Benjamin Block – Linux on Z Development
2018-05-09

FCP with Linux on Z and LinuxONE: SCSI over Fibre Channel Best Practices

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Because of the large number of products marketed by IBM, IBM's practice is to list only the most important of its common law marks. Failure of a mark to appear on this page does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies.

A current list of IBM trademarks is available on the Web at: http://www.ibm.com/legal/copytrade.shtml.

Those trademarks followed by ® are registered trademarks of IBM in the United States; those followed by ™ are trademarks or common law marks of IBM in the United States.

*, AIX®, AS/400e™, Db2®, DB2®, developerWorks®, DS8000®, ECKD™, eServer™, FICON®, GPFS™, HiperSockets™, HyperSwap®, IBM®, IBM (logo)®, IBM FlashSystem®, IBM LinuxONE™, IBM LinuxONE Emperor™, IBM LinuxONE Emperor II™, IBM LinuxONE Rockhopper™, IBM Spectrum™, IBM Spectrum Control™, IBM Spectrum Storage™, IBM Z®, IBM z Systems®, IBM z13®, IBM z13s®, IBM z14™, ibm.com®, Linear Tape File System™, OS/390®, PR/SM™, RS/6000®, Redbooks®, Redpaper™, Redpapers™, S390-Tools®, S/390®, Storwize®, Storwize (logo)®, System Storage®, System Storage DS®, System z®, System z9®, System z10®, System/390®, Tivoli®, Tivoli (logo)®, XIV®, z Systems®, z9®, z10™, z13®, z13s®, z/Architecture®, z/OS®, z/VM®, z/VSE®, zEnterprise®

* All other products may be trademarks or registered trademarks of their respective companies.

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are registered trademarks of Hewlett Packard Enterprise, International Business Machines Corporation and Quantum Corporation in the United States and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

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

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

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- Introduction and Terminology
- Setup
  - I/O Configuration
  - Multipathing
  - LUN Management with ZFCP
- IPL (booting) over FCP

Introduction and Terminology

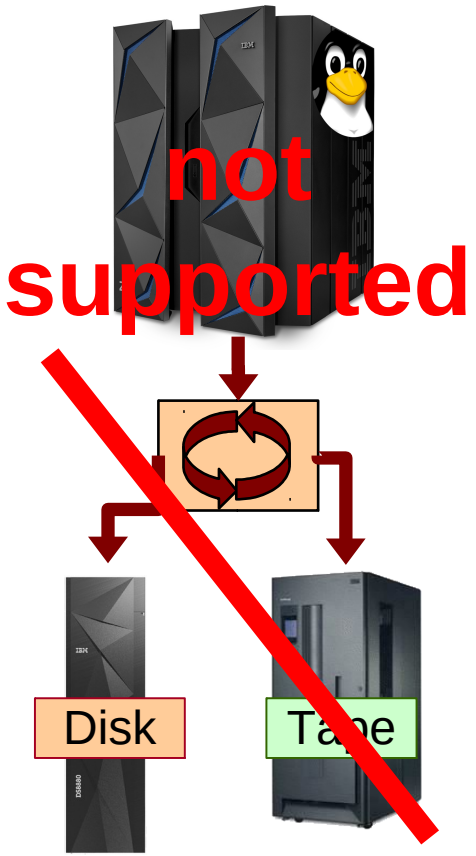
FCP in a nutshell

- Storage Area Networks (SANs) are specialized networks dedicated to the transport of mass storage data (block/object oriented)
- Today the most common SAN technology used is Fibre Channel (FC) [T11]
- The Fibre Channel standard was developed by the InterNational Committee for Information Technology Standards (INCITS)
- Over this FC transport, using the Fibre Channel Protocol (FCP) as encapsulation, the SCSI protocol is used to address and transfer raw data between server and storage device [T10]
- Each server and storage is equipped with a least two adapters which provide a redundant physical connection to a redundant SAN
- For Z or LinuxONE, any supported FCP adapter, such as FICON Express, can be used for this purpose.
  - Latest adapter card is: FICON Express16S+

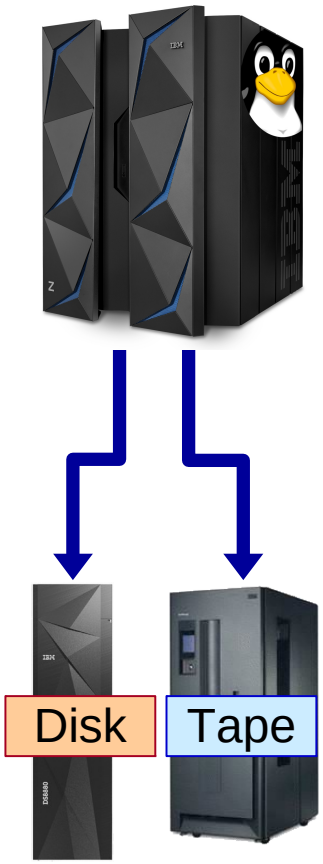
Throughout presentation, all royal blue text fragments are clickable hyperlinks!

SAN Topologies and IBM Z / LinuxONE

direct attached arbitrated loop [T11 FC-AL]

