LESS=MORE WITH VIRTUAL PROVISIONING AND LINUX ON SYSTEM Z

Gail Riley
EMC Corporation
August 15, 2013
Session Number 13534
Agenda

- Introduction to Virtual Provisioning
- Virtual Provisioning features for Linux on System z
  - FBA
  - CKD
- Virtual Provisioning Benefits
- Fully Automated Storage Tiering for Virtual Pools (FAST VP) Overview
Objectives

- Discuss the options for deploying virtual provisioning for both CKD and FBA devices in Linux on System z environment
- Understand the key components virtual provisioning
- Examine interrelationships that are built during the creation of virtual provisioned (thin) devices
Linux on System z Disk Attachment Options

IBM system z CPC (e.g. z10)

Channel Subsystem

FICON FCP FICON FCP

LPAR LPAR LPAR

z/VM

LPAR

FICON FCP

DASD DD DASD DD DASD DD DASD DD

SE SE SE SE

LINUX LINUX LINUX LINUX

5390 5390 5390

CKD CKD CKD

SCSI driver stack

CCW translation

EDEV 9336 (FBA)

DASD Device Drivers

FICON Director

FC SAN

/dev/dasdx

Linux device names

SE=Solutions Enabler (no minidisk support)

minidisk

dedicated

DASD DD= DASD Device Drivers

Complete your sessions evaluation online at SHARE.org/BostonEval
Virtual Provisioning = Thin Provisioning

- From wiki:
  - “Thin provisioning is the act of using virtualization technology to give the appearance of having more physical resources than are actually available.”
  - “Thin provisioning is a mechanism that applies to large-scale centralized computer disk storage systems, SANs, and storage virtualization systems. Thin provisioning allows space to be easily allocated to servers, on a just-enough and just-in-time basis.”
Data Layout – RAID group Allocation

- Capacity for a single logical volume is allocated from a group of physical disks
  - Example: RAID 5 with striped data + parity
- Workload is spread across a few physical disks
Data Layout – Pool-based Allocation Virtual Provisioning

- Storage capacity is structured in pools
- Thin devices are disk devices that are provisioned to hosts
Storage Requirement: Performance

- Storage Layout

  Go Wide Before Deep!

- Goal is to spread workload across all available system resources
  - Optimize resource utilization
  - Maximize performance

- Three approaches:
  - RAID data protection
  - Symmetrix Meta Devices
  - Virtual Provisioning
VP Components

- **Thin Data Device (TDAT)**
  - An internal, non-addressable device
  - Provides the physical storage for a thin device
  - Multiple RAID protection types
    - RAID 1, RAID 5, RAID 6

- **Thin Pool**
  - A shared, physical storage resource of a single RAID protection and drive technology
  - The first TDAT added determines the protection type for the pool

**Thin Pool**

- FC_Raid1
  - Add (4) x 25GB Raid 1 TDATs
VP Components

- Thin Device (TDEV) Host-addressable, cache only device
  - Bound to a thin pool and provisioned to hosts
  - Seen by the operating system as a “normal” device
  - Can be replicated both locally and remotely
  - Physical storage need not be completely allocated at device creation
  - Physical storage is allocated from a thin pool of DATA devices
- Thin Device Extent (aka track group for CKD)
  - unit of allocation from a thin pool when a host writes to a new area of a thin device
  - 12 Symmetrix tracks, (768 KB for FBA, 680KB for CKD)
Virtual Provisioning for FBA as SCSI devices with Linux on System z
VP Concepts for FBA as a SCSI LUN

- Thin Provisioning - SCSI
  - Space efficient technology
  - Data storage never 100% full
  - Present thin device to Linux
  - Only consumes storage as the host writes
  - Physical storage allocated from a shared pool

- Over Subscription
  - Thin device capacity > pool
Binding a Thin Device

- A thin device must be bound to a pool in order to be allocated any storage
- One extent is allocated from the pool when it’s bound
- Any write to a new area of a thin device will trigger an extent allocation from the pool the device is bound to
  - New allocations are performed using a round robin algorithm to spread extents across all of the enabled data devices in the thin pool
Virtual Provisioning Bind

- A thin device must be bound to a pool to allocate space
- Bind allocates initial extent in thin pool
Virtual Provisioning Writes

- Write to new area of tdev will allocate extents round robin across the pool
### VP Threshold Settings

