFCKD1
EMCU184I Thin Devices on 0001957-00086 Bound to Pool MFCKD1        API Ver: 7.50
EMCU108I Device#    CUU   Emul  Volser Rdy S/E      Cyls  Typ  Com  Task        Status
EMCU110I 00000160  3A70  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000161  3A71  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000162  3A72  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000163  3A73  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000164  3A74  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000165  3A75  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000166  3A76  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000167  3A77  3390  ******  Y   N        1113  R1   N                Done
EMCU110I 00000187  3A97  3390  SYM187  Y   N        1113       N                Done
EMCU110I 00000E30  3B00  3390  ******  Y   N        1113  R1   Y                Done
EMCU110I 00000E31  3B01  3390  MF3B01  Y   N        1113  R1   Y                Done
EMCU110I 00000E32  3B02  3390  MF3B02  Y   N        1113  R1   Y                Done
EMCU110I 00000E33  3B03  3390  MF3B03  Y   N        1113  R1   Y                Done
EMCU110I 00000E34  3B04  3390  MF3B04  Y   N        1113  R1   Y                Done
EMCU110I 00000E35  3B05  3390  MF3B05  Y   N        1113  R1   Y                Done
EMCU110I 00000E36  3B06  3390  MF3B06  Y   N        1113  R1   Y                Done
EMCU110I 00000E37  3B07  3390  MF3B07  Y   N        1113  R1   Y                Done
EMCU110I 00000E38  3B08  3390  MF3B08  Y   N        1113  R1   Y                Done
EMCU110I 00000E39  3B09  3390  MF3B09  Y   N        1113  R1   Y                Done
EMCU110I 00000E3A  3B0A  3390  MF3B0A  Y   N        1113  R1   Y                Done
EMCU110I 00000E3B  3B0B  3390  MF3B0B  Y   N        1113  R1   Y                Done
EMCU110I 00000E3C  3B0C  3390  MF3B0C  Y   N        1113  R1   Y                Done
EMCU110I 00000E3D  3B0D  3390  MF3B0D  Y   N        1113  R1   Y                Done
EMCU110I 00000E3E  3B0E  3390  MF3B0E  Y   N        1113  R1   Y                Done
EMCU110I 00000E3F  3B0F  3390  MF3B0F  Y   N        1113  R1   Y                Done
EMCU001I GPM command complete
```

A display of the thin pool now shows compression enabled and the percent of tracks used is now at two percent (down from 31 percent). The following commands were used to display the the pool and query the data devices:

DISPLAY LCL 3A00

QUERY DATAD LCL UNIT 3A00 POOL MFCKD1 SUM

Sample response:

```
EMCU500I DISPLAY LCL UNIT 3A00
EMCP001I GPM DISPLAY LCL UNIT 3A00
EMCU010I Pools on Controller 0001957-00086                          API Ver: 7.50
EMCU011I Pool name       Id  Typ Stat   Emul  Class  Speed  Alarms   MaxO   ActO  % Used  Compress
EMCU012I DEFAULT_POOL   0000  S
EMCU012I MFCKD1         0001  T  Avail  3390  FIBRE  10K    70  80             1      2    Enabled
EMCU012I MFTEST         0002  T  Avail  3390  FIBRE  10K    70  80             2      1    Disabled
EMCU012I DF_DDEV_POOL   0100  T
EMCU001I GPM command complete


EMCU500I QRY DATAD LCL UNIT 3A00 POOL MFCKD1 SUM
EMCP001I GPM QRY DATAD LCL UNIT 3A00 POOL MFCKD1 SUM
EMCU064I Totals:
EMCU064I     3390:        9528  used tracks,    374112  free tracks,   2% used
EMCU064I     Act :        9528  used tracks,    340752  free tracks,   2% used
EMCU001I GPM command complete
```

TimeFinder thick-to-thin clone with auto bind/unbind

TimeFinder/Clone allows you to create and manage point-in-time copies within the VMAX Family. With virtually provisioned storage, the target of a clone operation can be a thin device that has added advantages with allocation utilization and performance through wide striping. A new automatic binding and unbinding feature brings the ease-of-use benefit of tiered storage flexibility to the cloning operation. Take the scenario where one point-in-time copy of a workload may favor a copy on low-cost SATA drives for a backup process and another copy on high-speed Flash drives for fast on-demand reporting process. Being able to automatically bind the target clone on the right storage tier, at the right time, and then automatically unbind and remove the clone allocated space when the point-in-time process is complete allows for quicker and more efficient use of storage resources.

The following is an example illustrating the steps for using the TimeFinder auto bind/unbind feature. Refer to *EMC Mainframe Enablers TimeFinder/Clone Snap Facility Version Product Guide* for more information on TimeFinder operations.

You can query thin devices to find available devices (unbound) for binding to pools using the following command:

QUERY THINDEV LCL(UNIT(3800))

Sample query response:

```
EMCU500I  QUERY THINDEV LCL(UNIT(3800))
EMCU184I Thin Devices on 0001957-00079                            API Ver: 7.40
EMCU108I Device#   CUU   Emul   Volser  Bound To      Rdy S/E       Cyls
EMCU110I 00000160  3E70  3390   ******  *Unbound       N   N        3339
EMCU110I 00000161  3E71  3390   ******  *Unbound       N   N        3339
EMCU110I 00000162  3E72  3390   ******  *Unbound       N   N        3339
EMCU110I 00000163  3E73  3390   ******  *Unbound       N   N        3339
EMCU110I 00000164  3E74  3390   ******  *Unbound       N   N        3339
EMCU110I 00000165  3E75  3390   ******  *Unbound       N   N        3339
EMCU110I 00000168  3E78  3390   ******  *Unbound       N   N        3339
```

To create and activate a clone session between a standard (thick) device and an unbound thin device, use the following TimeFinder job format:

```
//jobname  JOB   (EMC),pgmmrname,CLASS=A,MSGCLASS=X
//CLONEJOB EXEC  PGM=EMCSNAP,REGION=0k
//STEPLIB  DD    DISP=SHR,DSN=ds-prefix.LINKLIB
```

```
//SCF$nnnn DD    DUMMY
//SYSPRINT DD    SYSOUT=*
//QCOUTPUT DD    SYSOUT=*
//QCINPUT  DD    *
SNAP VOLUME ( SOURCE (UNIT(13C5)) TARGET (UNIT(3E75)) NEWVOLID(CK3E75) -
   POOL(SATA_R6) AUTO_BIND_TDEV(YES) )
/*
```

Sample output:

```
 *** TIMEFINDER MF SNAP V7.4.0 (000) - SCF V07.04.00
ESNP010I BEGINNING COMMAND PARSE
ESNP011I PARSING STATEMENT #1
EMCP001I  GLOBAL MAXRC(4)
ESNP011I PARSING STATEMENT #2
EMCP001I  SNAP VOLUME (                                    -
EMCP001I  SOURCE(UNIT(13C5)) TARGET(UNIT(3E75)) NEWVOLID(CK3E75) -
EMCP001I  POOL(SATA_R6) AUTO_BIND_TDEV(YES)                -
EMCP001I  )
ESNP500I UNIT 13C5 WAS REQUESTED, FOUND WITH VOLUME CK13C5 MOUNTED
ESNP0K1I AUTO BIND OCCURRED FOR DEVICE 00 TO POOL SATA_R6
ESNP504I UNIT 3E75 WAS REQUESTED, FOUND OFFLINE
ESNP011I PARSING STATEMENT #3
ESNPQ55I AUTOMATIC_ACTIVATE ADDED TO REQUEST STREAM FOLLOWING STATEMENT #2
ESNP017I COMMAND PARSE COMPLETE
ESNP040I PROCESSING REQUESTS
ESNP460I PROCESSING FOR STATEMENT #2 BEGINNING, COPY FROM VOLUME CK13C5 TO VOLUME *3E75*
ESNPJ31I PROCESSING FOR STATEMENT #2 SUSPENDED FOR PENDING ACTIVATE
ESNPF50I PROCESSING FOR STATEMENT #3 BEGINNING, ACTIVATE SNAP
ESNP112I COPY HAS BEEN INITIATED FOR 1 EXTENT(S) - 150255 TRACK(S)
         - FROM VOLUME CK13C5 (S/N 0001957-00455/00000147) TO VOLUME *3E75* (S/N 0001957-00455/00000165)
ESNPF51I PROCESSING FOR STATEMENT #3 COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
ESNPJ30I PROCESSING FOR STATEMENT #2 RESUMED, COPY FROM VOLUME CK13C5 TO VOLUME *3E75*
ESNP461I PROCESSING FOR STATEMENT #2 COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0