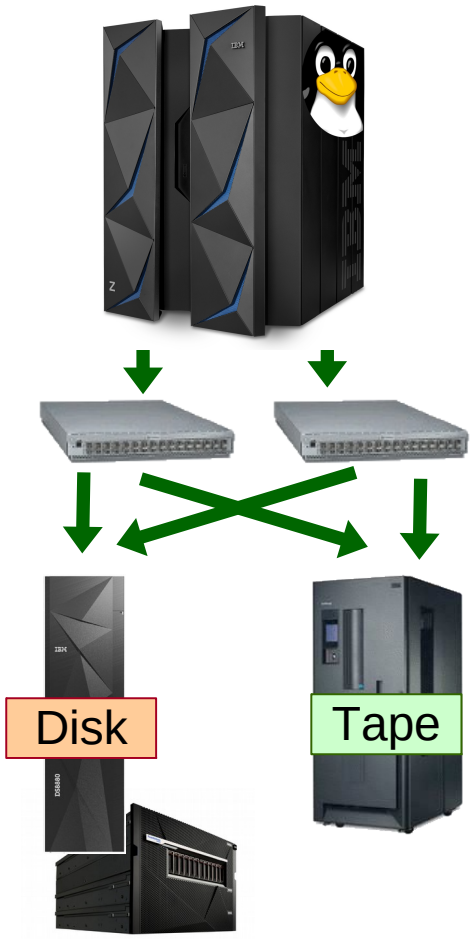


point-to-point

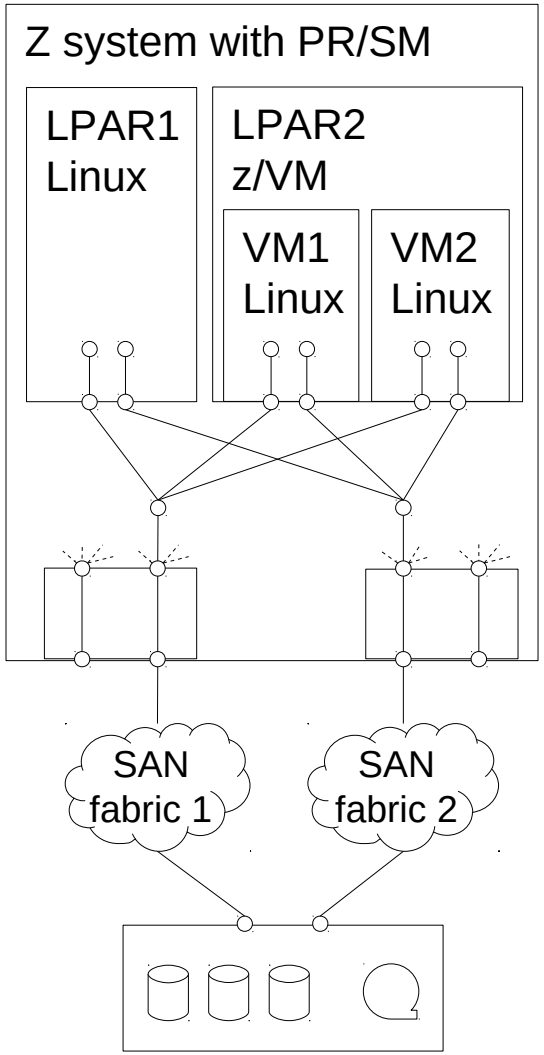


focus

switched fabric [T11 FC-SW]



FCP with IBM Z / LinuxONE



hypervisors / virtual machines

devno

FCP device in Linux: /sys/devices/css0/0.3.001f/0.3.5a00

subchannel set

subchannel bus-ID

device bus-ID

FCP devices (direct-attached) / virtual HBAs

FCP subchannels

I/O Configur. (IOCDS)

CHPIDs (1 per PCHID if spanned), type FCP

PCHIDs / FCP Channels / HBAs (2 per card)

FICON Express16S+ cards

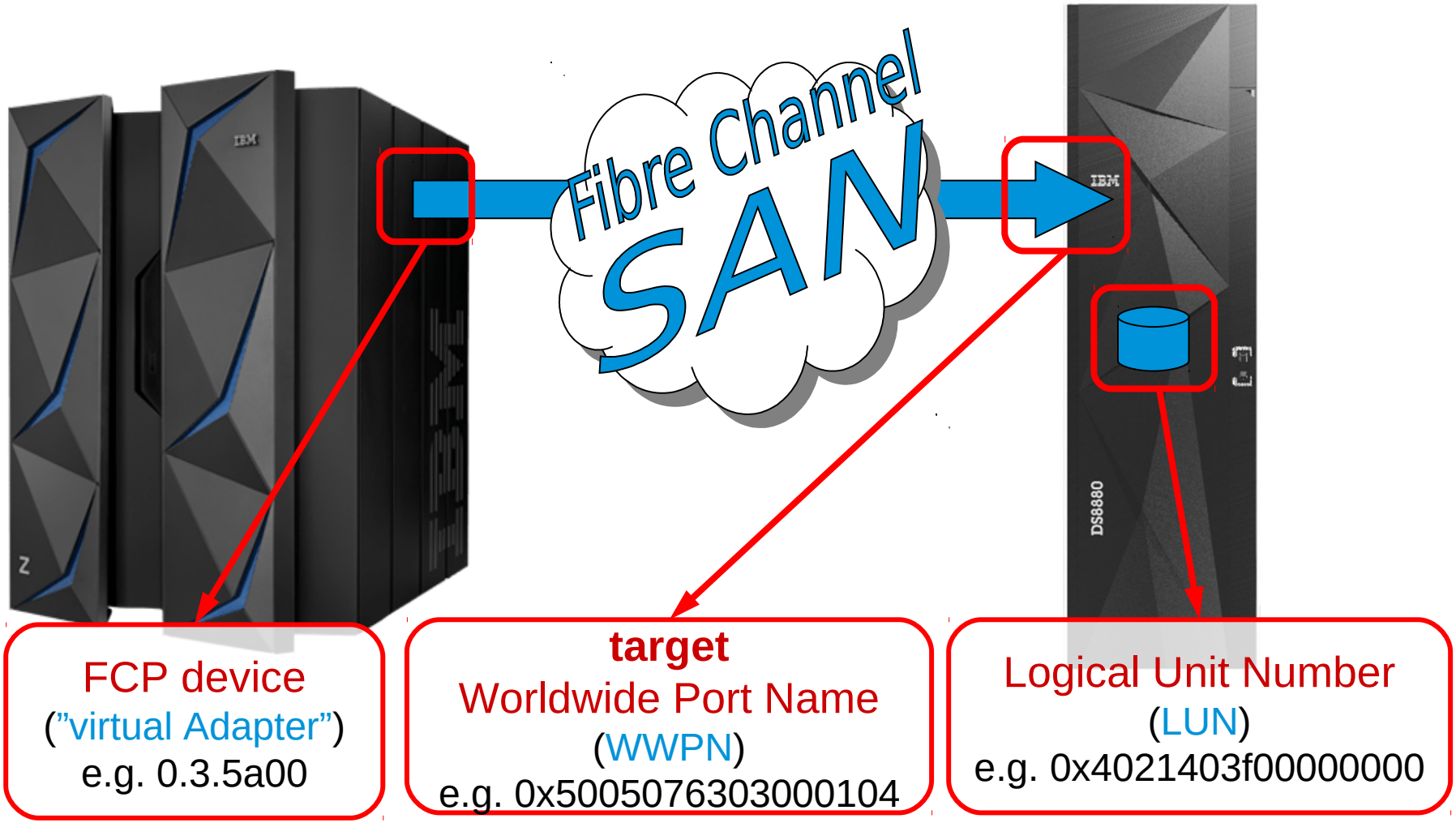
initiator physical fibre ports (2 per card)

paths over physically redundant FC fabrics

target physical fibre ports

storage target

SAN Addressing for One (of Multiple) Paths



Linux kernel parameters and zipl target

- kernel parameters
  - RHEL, SLES≤11, Ubuntu [doc]
    1. edit variable 'parameters' in /etc/zipl.conf
    2. if changes affect root file system dependencies, run on
       RHEL≤5, SLES≤11: mkinitrd ; RHEL≥6: dracut -f ; Ubuntu: update-initramfs -u
    3. run [optional with Ubuntu, otherwise mandatory]: /sbin/zipl
  - SLES12 [doc]: no zipl.conf ; use 'yast bootloader' or:
    1. edit variable 'GRUB_CMDLINE_LINUX_DEFAULT' in /etc/default/grub
    2. if changes affect root file system dependencies, run: dracut -f && grub2-install
    3. run: grub2-mkconfig -o /boot/grub2/grub.cfg
  - For dynamic mechanism, see
    slide SCSI IPL Dynamically Pass Kernel Parameters
  - RHEL, SLES≤11, Ubuntu [doc]
    /boot/ manually maintained by administrator
  - SLES12 [doc]
    /boot/zipl/ automatically maintained (do not touch) by grub2 toolchain

Setup

Setup Overview for FCP with Linux on Z and LinuxONE

1) Optionally: Early Preparation.
2) Define FCP devices within the mainframe (I/O Configuration), dedicate in z/VM.
3) Enable NPIV for the FCP devices (Service Element / HMC).
4) Configure zoning for the FCP devices to gain access to desired target ports within a SAN, max. one single initiator (virtual) WWPN per zone.
5) Configure LUN masking for the FCP devices at the target device to gain access to desired LUNs.
6) In Linux, configure multipathing
7) In Linux, configure target WWPNs and LUNs to obtain SCSI devices.