**EMC Unisphere for VMAX** V1.5.0.6

#### Alert Thresholds

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>Category</th>
<th>Instance</th>
<th>State</th>
<th>Notification</th>
<th>Warning</th>
<th>Critical</th>
<th>Fatal</th>
</tr>
</thead>
<tbody>
<tr>
<td>000195700398</td>
<td>Fast VP Policy Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700398</td>
<td>Snap Pool Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700398</td>
<td>Thin Pool Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700455</td>
<td>Fast VP Policy Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700455</td>
<td>Snap Pool Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700455</td>
<td>Thin Pool Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700486</td>
<td>Fast VP Policy Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700486</td>
<td>Snap Pool Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>000195700486</td>
<td>Thin Pool Utilization</td>
<td>*</td>
<td>enabled</td>
<td></td>
<td>60%</td>
<td>80%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Complete your sessions evaluation online at SHARE.org/BostonEval
Attributes of a FBA/SCSI device Thin Pool

- A thin pool can be over subscribed
  - Provision more space than exists in the pool
- Maximum Subscription % - controls whether a pool can be over subscribed (allocated)
- Pool Reserve Capacity (PRC) – pools enabled capacity to be reserved for allocating new extents for the bound devices in the pool
Space Reclamation Feature for FBA/SCSI

• Available capacity in the thin pool can be maximized by returning unneeded extents
• Reclamation eligibility is based on
  • tracks that contain all-zero data
  • tracks Never Written By Host (NWBH)
Space Reclamation for SCSI LUN on Linux

- Reclaims thin pool storage by deallocating unnecessary track groups
  - Scans each track group and discards those containing all zeros
  - Deallocated tracks are presented as all zeros by Symmetrix to host

- Primary use is post migration from “thick” to “thin”
  - Migration performed using TimeFinder/Clone
  - Reclamation should be run prior to configuring any replication relationships
    - Thin devices in existing TimeFinder or SRDF relationships will be skipped
Thin Provisioning “cleanup”

- Cleanup terms are used loosely which can be confusing
- New host based SCSI commands* for thin device cleanup
  - SCSI unmap
  - SCSI write same with unmap
- SCSI standard (t10.org) - T10 Technical Committee on SCSI Storage Interfaces
- Support for these SCSI commands are
  - Kernel dependent – Linux vendor and release
  - Storage array dependent

* Any new technology should be tested and fully understood before being put into production
Thin Provisioning Cleanup from Linux on System z

- **SCSI commands**
  - Unmap - sent to thin device to unmap (or deallocate) one or more logical blocks
  - Write Same (with unmap flag) - writes at least one block and unmap(s) other logical blocks
- **fstrim** – executable, batch command used on filesystems
- **Discard**
  - option on mkfs and mount command for ext4 and xfs filesystems
  - controls if filesystem supports the SCSI unmap command so it can free specific blocks on thin devices at file deletion
Linux SCSI Cleanup Support Requirements

- Linux Releases supporting the discard option on the filesystem mount command
  - SLES* 11 SP2
  - RHEL* 6.2 with a hot fix and ext4
  - RHEL* 6.3 and ext4
  - LVM – RHEL - /etc/lvm.conf

- Storage Array
  - EMC VMAX 40k @ Enginuity 5876.159.102 + Epack (fix 65470)
  - *Check the vendor’s support matrix for the latest specific details
Verification of discard support

- Thin device must be mapped and masked to Linux
- Examine file(s) to verify discard support for the device
  - /sys/block/<device>/queue/discard_max_bytes

```
# cat /sys/block/sdc/queue/discard_max_bytes
25165824
```

- from kernel.org:
  
  “The discard_max_bytes parameter is set by the device driver to the maximum number of bytes that can be discarded in a single operation. Discard requests issued to the device must not exceed this limit. A discard_max_bytes value of 0 means that the device does not support discard functionality.”

Complete your sessions evaluation online at SHARE.org/BostonEval
Create ext4 filesystem with discard

- ext4 filesystem created with discard first discards blocks on thin device, then creates filesystem

```bash
# mke2fs -F -t ext4 -E discard -vvv /dev/sdb
mke2fs 1.41.12 (17-May-2010)
fs_types for mke2fs.conf resolution: 'ext4', 'default'
Discarding device blocks: done
Discard succeeded and will return 0s - skipping inode table wipe
```

Complete your sessions evaluation online at SHARE.org/BostonEval
mount ext4 with discard

- Filesystem mounted with the discard option
  - Frees up space on thin device at time of file deletion
  - And when the array receives the actual write request

NOTE: there is overhead associated with active discard so this should be tested in your own environment

```bash
mount -o discard -t ext4 /dev/sdb /thin_mount
# mount /dev/sdb on /thin_mount type ext4 (rw,discard)
```
Linux fstrim