ESNP440I PROCESSING COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
```

You can query the thin device used as a clone target by issuing the following command:

QUERY THINDEV LCL(UNIT(13C0))DEV(165)

Sample query response:

```
EMCU500I  QUERY THINDEV LCL(UNIT(13C0) DEV(165)
EMCU184I Thin Devices on 0001957-00455                    API Ver: 7.40
EMCU108I Device#  CUU   Emul  Volser  Bound To    Rdy S/E     Cyls
EMCU110I 00000165  3E75  3390  ****** SATA_R6      Y   N     10017
EMCU001I GPM command complete

EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

You can display the thin pool using the following command:

QUERY THINDEV LCL(UNIT(13C0)) POOL(SATA_R6)

Sample display reponse:

```
EMCU500I  DISPLAY LCL(UNIT(13C0) POOL(SATA_R6)
EMCU013I Devices in Thin Pool SATA_R6 on 0001957-00455        API Ver: 7.40
EMCU014I Device#   Emul   State     Used      Free
EMCU015I 00000356  3390   Act        12    3939948
EMCU015I 00000357  3390   Act        12    3939948
EMCU015I 00000358  3390   Act        12    3939948
EMCU015I 00000359  3390   Act        12    3939948
EMCU015I 0000035A  3390   Act        12    3939948
EMCU015I 0000035B  3390   Act        12    3939948
EMCU015I 0000035C  3390   Act        12    3939948
EMCU015I 0000035D  3390   Act        12    3939948
EMCU015I 0000035E  3390   Act        12    3939948
EMCU001I GPM command complete
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

You can stop the clone session (STOP SNAP) with unbind by using the following command format:

STOP SNAP TO VOLUME (TARGET(UNIT(3E75)) AUTO_UNBIND_TDEV(YES))

Sample output:

```
 *** TIMEFINDER MF SNAP V7.4.0 (000) - SCF V07.04.00 (0000)
ESNP191I BCVGROUP STATEMENT #1
ESNP010I BEGINNING COMMAND PARSE
ESNP011I PARSING STATEMENT #1
EMCP001I  GLOBAL MAXRC(4)
ESNP011I PARSING STATEMENT #2
EMCP001I  STOP SNAP TO VOLUME -
EMCP001I  ( TARGET(UNIT(3E75)) AUTO_UNBIND_TDEV(YES) )
ESNP504I UNIT 3E75 WAS REQUESTED, FOUND OFFLINE
ESNP011I PARSING STATEMENT #3
ESNP017I COMMAND PARSE COMPLETE
ESNP040I PROCESSING REQUESTS
ESNPC00I PROCESSING FOR STATEMENT #2 BEGINNING, STOP SNAP TO VOLUME *3E75*
ESNPC12I SNAP HAS BEEN STOPPED FOR 1 EXTENT(S) ON VOLUME *3E75* (S/N 0001957-00455/00000165)
ESNPI40I DEVICE SUCCESSFULLY MADE NOT-READY
ESNPC01I PROCESSING FOR STATEMENT #2 COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
ESNP440I PROCESSING COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0

ESNP440I PROCESSING COMPLETED, HIGHEST RETURN CODE ENCOUNTERED IS 0
```

If you display the thin pool after the STOP SNAP with the Auto_unbind option, you will notice that the space was automatically deleted.

DISPLAY LCL(UNIT(13C0) POOL(SATA_R6)

Sample response:

```
EMCU500I  DISPLAY LCL(UNIT(13C0) POOL(SATA_R6)
EMCU013I Devices in Thin Pool SATA_R6 on 0001957-00455       API Ver: 7.40
EMCU014I Device#   Emul   State      Used        Free
EMCU015I 00000356   3390   Act         0      3939960
EMCU015I 00000357   3390   Act         0      3939960
EMCU015I 00000358   3390   Act         0      3939960
EMCU015I 00000359   3390   Act         0      3939960
EMCU015I 0000035A   3390   Act         0      3939960
EMCU015I 0000035B   3390   Act         0      3939960
EMCU015I 0000035C   3390   Act         0      3939960
EMCU015I 0000035D   3390   Act         0      3939960
EMCU015I 0000035E   3390   Act         0      3939960
EMCU001I GPM command complete
EMCU006I COMMAND PROCESSED SUCCESSFULLY.
```

Device mobility with Virtual Provisioning

EMC Virtual Provisioning provides Enginuity technology (VLUN VP) that enables movement of Symmetrix thin devices from one thin pool to another without disrupting user applications and with minimal impact to host I/O. Users may move thin devices between thin pools to change disk media, RAID protection, or performance tier. VLUN VP is also the enabling technology for FAST VP, where only portions of a thin device (sub volume) are moved from one pool to another. The VLUN VP feature can be exploited with the GPM device-oriented command `MOVE`.

The `MOVE` command moves existing track-group assignments from a thin pool, where the thin devices were previously bound (as specified on the SRCPOOL parameter), to the pool where the thin devices are currently bound (as specified on the POOL parameter), without the loss of data. The track groups assigned for the devices in the source pool are moved to the pool where the devices are currently bound.

If the `REBIND` parameter is specified, the thin devices are first rebound to the target pool (specified on the POOL parameter), and the tracks allocated for the devices in the source pool are moved to the target pool where the devices are bound.

For a use case of the `MOVE` command, take the previous scenario where TimeFinder thick-to-thin clone with auto bind was used to take a point-in-time copy of database volumes to the low-cost SATA thin pool. The SATA pool can meet the I/O profile demands of a high-sequential, high-read-hit workload. However, if periodically an end user had a requirement to also use the point-in-time database copy for a high-priority random read-miss query workload, the EFD pool would be more appropriate storage pool. The GPM `MOVE` command can be exploited to accomplish the data mobility task.

If you display the EFD thin pool, you can see the used and free tracks as shown in the following command:

`DISPLAY LCL(UNIT(1000) POOL(ZOS_EFD_R5)`

Sample response:

```
15.44.50 S0022857  EMCU500I DISPLAY LCL(UNIT(1000)) POOL(ZOS_EFD_R5)
15.44.50 S0022857  EMCP001I GPM DISPLAY LCL(UNIT(3800)) POOL(ZOS_EFD_R5)
15.44.50 S0022857  EMCU013I Devices in Thin Pool ZOS_EFD_R5 on 0001957-00079   API Ver: 7.40
15.44.50 S0022857  EMCU014I Device#   Emul    State       Used        Free
15.44.50 S0022857  EMCU015I 00000880  3390    Act            0       16680
15.44.50 S0022857  EMCU015I 00000881  3390    Act            0       16680
15.44.50 S0022857  EMCU015I 00000882  3390    Act            0       16680
15.44.50 S0022857  EMCU015I 00000883  3390    Act            0       16680
15.44.50 S0022857  EMCU001I GPM command complete
```

You can issue the MOVE command (with REBIND) to non-disruptively rebind and move the track groups of the thin devices that were the target of the TimeFinder/Clone snap volume command using auto bind. The following is an example of the GPM MOVE command:

```
MOVE LCL(UNIT(100)) DEV(165) POOL(ZOS_EFD_R5) -
SOURCEPOOL(ZOS_SATA_R6) REBIND
```

Sample response:

```
15.45.44 S0022857  EMCU500I MOVE LCL(UNIT(100)) DEV(165) POOL(ZOS_EFD_R5) -
15.45.44 S0022857  EMCU500I SOURCEPOOL(ZOS_SATA_R6) REBIND
15.45.44 S0022857  EMCP001I GPM MOVE LCL(UNIT(100)) DEV(165) POOL(ZOS_EFD_R5) -
15.45.44 S0022857  EMCU500I SOUCEPOOL(ZOS_SATA_R6) REBIND
15.45.44 S0022857  EMCU009I Requested devices
15.45.44 S0022857  EMCU009I  0165
15.45.44 S0022857  EMCU00AI Eligible devices
15.45.44 S0022857  EMCU00AI  0165
15.45.44 S0022857  EMCU00BI Completed devices
15.45.44 S0022857  EMCU00BI  0165
15.45.44 S0022857  EMCU002I GPM command successful
```

Again, display the EFD thin pool and observe the used and free tracks in the output:

```
DISPLAY LCL(UNIT(1000) POOL(ZOS_EFD_R5)
```

Sample response:

```
15.54.30 S0022857  EMCU500I DISPLAY LCL(UNIT(1000)) POOL(ZOS_EFD_R5)
15.54.30 S0022857  EMCP001I GPM DISPLAY LCL(UNIT(3800)) POOL(ZOS_EFD_R5)
15.54.30 S0022857  EMCU013I Devices in Thin Pool ZOS_EFD_R5 on 0001957-00079   API Ver: 7.40
15.54.30 S0022857  EMCU014I Device#   Emul    State       Used        Free
15.54.30 S0022857  EMCU015I 00000880  3390    Act           36       16644
15.54.30 S0022857  EMCU015I 00000881  3390    Act           36       16644
15.54.30 S0022857  EMCU015I 00000882  3390    Act           24       16656
15.54.30 S0022857  EMCU015I 00000883  3390    Act           24       16656
15.54.30 S0022857  EMCU001I GPM command complete
```

The point-in-time database copy for the high-priority, random read-miss query workload can be started as soon as the MOVE/REBIND job completes.

Monitoring the Virtual Provisioning pools

The Symmetrix Control Facility currently supports persistent monitors for the SAVE device pools used by TimeFinder/Snap and the DSE pool used by the SRDF/A Delta Set Extension feature. MFE V7.4 extends this monitoring function to virtual-pool capacity on behalf of Virtual Provisioning. The controls for the thin device monitor are identical to those for SAVE and DSE monitors and are denoted by THN in the SCF monitor control statement syntax. Values for alerts and actions can be set at a global, controller or individual pool level. An example of global setting for the thin device monitor follows:

```
SCF.THN.LIST=ENA
SCF.THN.DEBUG=YES
SCF.THN.STATS=YES
SCF.THN.01.LIST=PERCENT(01,40)              For pool utilization btwn 1 and 40%
SCF.THN.01.LIST=DURATION=20                 check every 20 min
SCF.THN.01.LIST=ACTION=MES(LOWUSAGE)        Issue this message
SCF.THN.01.LIST=FREQUENCY=REPEAT            Perform action each time or ONCE
SCF.THN.02.LIST=PERCENT(41,75)
SCF.THN.02.LIST=DURATION=05
SCF.THN.02.LIST=ACTION=MES(MEDUSAGE)
SCF.THN.02.LIST=FREQUENCY=REPEAT
SCF.THN.03.LIST=PERCENT(76,90)
SCF.THN.03.LIST=DURATION=05
SCF.THN.03.LIST=ACTION=MES(NEARFULL)
SCF.THN.03.LIST=FREQUENCY=REPEAT
```

The following are examples of the z/OS console messages that are generated based on THN pool monitor settings:

```
SCF1161I THN POOL 0001956-00057 - FCPOOL IS AT 3% UTILIZATION OF THINPOOL SPACE - LOWUSAGE
SCF1161I THN POOL 0001956-00057 - FCPOOL IS AT 42% UTILIZATION OF THINPOOL SPACE – MEDUSAGE
SCF1161I THN POOL 0001956-00057 - FCPOOL IS AT 88% UTILIZATION OF THINPOOL SPACE – NEARFULL
```

ACTION specifications can also be used to name a user exit module that would be invoked at set thresholds. For example, the exit may be used to request the Thin Reclaim Utility to perform SCAN/RECLAIM processing on all devices in a pool.

Thin pool space reclamation

The second major component of MFE V7.4 and later Virtual Provisioning support is another new Symmetrix Control Facility environment called the Thin Reclaim Utility (TRU).

For thin devices that are not bound with the PERSIST and PREALLOCATE attributes, TRU enables the reclamation of thin device track groups for reuse within the virtual pool by other thin devices. It does this by first identifying the free space in VTOC, initially by way of a scan function, then on an ongoing basis by the z/OS SCRATCH exit. It then periodically performs a reclaim operation, which marks tracks as empty in the Symmetrix array (no user records, only standard R0). The Symmetrix zero-space-reclaim background task then returns these empty track groups to the free list in the virtual pool. Figure 4 depicts this process graphically.

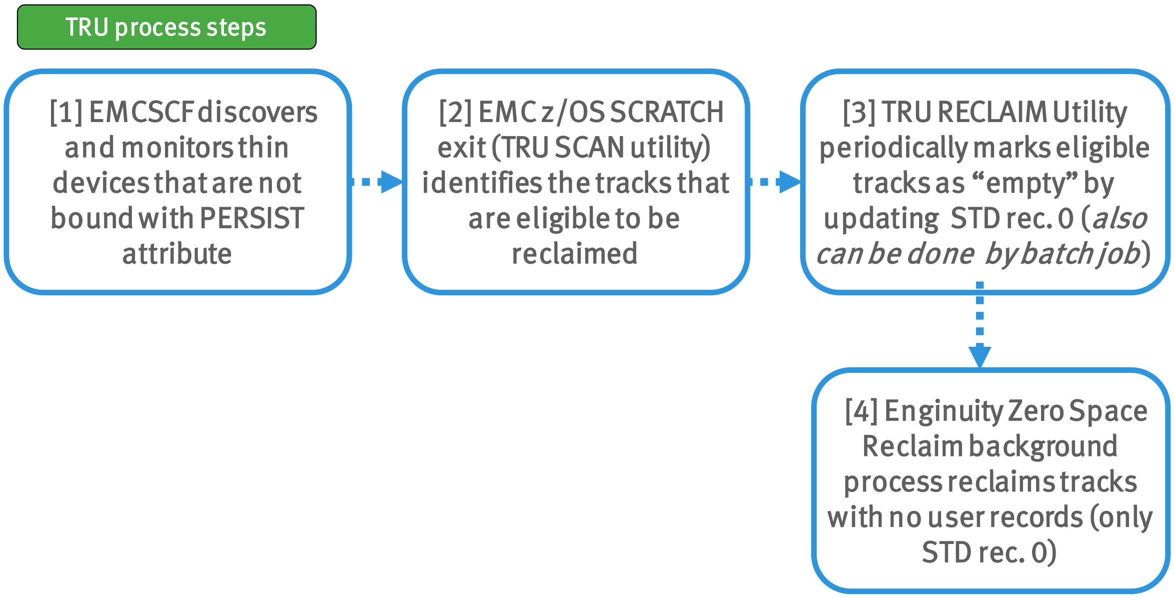

Figure 4. Thin Reclaim Utility (TRU) process steps

Note: Thin volumes that are bound with PERSIST attribute are not eligible for reclamation.

Space reclamation requires that the volume be channel-attached and have a control unit (CCUU) address available. It is recommended that the volume have an indexed VTOC. The TRU only monitors devices that have been included with the SCF.TRU.DEV.INCLUDE.LIST statement. An example of SCF initialization parameter settings for the Thin Reclaim Utility follows:

```
SCF.TRU.DEV.INCLUDE.LIST=3E00-3E80   Devices that TRU will monitor
SCF.TRU.RECLAIM.SCRATCH.WAIT=100     1 second wait before RECLAIM
SCF.TRU.RECLAIM.STCNAME=EMCTRCLM     RECLAIM by started task
SCF.TRU.RECLAIM.TASK.LIMIT=5         max simultaneously tasks
SCF.TRU.RECLAIM.METHOD=3             order segments large to small
```

TRU considerations and limitations:

- The z/OS Scratch Exit should be run on all LPARs attached to the Symmetrix in order to capture and record the scratch activity. If it is not running on an LPAR that is scratching datasets on thin devices, the space is not automatically reclaimed by this implementation.
- It is recommended that RECLAIM be set up to run as a started task in order to limit the amount of time a volume is RESERVED and the overall impact on SCF processing.
- Reclaim will not occur while a device has either active clone, snap, or virtual sessions.

The following example shows a thin device (MF3E80) with seven datasets and an indexed VTOC:

```
DSLIST - Data Sets on volume MF3E80                            Data Set - Browsed
 Command ===>                                                   Scroll ===> CSR

 Command - Enter "/" to select action                  Message            Volume
 -------------------------------------------------------------------------------
          PRODPS1T.CYL100.MF3E80.X1                    Browsed            MF3E80
          PRODPS1T.CYL100.MF3E80.X2                    Browsed            MF3E80
          PRODPS1T.CYL100.MF3E80.X3                    Browsed            MF3E80
          PRODPS1T.CYL100.MF3E80.X4                    Browsed            MF3E80
          PRODPS1T.CYL100.MF3E80.X5                    Browsed            MF3E80
          PRODPS1T.CYL100.MF3E80.X6                    Browsed            MF3E80
          PRODPS1T.CYL100.MF3E80.X7                    Browsed            MF3E80
          SYS1.VTOCIX.MF3E80                                              MF3E80
```

To display the thin pool and see the tracks used and free counts, you can use the following command:

DISPLAY LCL(UNIT(1000)) POOL(ZOS_FC_2M)

Sample response:

```
13.52.09 S0024069  EMCU500I DISPLAY LCL(UNIT(1000)) POOL(ZOS_FC_2M)
13.52.09 S0024069  EMCP001I GPM DISPLAY LCL(UNIT(1000)) POOL(ZOS_FC_2M)
13.52.09 S0024069  EMCU013I Devices in Thin Pool ZOS_FC_2M  on 0001957-00079   API Ver: 7.40
13.52.09 S0024069  EMCU014I Device#   Emul   State      Used        Free
13.52.09 S0024069  EMCU015I 0000028C  3390   Act        8388        8292
13.52.09 S0024069  EMCU015I 0000028D  3390   Act        8388        8292
13.52.09 S0024069  EMCU001I GPM command complete
```

If you scratch three of the datasets on the thin device volume MF3E80, the dataset list only shows four datasets and the index VTOC.

```
DSLIST - Data Sets on volume MF3E80                          Row 1 of 5
 Command ===>                                                Scroll ===> CSR

 Command - Enter "/" to select action                  Message           Volume
 -------------------------------------------------------------------------------
          PRODPS1T.CYL100.MF3E80.X1                                      MF3E80
          PRODPS1T.CYL100.MF3E80.X3                                      MF3E80
          PRODPS1T.CYL100.MF3E80.X5                                      MF3E80
          PRODPS1T.CYL100.MF3E80.X7                                      MF3E80
          SYS1.VTOCIX.MF3E80                                             MF3E80
```

To initiate the VTOC free space scan on-demand for the deleted datasets, run the utility to perform the SCAN task in batch.

Sample JCL:

```
//SCFTRU   JOB EMC,EMC,CLASS=A,MSGCLASS=X
//JS10     EXEC PGM=ESFTRURC,PARM='SCAN,3E80'
//STEPLIB   DD DISP=SHR,DSN=EMC.MFE740.LINKLIB
//ESFTRURC  DD SYSOUT=*
//SCF$MF74  DD DUMMY
//
```

To initiate the space reclaim on-demand for the deleted datasets, run the utility to perform the RECLAIM task in batch.

Sample JCL:

```
//SCFTRU   JOB EMC,EMC,CLASS=A,MSGCLASS=X
//JS10     EXEC PGM=ESFTRURC,PARM='RECLAIM,3E80'
//STEPLIB   DD DISP=SHR,DSN=EMC.MFE740.LINKLIB
//ESFTRURC  DD SYSOUT=*
//SCF$MF74  DD DUMMY
//
```

The RECLAIM utility analyzes the volume free space and SDDF session information, and then performs the empty track updates. Next, the Enginuity zero-space-reclaim feature is initiated. It detects tracks that only have a standard record zero present (and no other records) and returns those tracks to the thin free pool. This record-zero state is the normal state for a track in the thin free pool. If a host attempts to read a thin device track that is not assigned to physical storage, a dummy track image is constructed that looks like a track with home address and record zero present.

To see the results of the space reclamation process, you can display the thin pool and see changes in tracks used and free counts.

DISPLAY LCL(UNIT(1000)) POOL(ZOS_FC_2M)

Sample response:

```
14.04.57 S0024069  EMCU500I DISPLAY LCL(UNIT(1000)) POOL(ZOS_FC_2M)
14.04.57 S0024069  EMCP001I GPM DISPLAY LCL(UNIT(1000)) POOL(ZOS_FC_2M)
14.04.57 S0024069  EMCU013I Devices in Thin Pool ZOS_FC_2M     on 0001957-00079     API Ver:
7.40
14.04.57 S0024069  EMCU014I Device#   Emul    State      Used       Free
14.04.57 S0024069  EMCU015I 0000028C  3390    Act        5016       11664
14.04.57 S0024069  EMCU015I 0000028D  3390    Act        5100       11580
14.04.57 S0024069  EMCU001I GPM command complete
```

Space reclaimed

Note, that when you use ICKDSF to erase a thin volume, the SCF scratch exit is not invoked, and thus is not aware of the space being deleted. In this case, the batch utility should be run to reclaim all of the space for the volume.

SRDF thick R1 reclaim feature

With Enginuity 5876 (Q4-2012 Service Release) and Mainframe Enablers (MFE) V7.5, a new SRDF R1 reclaim process is added to the Thin Reclaim Utility to scan all R1 CKD devices and determine whether the R2 side (or any cascaded SRDF device) is a thin device. If an R1 is connected to a thin R2, the R1 is monitored and processed even if it is a thick device. This feature is enabled by coding a new SCF initialization parameter as follows:

```
SCF,TRU.THICKR1=YES   Enable support for thick R1 scanning (default=YES)
```

This provides support for downstream R2 devices, even if cascaded from an R21. This supports both thick and thin R1 devices. Concurrent, cascaded, and multiple site SRDF topologies are all supported.

The thin reclaim process for locally attached CKD devices is driven by the Thin Reclaim Utility environment in the SCF address space running on z/OS (see 'Thin pool space reclamation' on Page 28 for TRU information). With the SRDF thick R1 Reclaim feature the process for reclaiming tracks on the SRDF R2 is slightly different. The Enginuity zero-space-reclaim function is not initiated on the R2 control unit by TRU to run as a background task. Instead, the SRDF adapter on the R2 recognizes that the track has a standard R0 with no user records and actively does the reclaim itself as part of processing the write (or copy, in the case of a SRDF resynchronization). This is more of an *active* reclaim driven by SRDF activity on an I/O basis compared to the more *passive* reclaim done by an Enginuity background task that has been initiated by TRU on a volume basis. In the case of SRDF/A, it is worth noting that the reclaim occurs as part of the APPLY cycle. The SRDF-based process above applies to any thin R2 device, regardless of whether the R1 is thick or thin. In a thick-to-thin SRDF configuration, the host-based TRU SCAN and RECLAIM functions execute against thick R1 devices that are associated with thin R2 devices which have not been bound with the PERSIST attribute.

Conclusion

Enginuity 5876 marks a significant change in VMAX support for mainframe environments with the introduction Virtual Provisioning for CKD volumes. It is critical to define the processes, applications, and workloads for which Virtual Provisioning can be used most effectively, as well as the specific benefits that can be achieved. When implemented appropriately, Virtual Provisioning can be a powerful complement to organizations' processes and technologies for improving ease of provisioning, enhancing performance, and utilizing storage capacity more efficiently. Symmetrix Virtual Provisioning successfully integrates into z/OS Operating System environments, existing management and business continuity technologies, and is an important advancement in capabilities for Symmetrix VMAX array customers.

References

New Features in EMC Enginuity 5876 for Mainframe Environments

Implementing Fully Automated Storage Tiering for Virtual Pools (FAST VP) for EMC Symmetrix VMAX Series Arrays

DB2 for z/OS Best Practices with Virtual Provisioning

EMC Mainframe Enablers ResourcePak Base for z/OS Version 7.5 Product Guide

EMC Mainframe Enablers TimeFinder/Clone Snap Facility Version 7.5 Product Guide