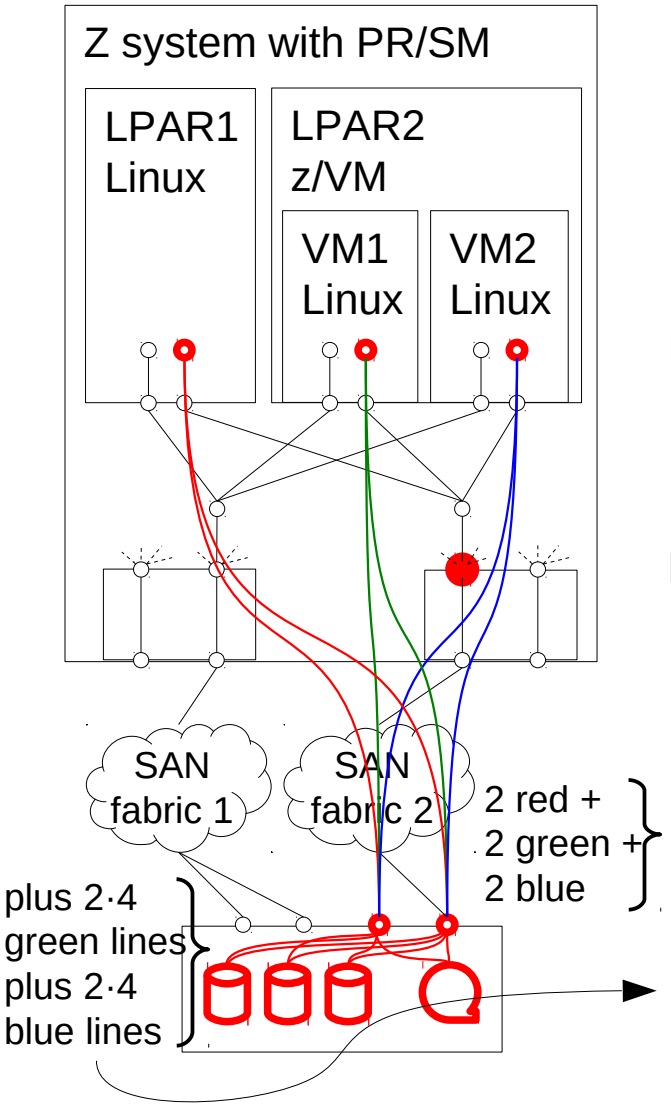
Note I: If FCP Channel is directly connected to a target device (point-to-point), steps 3 & 4 do not apply. After preparation, steps 4 & 5 can be conducted before or in parallel to step 3.

Note II: For steps 2, 3, 4 and 5 there are additional slides in the ‘Additional Slides’ part of the presentation.

I/O Configuration for FCP Devices

- for LPAR hypervisor (PR/SM): use Dynamic Partition Manager (DPM) [doc][z13 GA2], or explicit virtual device config & passthrough in IOCDS
- for z/VM: dedicate 1 NPIV FCP device per CHPID per z/VM guest in its user dir.
- for KVM on IBM Z: on host, use 1 NPIV FCP device per CHPID per KVM guest
- Use N_Port ID Virtualization (NPIV) whenever possible
- We recommend the use of strict Single Initiator Zoning in the SAN

Z / LinuxONE Hardware for FCP: Limits per Channel (PCHID)



assuming one online FCP device per VM per PCHID
V: # of VMs per PCHID
P: # of target ports per NPIV-enabled FCP device
L: # of LUNs per target port
assuming equal distribution of resources:
V ≤ 64(32) && V·(P+1) ≤ 1000(500) && V·P·L ≤ 8192(4096)

FCP devices (direct-attached) / virtual HBAs:
≤64(32) online NPIV-enabled FCP devices per PCHID → Linux
≤255 defined FCP devices per LPAR per CHPID → IOCDS
≤480 defined FCP devices per CHPID → IOCDS

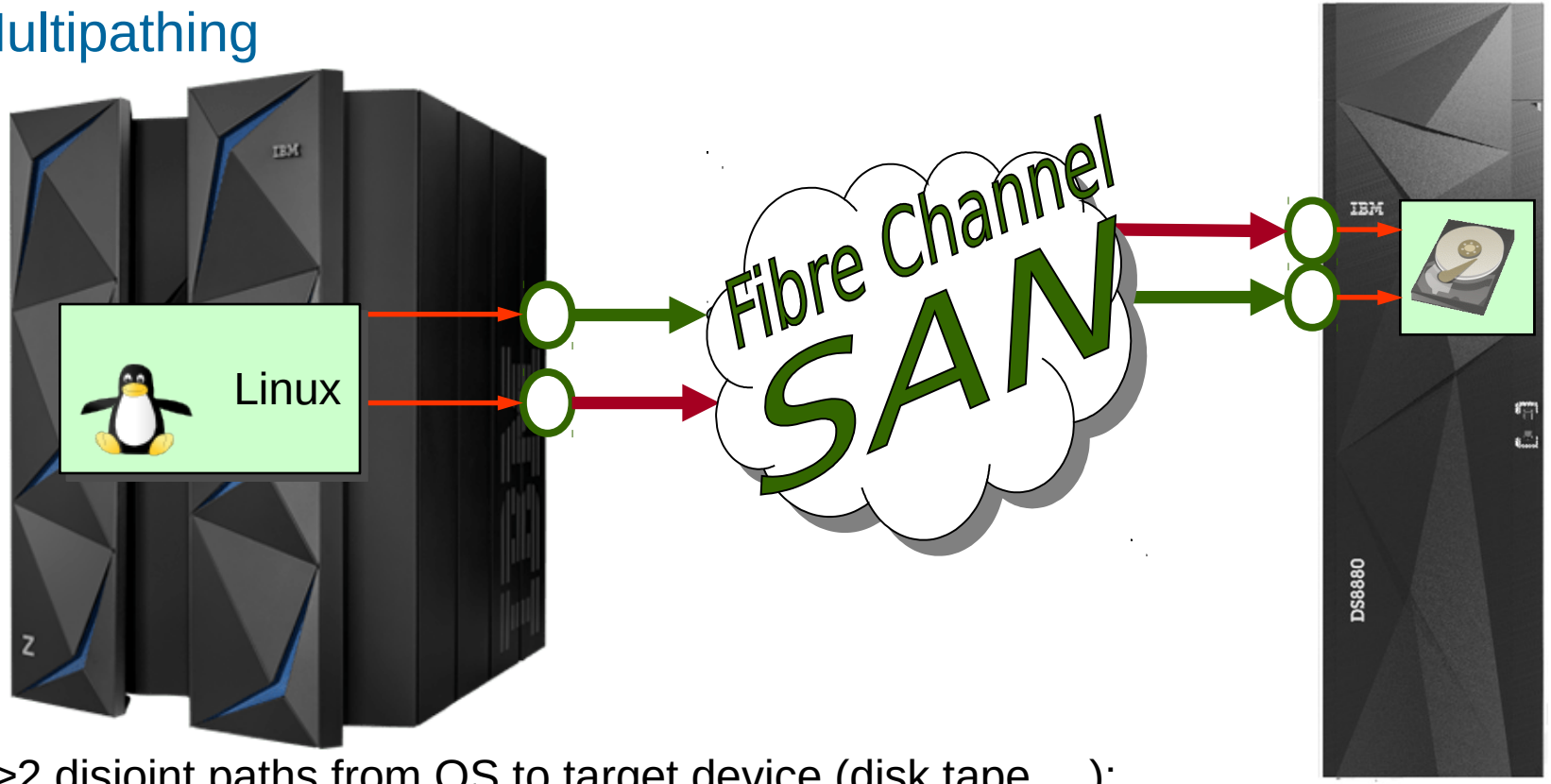
PCHID / FCP Channel / HBA

(dark magenta values in parentheses are old limits with adapters older than FICON Express16S & 16S+)

≤1000(500) open target ports per PCHID → zoning
account for 1 zfcp-internal nameserver port per FCP device!