- Filesystem mounted without the discard option
  - Does not frees up space on thin device at time of file deletion
- mount ext4 filesystem without discard mount option
- Use fstrim to free up space on a filesystem on a thin device, where files were previously deleted
- fstrim is executed against a filesystem and it’s underlying thin device
- Linux support –
  - release and vendor dependent
  - check vendor’s support matrix for proper support requirements
Virtual Provisioning for CKD devices with Linux on System z
Data Layout – Pool-based Allocation Virtual Provisioning

- Storage capacity is structured in pools
- Thin devices are disk devices that are provisioned to hosts
VP Components for CKD

- CKD VP components are same for CKD as they are for FBA:
  - Thin Pool – a shared, physical storage resource of a single RAID protection and drive technology
  - Data Device (TDAT) – RAID protected devices that provide the actual storage for a thin pool
  - Thin Device (TDEV) – cache only devices that are bound to a thin pool and provisioned to hosts
  - Track Group– allocation unit from a thin pool when a host writes to a new area of a thin device
    - 12 Symmetrix tracks, 680 KB (aka thin device extent)
VP for CKD with Linux on System z

• Thin CKD device supports z/VM and/or Linux on z
• Thin CKD device must be fully provisioned and persistent for z/VM and Linux
• Initial format of thin CKD device fully allocates device
  • cpfmtxa
  • dasdfmt
• Space reclamation and cleanup are not supported
Benefits of VP with CKD for Linux on System z

• Ease provisioning
• Wide striping for better performance
• EMC FAST VP – Fully Automated Storage Tiering
Common Functions of VP for CKD and FBA

• Underlying VP technology is the same for FBA and CKD therefore certain management activities are also the same
  • Pool Rebalancing
  • TDAT Drain – for device removal
  • Fully Automated Storage Tiering VP (FAST VP)
Automated Pool Rebalancing

- Rebalances allocated tracks across data devices contained within thin pool
- Levels out imbalances caused by:
  - Thin pool expansion
  - Unbinding thin devices from the thin pool
Pool Rebalancing

- Scheduled process that runs at given intervals
- Can be influenced by two extended pool attributes:
  - Rebalancing Variance %
    - controls whether a data device (TDAT) will be chosen for a possible rebalance
  - Maximum Rebalance Scan Device Range
    - the maximum number of data devices (TDATs) to concurrently balance at any one time
- Runs at a very low priority
VP Benefits

- Improved capacity utilization (with VP LUNs and Linux)
  - Reduces the amount of allocated but unused physical storage
  - Avoids over-allocation of physical storage to applications
- Efficient utilization of available resources
  - Wide striping distributes I/O across spindles
  - Reduces disk contention and enhances performance
  - Maximizes return on investment
- Ease and speed of provisioning
  - Simplifies data layout
  - Lowers operational and administrative costs
- Basis for Automated Tiering (FAST VP)
  - Active performance management at a sub-volume, sub dataset level
Virtual Provisioning with Tiers
Fully Automated Storage Tiering VP

- FAST VP is a policy-based system that promotes and demotes data at the sub-volume (LUN), and more importantly, file, which makes it responsive to the workload and efficient in its use of control unit resources.
- Performance behavior analysis is ongoing.
- Active performance management.
- FAST VP delivers all these benefits without using any host resources.
Storage Elements

FAST Storage Groups
- VP_Prod_DB2
- VP_QA_DB2

FAST Policies
- Automatic
  - <100%
  - 100%
- Custom
  - x%
  - y%
  - z%

Symmetrix Tiers
- R53_EFD_200GB
  - 200 GB EFD
  - RAID 5 (3+1)
- R1_FC_450GB
  - 450 GB 15K FC
  - RAID 1
- R66_SATA_1TB
  - 1 TB SATA
  - RAID 6 (6+2)
FAST VP Implementation – Task Segmentation

• Performance data
  • Analysis every 10 minutes
    • Provides thresholds
  • ‘Decay’ over time

• Intelligent Tiering
  • Thresholds from analysis
  • Performance move needs

• Allocation Compliance
  • Capacity move needs
Summary

- Virtual Provisioning is available for FBA/SCSI and CKD devices
- FBA as SCSI devices
  - Space is allocated as needed
  - Over subscription is allowed
  - Cleanup of unused space via space reclamation or T10 SCSI commands
  - Linux and Storage array dependent
- CKD for Linux on System z
  - Fully allocated
  - No reclaim or cleanup
- Wide Striping for better performance
- FAST VP – Fully Automated Storage Tiering VP
Thank you!