Understanding the Benefits of SCSI for Linux on z Systems
Agenda

- Storage device attributes
- Ease of administration
- Flexibility of FBA devices
- Solutions and innovation with SCSI fiber channel protocol
Please Note…

• Not recommending one technology over another, the focus is on the benefits.

• In the end, the technology is there, it is your decision on how to leverage it!
Common Disk Attachment Options

IBM z Systems CPC (e.g. z13)

FICON

LPAR

FICON

LPAR

FICON

LPAR

FCP

EDEV /CCW (FBA)

SE=Solutions Enabler (no minidisk support)

minidisk

dedicated

Linux device names

SE

FICON Director

FC SAN

DASD DD=
DASD Device Drivers

PR/SM

Channel Subsystem

SE

SE

SE

SE

DASD DD

DASD DD

DASD DD

DASD DD

3390

3390

3390

3390

CKD

CKD

CKD

CKD

/dev/dasdx

/dev/sdx

/dev/sdx

/dev/sdx

SCSI

FBA

FBA

FBA

8/12/2015

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Session 17494 Understanding the Benefits of SCSI in Linux on z Systems Environments
Fixed Block Architecture Device Basics

- FBA devices are fixed byte block (512 bytes)
- FBA device size limited by Linux kernel definition
  - Current limitation 2TB maximum
  - Variable device size
- Best use of physical device space
FBA as SCSI LUN devices

- Provision new FBA devices on storage array
- Dynamic LUN allocation to Linux
- Same protocol as used in open systems environment
- Multipath is handled by Linux on z Systems
  - Hardware independence
- Many databases utilize SCSI LUN devices
- Ability to exploit open systems features
  - e.g. – DB2 – the `no filesystem caching` option is supported for SCSI LUNs
Ease of Administration

• No format is required on a SCSI LUN
• No IOCDS change required
  – Except when NPIV is used, additional configuration needed
• No additional z/VM changes needed to provision additional SCSI LUNs to a Linux host
  – No directory changes, no additional mdisks
• Utilizes existing SAN infrastructure
Existing Infrastructure

- Use of existing SAN infrastructure used by open systems
- Use of existing FICON components
  - FICON Express cards
  - FC switches and cabling
Flexibility

- FBA devices
  - Defined as SCSI LUN to Linux
  - Defined as a emulated device (edev, 9336) to z/VM
- Both communicate to the storage array in SCSI fibre channel protocol
- SCSI LUN, or logical unit number
  - Number used to identify a logical unit, which is a device addressed by the SCSI protocol or protocols which encapsulate SCSI, such as Fibre Channel
Storage devices usually comprise many *logical units* - volumes, tape drives, etc.

A logical unit is identified by its Fibre Channel Protocol Logical Unit Number (FCP LUN).
Multipathing in Linux

- Multiple paths from OS to storage
- Why?
- Implemented in Linux in multipath-tools package, together with the device-mapper in the Linux kernel, or through 3rd party products
- SCSI device ("LUN") in Linux represents one path to the disk volume on the storage server
- Multipath devices are block devices in Linux
Multipath Device Using Native Linux Multipathing

Excludes edev...

```
bash-3.2# multipath -ll
mpath2 (360000970000192604545533031304435) dm-3 EMC,SYMMETRIX
[size=898M] [features=0] [hwhandler=0] [rw]
  round-robin 0 [prio=2][active]
    0:0:0:3 sdc  8:32 [active][ready]
    1:0:0:3 sdh  8:112 [active][ready]

mpath1 (360000970000192604545533031304434) dm-2 EMC,SYMMETRIX
[size=898M] [features=0] [hwhandler=0] [rw]
  round-robin 0 [prio=2][active]
    0:0:0:2 sdb  8:16 [active][ready]
    1:0:0:2 sdg  8:96 [active][ready]

......
```
Linux Notes

• There is no emulation overhead
• With SCSI - Linux handles IO and errors
• This is familiar to open systems admin’s
• Multiple IOs can be issued and outstanding
• NPIV can benefit performance but is primarily used for security reasons
• SCSI uses a customizable field for queuing
  – queue_depth
  – Can be set for each device
Linux Queue Depth

• For example:
  # lszfcp -l 0x0001000000000000 -a|grep queue_depth
  queue_depth = "32"
  queue_depth = "32"
  queue_depth = "32"
  queue_depth = "32"
  queue_depth = "32"  default
**Isluns**

- Isluns command - looks for all available LUNs by FCP port or host

```
lv192130:~ # lsluns
lsluns
Scanning for LUNs on adapter 0.0.2d03 at port 0x5000144260070901:
  0x0000000000000000
  0x0001000000000000
  0x0002000000000000
  0x0003000000000000
  ...................................
```

FCP Port

WWPN of storage

LUN

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Session 17494 Understanding the Benefits of SCSI in Linux on z Systems Environments
What is NPIV?