≤8192(4096) attached LUNs per PCHID
→ LUN masking & zoning

Multipathing



- ≥2 disjoint paths from OS to target device (disk,tape,…); independent FCP cards, independent switches, and independent target ports.
  - Redundancy: Avoid single points of failure
  - Performance: I/O requests can be spread across multiple paths
  - Serviceability: When component of one path is in maintenance mode I/O continues to run through other path(s)
- Linux does multipathing differently for disks and tapes …

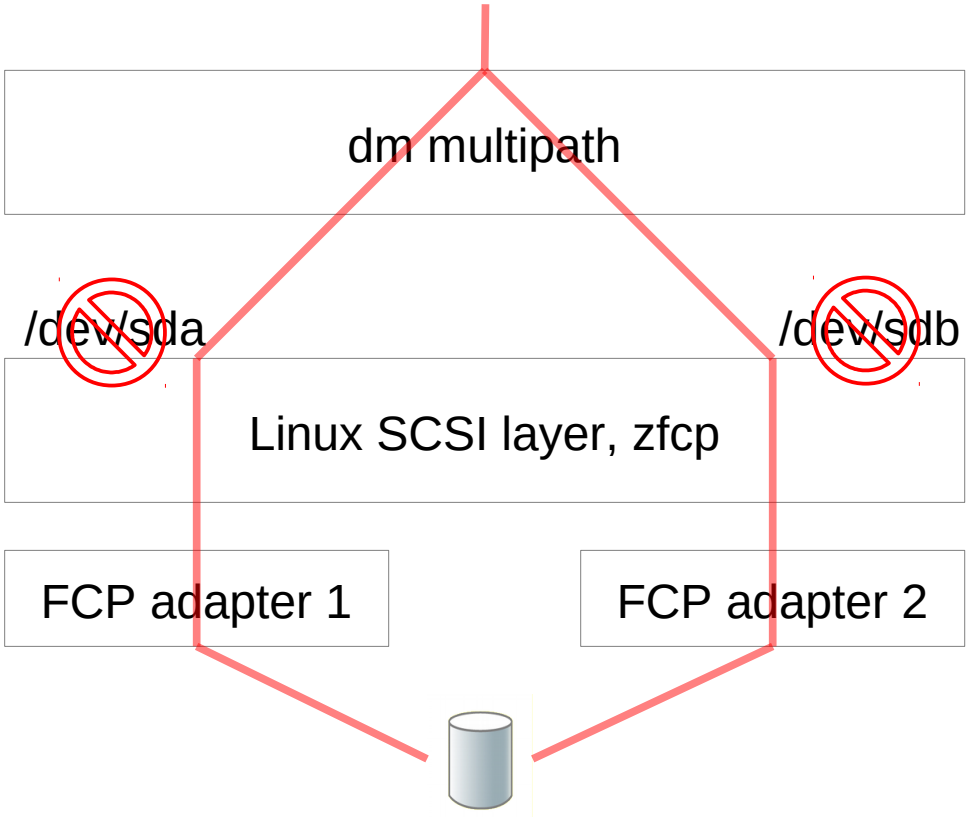
Multipathing for Disks – Persistent Configuration

- **Use multipathing on installation** for all disks incl. root-fs and zipl target:
  SLES,RHEL≥6,KVM,Ubuntu
  Lifting single path to multipath is difficult [≥S10,≥R6,U].
  - zipl target: use multipathing with sep. mountpoint, or place inside root-fs [S10.4,S11.1,R6,K,U],
    if stacking devices on top of multipathing see zipl_helper.device-mapper docs
  - Root-fs (/): always multipathing (optionally stack devices on top)
  - any other mountpoint or direct access block device: always multipathing
    (optionally any other virtual block devices such as LVM on top)
- **Post installation [SLES, RHEL, KVM, Ubuntu]:**
  - **ensure /etc/multipath.conf is suitable (esp. blacklist)**
  - **ensure multipathd is enabled and running (re-activates failed paths)**
    (NOTE: option rr_min_io is called rr_min_io_rq in more recent distros)

Multipathing for Disks – device-mapper multipath devices

- device-mapper multipath target in kernel creates one block device per disk:
  /dev/mapper/36005076303ffc5620000000000000010cc

unique WWID;
“user_friendly_names no”
In /etc/multipath.conf



- World-Wide Identifier (not LUN!) from storage server identifies volume / disk / path group
- each SCSI device represents a single path to a target device, do ***not*** use these devices directly!

Multipathing for Disks – device-mapper multipath devices (cont.)

- Multipath devices are created automatically when SCSI LUNs are attached

WWID for volume

pathgroup

```
# multipathd -k'show topo'
36005076303ffc56200000000000002006 dm-0 IBM     ,2107900
size=5.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 0:0:2:1074151456 sda   8:0   active ready running
  `- 1:0:5:1074151456 sdb   8:16  active ready running
```

- Multipath devices are virtual block devices, can be used as container for, e.g.
  - Partitions
  - Logical Volume Manager (LVM): more details
  - Other Device-Mapper Targets: e.g. DM-Crypt for encryption of data in flight and at rest
  - Directly for a file system or as raw block device (e.g. for RDBMS)
- Device to work with: e.g. /dev/mapper/36005076303ffc56200000000000002006 (or user-friendly / alias multipath names such as /dev/mapper/mpatha if enabled)

```
# mkfs.ext4 /dev/mapper/36005076303ffc56200000000000002006
# mount /dev/mapper/36005076303ffc56200000000000002006 /mnt
```

Multipathing – Error Recovery on FC Transport Layer

- on zfcp detecting broken target port (cable pull, switch maint., target logged out): tell FC transport class, which starts **fast_io_fail_tmo** & **dev_loss_tmo** for rport
- on **fast_io_fail_tmo**: zfcp port recovery returns pending IO with result DID_TRANSPORT_FAILFAST
- (on **dev_loss_tmo**: zfcp port recovery returns pending IO with result DID_NO_CONNECT **and** FC transport deletes SCSI target with its SCSI devices) ⇦ issues under IO ⇨ disable **dev_loss_tmo** and enable **fast_io_fail_tmo** (5 seconds):
  - for disks: “infinity” or “2147483647” for dev_loss_tmo in /etc/multipath.conf [RHEL,SLES,Ubuntu,KVM,MULTIPATH.CONF(5)]
  - for devices not handled by dm_multipath/multipath-tools (e.g.: tapes, libraries, …): refer to the respective manual, or if not specified, stay with the default-values
  - double check with “lszfcp -Pa”
- path failover: kernel dm_multipath can re-queue returned IO on another path

Multipathing – Handling on Losing Last Path

- if all paths gone at the same time (even for a split second),
  return I/O error (clusters and/or cluster applications) or queue I/O (other):
  disks: /etc/multipath.conf: 'no_path_retry queue' (alias feature queue_if_no_path);
  multipath.conf settings can contradict → double check if queueing is active:

```
# multipathd -k'list maps status'
name                                failback  queueing paths dm-st  write_prot
36005076802870052a000000000000318 immediate on       2     active rw
36005076304ffc3e80000000000002000 -         on       2     active rw
```

  otherwise data corruption can occur if applications don't handle I/O errors correctly
- above setting **required** for z/VM SSI live guest relocation (LGR)
  with dedicated FCP devices [z/VM docs1, docs2]

- if I/O is stuck due to queueing and paths won't return but you want to flush I/O:
  # dmsetup message <*mapname*> 0 fail_if_no_path [SLES,RHEL,KVM,Ubuntu]

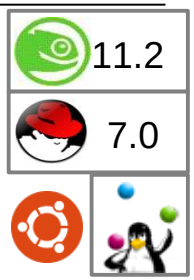
- do not queue for: Linux software storage site mirroring for disaster recovery

LUN Management with ZFCP: 2 Methods

1) automatic LUN scanning (new and only with NPIV-enabled FCP devices)
   - user specifies to only set FCP device online
   - zfcp attaches all paths visible through fabric zoning and target LUN masking
2) explicit manual LUN whitelist (traditional)
   - user specifies every single path using <FCP device,WWPN,FCP LUN>
   - zfcp only attaches these paths

- to ignore certain LUNs: disable automatic LUN scanning with kernel parameter "zfcp.allow_lun_scan=0", and then (before any reboot!) use explicit manual LUN whitelists for all FCP devices in such Linux instance
- do not mix up automatic LUN scanning (new) with automatic port scanning (no more “port_add”, since R6.0,S11SP1,KVM,Ubuntu)
- do not use zfcp sysfs interface nor cio_ignore directly, e.g. with own scripting; use tested & supported distribution mechanisms…

LUN Management with ZFCP: Automatic LUN Scanning for NPIV-enabled FCP Devices

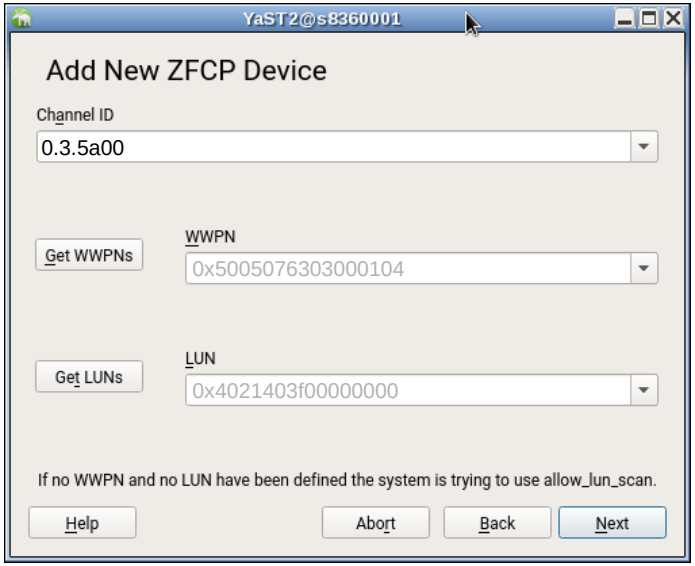


- With this feature, NPIV-enabled FCP devices attach LUNs automatically.
- Needs zoning and LUN masking per each FCP dev. to only access desired LUNs.
- Automatic LUN scanning is enabled by default,
  except for SLES11 which requires the kernel parameter "zfcp.allow_lun_scan=1"
- to manually trigger a LUN discovery:
  `# rescan-scsi-bus.sh -a`
- then check with “lszfcp -D” or with “lsscsi -vtxx”

```
# lszfcp -D
0.0.1700/0x500507630503c1ae/0x4022400000000000 0:0:12:1073758242
0.0.1700/0x500507630503c1ae/0x4022401000000000 0:0:12:1073883778
0.0.1700/0x500507630503c1ae/0x4022402000000000 0:0:12:1073889314
```

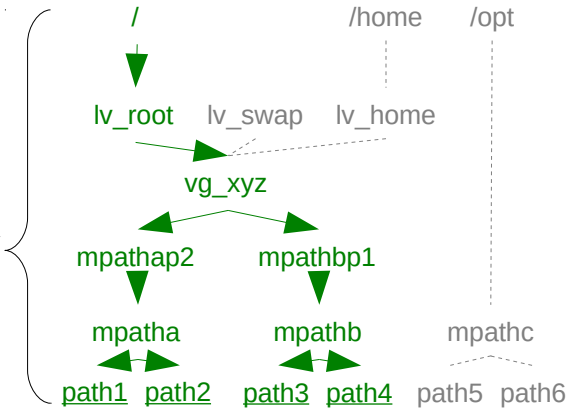
- there are no sysfs directories in the zfcp branch for automatically attached LUNs!
  /sys/bus/ccw/drivers/zfcp/<*FCP device bus-ID*>/0x<*WWPN*>/~~0x<*FCP LUN*>~~

LUN Management with ZFCP: SLES Post-Installation



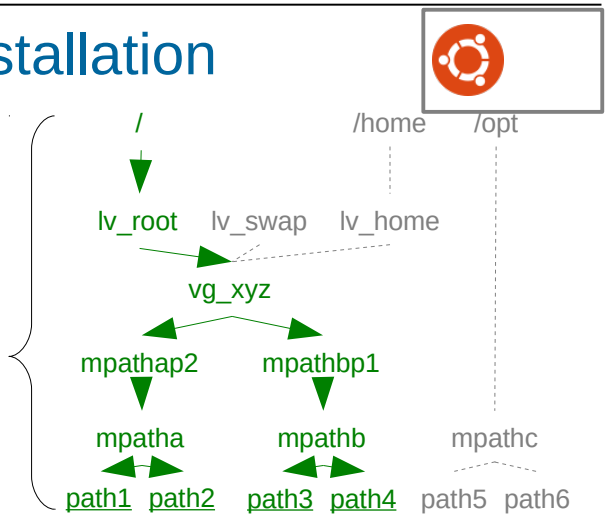
- Notes for steps during installation
- GUI: yast2 cio [SLES12] && yast2 zfcp
- TUI: yast cio [SLES12] && yast zfcp
- command line:
  - enable/disable FCP device:
    zfcp_host_configure 0.3.5a00 1/0
  - optionally discover WWPNs or LUNs manually:
    zfcp_san_disc
  - attach/detach FCP LUN to/from enabled FCP device:
    zfcp_disk_configure 0.3.5a00 0x5005076303000104 0x4021403f00000000 1/0
- GUI and TUI can discover available
  FCP devices, WWPNs, and LUNs
- if changes affect root-fs dependencies, process changes:
  SLES≤11: mkinitrd && zipl . SLES12: dracut -f && grub2-install
- auto LUN scan <SLES12SP2: only use zfcp_host_configure, nothing else.
  auto LUN scan ≥SLES12SP2: yast omitting WWPN&LUN, or zfcp_host_configure.