- N_port ID Virtualization allows many virtual WWPNs (FCP ports) to one physical WWPN of the CHPID
- Without NPIV all FCP ports on a CHPID have the same WWPN
- NPIV is becoming more popular
  - Non NPIV is still being used for their dev/test environments
- NPIV offers better security and easier administration of LUNs across FCP ports
How to Connect the NPIV Dots

- NPIV is enabled on the switch first
- NPIV is then enabled on the CHPID
- You can get a listing of each FCP port’s unique WWPN from the HMC
  - The base adapter retains its own original WWxN assigned by the manufacturer
- Each FCP port on the NPIV CHPID now has a unique virtual WWPN
- There is no requirement to manage a subset of LUNs at the Linux layer
- The HMC listing of the CHPID and its FCP ports will show you the virtual WWPNs for its ports
- You cannot tell by looking at the IOCDS if NPIV is enabled or not
- You should know if the FBA/SAN environment is using NPIV or not before you start debugging any issues
Query the FCP Devices

• From CP view all the FCP devices allocated to the Linux virtual machine
  
  # vmcp q fcp

  FCP  131F ON FCP   131F CHPID 84 SUBCHANNEL = 000F
  131F DEVTYPE FCP        CHPID 84 FCP
  131F QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
  WWPN C05076F1F000A09C

  FCP  141F ON FCP   141F CHPID 85 SUBCHANNEL = 0010
  141F DEVTYPE FCP        CHPID 85 FCP
  141F QDIO-ELIGIBLE      QIOASSIST-ELIGIBLE
  WWPN C05076F1F000A41C

• From Linux view the FCP devices (ports) allocated to the Linux instance
  
  # lszfcp
  0.0.131f host2
  0.0.141f host3
z/VM View of FCP

• `q chpid 84`
  Path 84 online to devices 1306 1310 1311 1312 1313 1314 1315 131A
  Path 84 online to devices 131B 131C 131D 131E 131F
  Ready; T=0.01/0.01 16:54:43
  (VARIED 1301 Online and attached it)

• `q 1301`
  FCP 1301 ATTACHED TO LINUX01 1301 CHPID 84
  WWPN C05076E4BD8050AC

• `q 1306`
  FCP 1306 FREE
  Ready; T=0.01/0.01 16:57:30
z/VM Directory Entry – FCP Devices

• Attach or dedicate (persistent across logoff/logon) FCP ports to Linux guest VM

• FCP ports may be allocated with a different virtual address than the real device address

USER LZ192139 CLASS 512M 1G G
INCLUDE LNXCLASS
FCP Ports for Linux Class
DEDICATE 1310 1310
DEDICATE 1410 1410
DEDICATE 1312 1312
DEDICATE 1412 1412
..........
FICON and FCP Mode

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Session 17494 Understanding the Benefits of SCSI in Linux on z Systems Environments

8/12/2015
DASD IO Stack

- **Host**
  - LVM and filesystem
  - Format, partition & vary on
    - dasdfmt
    - fdasd
    - chccwdev --online

- **z/VM**
  - add mdisk to a VM

- **CEC**
  - IOCDS – 3390’s

- **Storage Device**
  - Map CKD device
SCSI IO Stack

- **Host**
  - LVM and filesystem
  - Partition & vary on
    - `fdisk`
    - `zfcp_disk_configure`
    - `chccwdev --online`
- **z/VM**
  - Add FCP Ports to a VM
- **CEC**
  - IOCDS – FCP ports
- **SAN**
  - Zoning
- **Storage Device**
  - Map/Mask FBA device
SCSI Device Driver components

• There are several components that come together to execute SCSI IO
• Using the lsmod command you can see the relationship and other components that are needed in Linux

```
# lsmod|grep zfcp
Module                  Size  Used by
zfcp                   125380  32
scsi_transport_fc     71764   1  zfcp
qdio                   76842   3 qeth,13,zfcp,qeth
scsi_mod               303205  10
sg,sd_mod,zfcp,scsi_transport_fc,scsi_tgt,scsi_dh_alua,scsi_dh_hp_sw,scsi_dh_rdac,scsi_dh_emc,scsi_dh
```
SCSI Performance

• There is no emulation overhead
• With SCSI - Linux handles IO and errors
  – This is familiar to open systems admin’s
• Multiple IOs can be issued and outstanding
• NPIV can benefit performance but is primarily used for security reasons
• SCSI uses a customizable field for queuing
  – queue_depth
  – Can be set for each device

```
# lszfcp -l 0x0001000000000000 -a | grep queue_depth
queue_depth = "32"
queue_depth = "32"
queue_depth = 32
queue_depth = "32"
```

default
FBA as z/VM emulated devices (EDEV)