LUN Management with ZFCP: RHEL Post-Installation



- GUI only available during installation.
- SCSI disk paths (indirectly) required to mount root-fs, e.g. each path of all multipath PVs of a VG with root-LV
  - RHEL5: /etc/zfcp.conf (see below)
  - RHEL6: /etc/zipl.conf:
    … rd_ZFCP=0.3.5a00,0x5005076303000104,0x4021403f00000000 rd_ZFCP=…
  - RHEL7: /etc/zipl.conf:
    … rd.zfcp=0.3.5a00,0x5005076303000104,0x4021403f00000000 rd.zfcp=…
  - process changes: (mkinitrd … [RHEL5] or dracut -f [RHEL≥6]) && zipl
- any other SCSI devices such as data volumes or tapes,
  RHEL6: incl. all LUNs for kdump target even if on root-fs, rd_ZFCP not sufficient!
  - RHEL5/6/7: /etc/zfcp.conf:
    …
    0.3.5a00 0x5005076303000104 0x4021403f00000000
  - activate additions to /etc/zfcp.conf: zfcp_cio_free [R≥6] && zfcpconf.sh
- optionally discover LUNs manually: lsluns [prep: “modprobe sg” [RH1076689]]
- temp. workaround for auto LUN scan: specify just one valid path per FCP device

LUN Management with ZFCP: Ubuntu Post-Installation



- Notes for steps during installtion
- command line
  - auto LUN scan
    - active:
      online FCP device: chzdev zfcp-host 0.3.5a00 -e
      (attention, offline FCP device stops all its LUNs!: chzdev zfcp-host 0.3.5a00 -d)
    - Inactive:
      attach LUN: chzdev zfcp-lun 0.3.5a00:0x5005076303000104:0x4021403f00000000 -e
      remove LUN: chzdev zfcp-lun 0.3.5a00:0x5005076303000104:0x4021403f00000000 -d
  - if changes affect SCSI disk paths (indirectly) required to mount root-fs,
    e.g. each path of all multipath PVs of a VG with root-LV,
    process changes: update-initramfs -u [includes necessary zipl run!]
  - optionally discover LUNs manually: lsluns

IPL (booting) over FCP

SCSI IPL

- SCSI IPL expands the set of IPL'able devices
  - SCSI disk to boot Linux ("zipl target", /boot/(zipl/) mountpoint or inside root-fs)
  - SCSI disk for standalone zfcpdump (hypervisor-assisted system dumper)
- New set of IPL parameters
  - Required: address SCSI disk, pick **one** available path to zipl target / zfcpdump:
    - FCP device number
    - target WWPN
    - LUN
  - Select zipl boot menu entry with "bootprog", no interactive menu as with DASD
  - Pass kernel parameters with "OS specific load parms"/"scpdata" [≥S11SP1,R6,K,U]
  - Select grub2 boot menu entry with "loadparm" [SLES12+grub2-2.02~beta2-***54.1***]
- LPAR and z/VM guests supported
- SCSI (IPL) with z/VM Version 4.4 (with PTF UM30989) or newer

Summary of FCP

- available for IBM Z including zSeries and System z, and for LinuxONE
- based on existing Fibre Channel infrastructure
- integrates IBM Z / LinuxONE into standard SANs
- connects to switched fabric or point-to-point
- runs on all available z/VM and RHEL / SLES / KVM / Ubuntu versions
- multipathing for SCSI disks & tapes is a must
- gives you new storage device choices
- buys you flexibility at the cost of complexity
- tooling available, receiving better integration

More Information: IBM

- I/O Connectivity on IBM Z mainframe servers
  http://www.ibm.com/systems/z/connectivity/
- Supported Storage: IBM System Storage Interoperation Center
  http://www.ibm.com/systems/support/storage/ssic/
- Linux on Z and LinuxONE documentation by IBM
  http://www.ibm.com/developerworks/linux/linux390/distribution_hints.html, or
  http://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_distlibs.html
  - Device Drivers, Features, and Commands
  - Using the Dump Tools
  - Kernel Messages
  - How to use FC-attached SCSI devices with Linux on Z
- IBM Redbooks
  - Fibre Channel Protocol for Linux and z/VM on IBM System z
    http://www.redbooks.ibm.com/abstracts/sg247266.html
- KVM running on IBM Z:
  https://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_kvm_base.html

More Information: Linux Distribution Partners

- Red Hat Enterprise Linux 7:
  - Release Notes https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/7.1_Release_Notes/index.html
  - Installation Guide
    https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/chap-installer-booting-ipl-s390.html#sect-custom...
    https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-storage-devices-s390.html
    https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-kickstart-syntax.html#idp16814248
    https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-post-installation-fcp-attached-luns-s390.html#
  - DM Multipath https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/DM_Multipath/index.html
  - Storage Administration Guide
    https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Storage_Administration_Guide/index.html
- SUSE Linux Enterprise Server 12:
  - Release Notes https://www.suse.com/releasenotes/x86_64/SUSE-SLES/12/#InfraPackArch.SystemZ
  - Deployment Guide https://www.suse.com/documentation/sles-12/book_sle_deployment/data/sec_i_yast2_s390_part.html
  - Administration Guide https://www.suse.com/documentation/sles-12/book_sle_admin/data/sec_zseries_rescue.html
  - Storage Administration Guide https://www.suse.com/documentation/sles-12/stor_admin/data/stor_admin.html
  - AutoYAST for unattended installation https://www.suse.com/documentation/sles-12/book_autoyast/data/createprofile_partitioning.html
- Canonical Ubuntu 18.04 LTS Server Edition
  - Release Notes https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes
  - Ubuntu for IBM Z and LinuxONE https://www.ubuntu.com/download/server/s390x , https://wiki.ubuntu.com/S390X
  - Installation Guide https://help.ubuntu.com/lts/installation-guide/s390x/index.html
    updated temporary Installation Guide incl. Z-specific preseeding information https://wiki.ubuntu.com/S390X/InstallationGuide
    FCP device management for the Debian Installer https://anonscm.debian.org/cgit/d-i/s390-zfcp.git/tree/README
  - Server Guide https://help.ubuntu.com/lts/serverguide/index.html

Questions?

Benjamin Block

Linux on Z Development

Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-1632
bblock@de.ibm.com

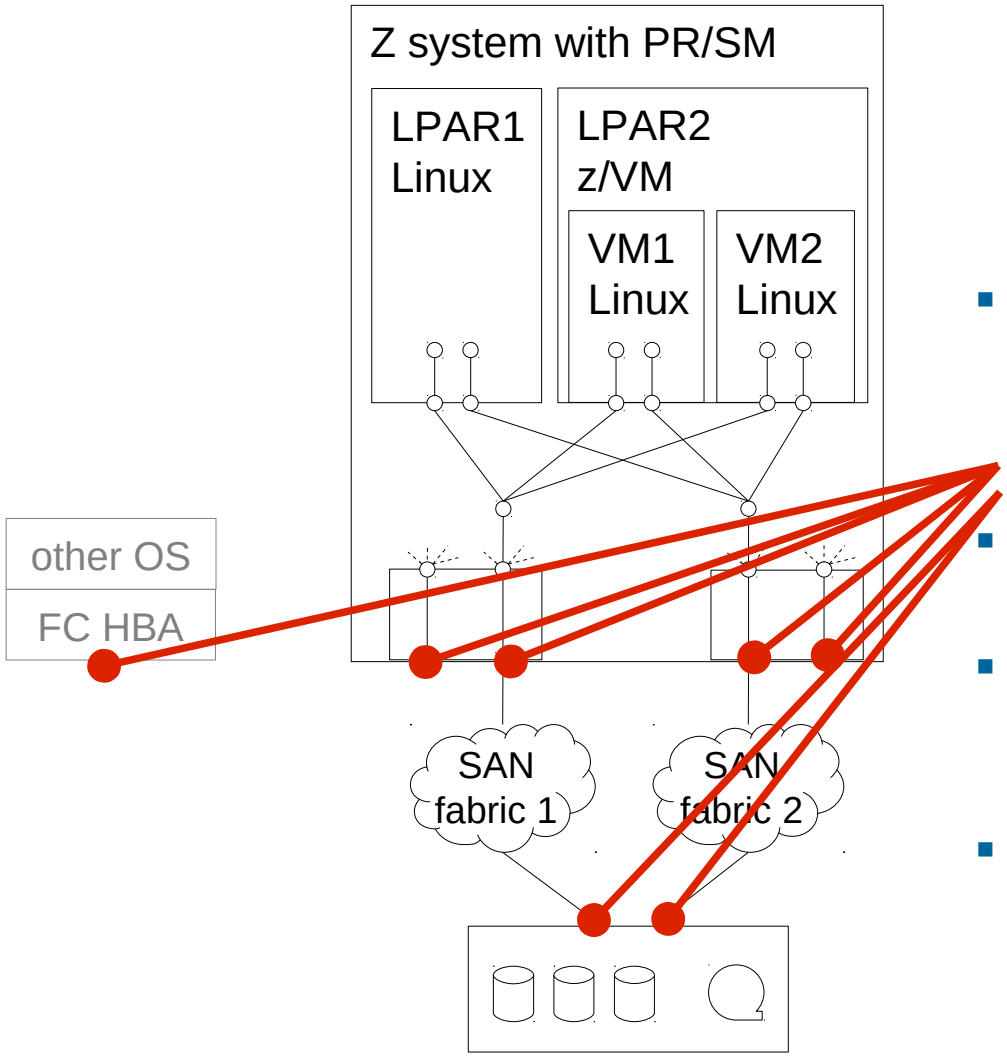
Additional Slides …

Introduction and Terminology

FCP Compared to Channel I/O

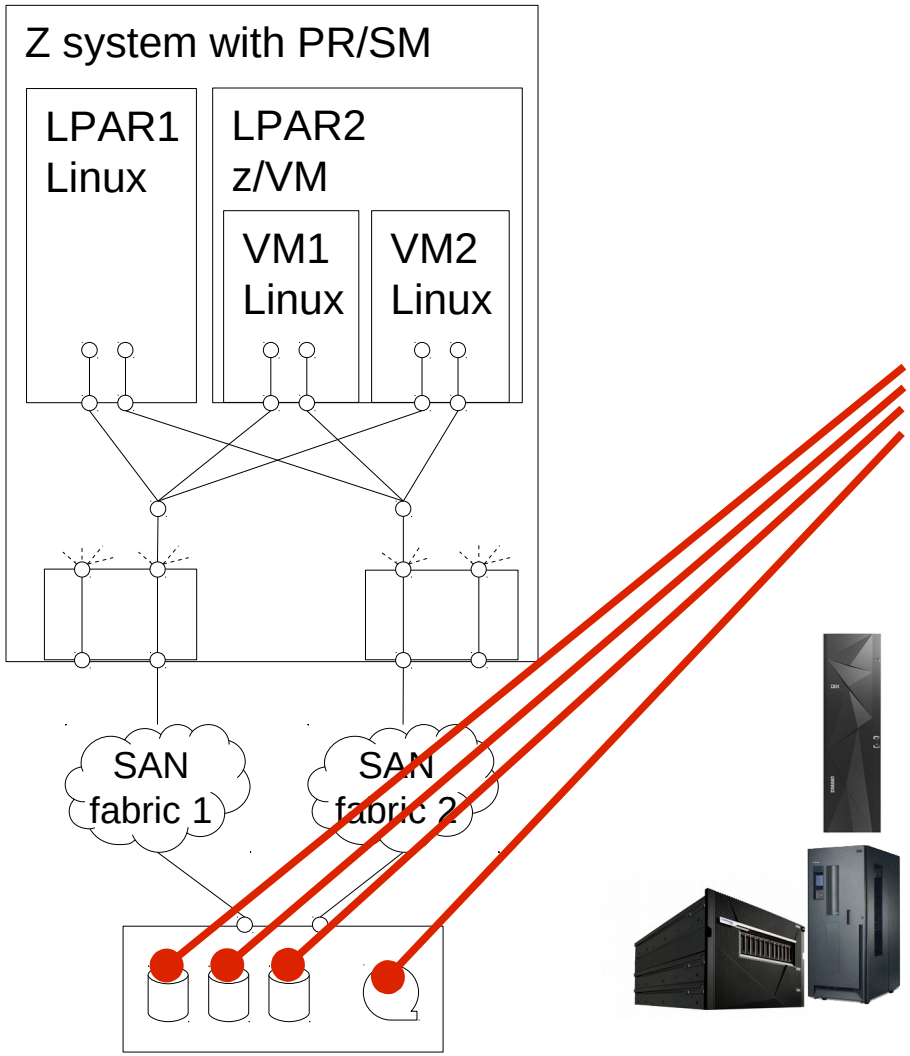
	FCP	Channel I/O
OS	**multipathing** handled in operating systems	multipathing handled in IBM Z firmware
	port and LUN attachment handled in operating systems	port attachment handled in IBM Z I/O configuration
fabric	FCP device represents **virtual adapter** to the Fibre Channel SAN	DASD device represents **disk** volume (ECKD)
	FCP device defined in IBM Z I/O config. ⇨ add new storage without IOCDS change	**disk** defined in IBM Z I/O configuration
	both use **existing FC SAN**: FICON Express cards, switches, cabling, storage subsystems	
	additional **configuration** beyond IBM Z: • Zoning in the SAN fabric switches • LUN masking on the storage server	Switch configuration via IBM Z I/O configuration
disk	no restrictions for SCSI disk size	**disk size** restrictions to Mod 54 / Mod A
	0–15 **partitions** per disk	1–3 partitions per disk
	no low-level formatting	**low-level formatting** ⇨ wastes disk space
	no emulation ⇨ performance	ECKD **emulation** overhead
	built-in **asynchronous I/O** ⇨ performance	async I/O requires Parallel Access Volumes

Worldwide Port Names (WWPNs)



- Servers (initiators) and storage devices (targets) attach through Fibre Channel ports (called N_Ports).
- An N_Port is identified by its Worldwide Port Name (WWPN).
- For redundancy, servers and storage should attach through several N_Ports.
- sample WWPNs:
  FCP channel: `0xc05076ffe4803931`
  storage target: `0x500507630300010 4`

Logical Unit Numbers (LUNs)



Storage devices usually comprise many logical units (volumes, tape drives, ...).

A logical unit behind a target WWPN is identified by its
Fibre Channel Protocol
Logical Unit Number (FCP LUN).