- Defined in z/VM as 9336 or FB-512 type device
- AKA EDEVs
- Emulation is used at the z/VM and Linux layer
- z/VM communicates to storage array with SCSI fibre channel protocol
- Can be setup as minidisk or direct attached device
- IO handled by Linux and z/VM
- Multipath support handled by z/VM
- Storage can be managed and monitored from z/VM
- Commonly used for Linux OS
FBA as z/VM edev for Paging

• *May be used for z/VM paging devices*
• *Please see IBM z/VM 6.3 Resource Overcommitment paper at:

“The Large Memory Support and the HiperDispatch features introduced with z/VM 6.3 significantly improved the resource overcommitment behavior, as opposed to z/VM 6.2. In addition, the use of EDEV-SCSI devices for z/VM paging allowed substantially higher memory overcommitment levels when compared to using ECKD paging devices. z/VM 6.3 with EDEV-SCSI paging devices can be highly recommended for environments running at high memory overcommitment levels.”
q edev d000 details
EDEV D000 TYPE FBA ATTRIBUTES SCSI
  VENDOR: EMC PRODUCT: Invista REVISION: 5400
  BLOCKSIZE: 512 NUMBER OF BLOCKS: 33555840
PATHS:
  FCP_DEV: 2D03 WWPN: 5000144260070901 LUN: 000D000000000000
    CONNECTION TYPE: SWITCHED STATUS: ONLINE
  FCP_DEV: 2D23 WWPN: 5000144270070901 LUN: 000D000000000000
    CONNECTION TYPE: SWITCHED STATUS: ONLINE
  FCP_DEV: 100C WWPN: 5000144260061101 LUN: 000D000000000000
    CONNECTION TYPE: SWITCHED STATUS: ONLINE
  FCP_DEV: 110C WWPN: 5000144270061101 LUN: 000D000000000000
    CONNECTION TYPE: SWITCHED STATUS: ONLINE
EQID: 6000144000000010F007092A6B3D4AF6F7000000000200057F

WWPN of Storage

FCP Ports
ENVIRONMENT/PLATFORM BENEFITS

Mainframe
Reliability
Availability
Serviceability

Open Systems
Open source
Worldwide innovation & collaboration
Adoption by a community of experts

SCSI continues to evolve…
Flexibility: Best of Both Worlds

- **z/VM**
  - Mature virtualization
  - Removes physical limitations dynamically

- **Linux**
  - Enterprise OS based on UNIX standards
  - Innovative
  - Open source Community driven

- **Linux on z/VM - Best of both worlds**
  - Enables throughput benefits for Linux guest images
  - Enhances overall system performance and scalability
SCSI Innovation

• New host based SCSI commands for thin device cleanup
  – SCSI standard (t10.org) - T10 Technical Committee on SCSI Storage Interfaces
  – SCSI unmap
    • SCSI write same with unmap
  – Support for these SCSI commands are
    • Kernel dependent – Linux vendor and release
    • Storage array dependent
Flexibility

- Ability to exploit open systems solutions
  - Storage virtualization appliances
    - EMC VPLEX, IBM SVC
  - Virtual provisioning or Thin provisioning

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Session 17494 Understanding the Benefits of SCSI in Linux on z Systems Environments
Storage Optimization

- Virtual Provisioning (VP) simplifies Storage Management for FBA
  - Removes data placement requirements from administrators
  - Introduces thin devices
  - Allows for over subscription of storage
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
  – Use what is needed
SCSI Cleanup for Linux on z Systems

- 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
Benefits – Why FCP & SCSI

- Performance advantages
  - SCSI continues to evolve in performance
  - Reason 1: asynchronous I/O
  - Reason 2: no emulation overhead
- User definable FBA disk up to 2TB (today)
- Up to 15 partitions (16 minor numbers per device)
- FBA as SCSI LUNs maximize disk space
  - no low-level formatting
- z Systems integration in existing FC SANs
- Use of existing FICON infrastructure
  - FICON Express adapter cards
  - FC switches / Cabling
  - Storage subsystems
- Dynamic configuration
  - Adding of new LUNs is possible without IOCDS change
Summary

• FBA has best use of physical device space
• Talk to your Storage Admins. They can help demystify this
• SCSI is an industry standard
• SCSI LUNs
  – Can be provisioned rapidly, enabling cloud deployment
  – Is favored for performance
  – Solution innovations
Questions?

Johnathan Crossno
VMAX Principal Product Manager
z/VM and Linux on System z

johnathan.crossno@emc.com

EMC Corporation 176 South Street, Hopkinton, Massachusetts 01748-9103 www.emc.com