Mind different LUN formats [T10 SAM], e.g.:

- DS8000 (pseudo flat space addressing, but with 2nd level: “SCSI MASK”):
  `0x40214040f00000000`
- SVC / V7000, XIV, FlashSystem, Tape (pseudo peripheral device addressing):
  `0x01c8000000000000`
  (FlashSystem LUN>255: flat space):
  `0x41fe000000000000`

Setup

Early Preparation

- Installation of a **new machine** using the WorldWide PortName Prediction Tool [http://www.ibm.com/servers/resourcelink/]
  - Input: IBM Z I/O configuration
  - Output: all virtual NPIV WWPNs for all FCP devices
    - can be used for early SAN zoning and storage target LUN masking even before activation of Z / LinuxONE machine
- **MES upgrade** of a machine migrating existing FCP workload without changing zoning or LUN masking
  - export WWPNs on old machine and import on new machine
  - Always transparent to Linux (it does not care about initiator WWPNs, only about target WWPNs and they only change with the storage)

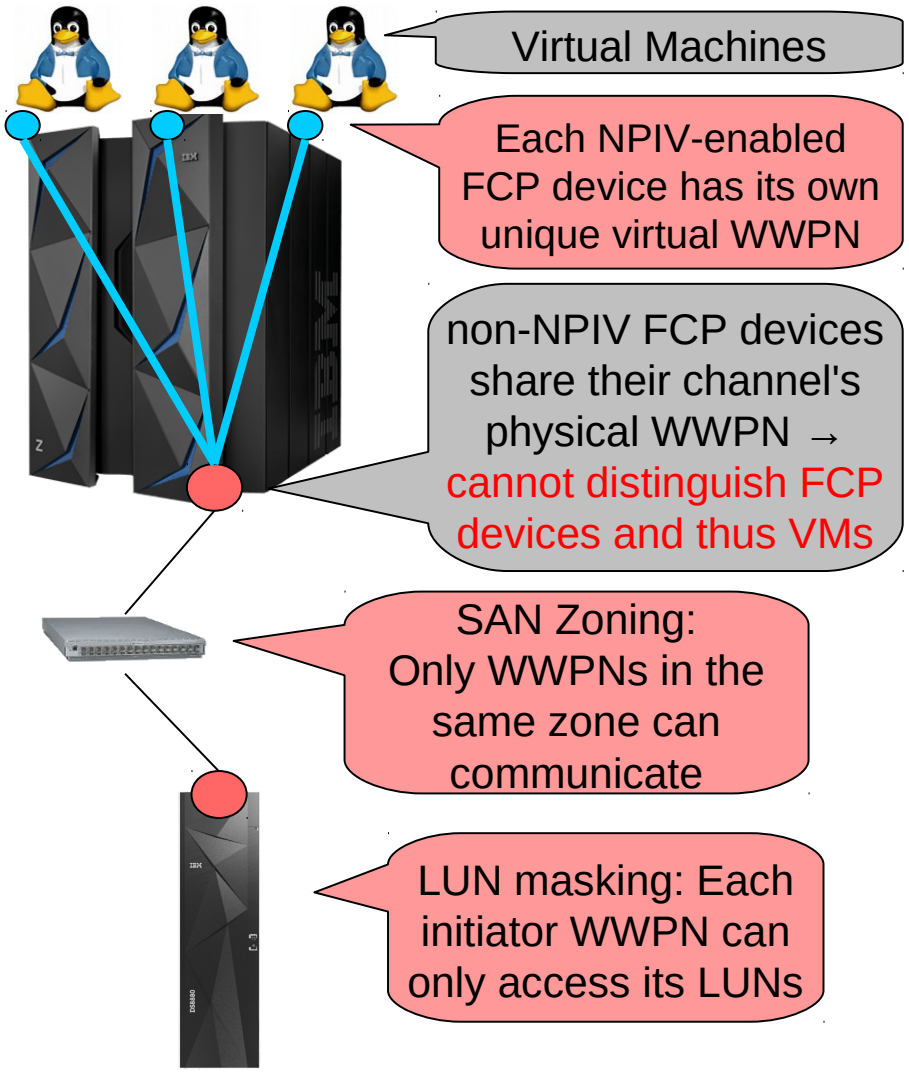
Define FCP Devices

- for LPAR hypervisor (PR/SM): use Dynamic Partition Manager (DPM) [doc][z13 GA2], or explicit virtual device config & passthrough as in following IOCDS example:

```
CHPID PATH=(CSS(0,1,2,3),51),SHARED,                              *
      NOTPART=((CSS(1),(TRX1),(=)),(CSS(3),(TRX2,T29CFA),(=)))*
      ,PCHID=1C3,TYPE=FCP
CNTLUNIT CUNUMBR=3D00,                                            *
      PATH=((CSS(0),51),(CSS(1),51),(CSS(2),51),(CSS(3),51)), *
      UNIT=FCP
IODEVICE ADDRESS=(3D00,001),CUNUMBR=(3D00),UNIT=FCP,SCHSET=3
IODEVICE ADDRESS=(3D01,007),CUNUMBR=(3D00),                       *
      PARTITION=((CSS(0),T29LP11,T29LP12,T29LP13,T29LP14,T29LP*
      15),(CSS(1),T29LP26,T29LP27,T29LP29,T29LP30),(CSS(2),T29*
      LP41,T29LP42,T29LP43,T29LP44,T29LP45),(CSS(3),T29LP56,T2*
      9LP57,T29LP58,T29LP59,T29LP60)),UNIT=FCP
IODEVICE ADDRESS=(3D08,056),CUNUMBR=(3D00),                       *
      PARTITION=((CSS(0),T29LP15),(CSS(1),T29LP30),(CSS(2),T29*
      LP45),(CSS(3),T29LP60)),UNIT=FCP
```

- for z/VM: dedicate 1 NPIV FCP device per CHPID per z/VM guest in its user dir.
- for KVM on IBM Z: on host, use 1 NPIV FCP device per CHPID per KVM guest

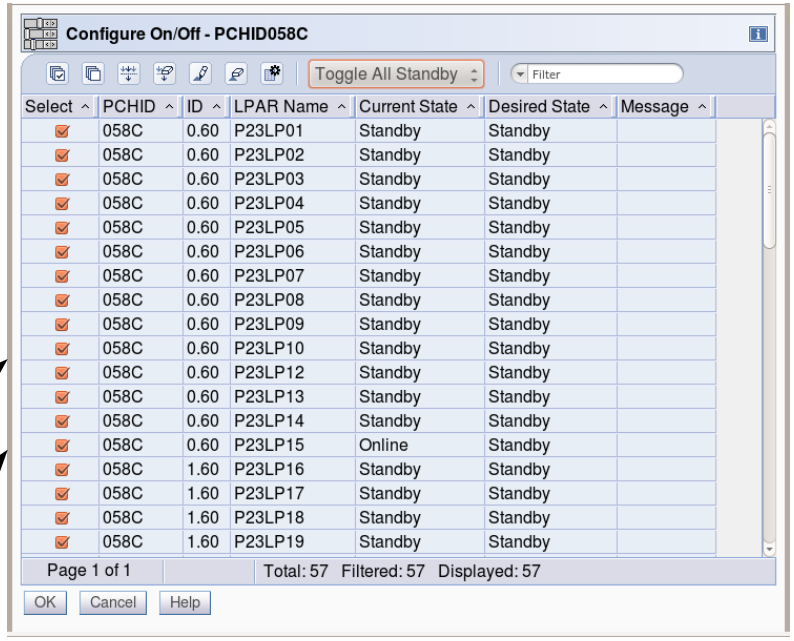
NPIV: N_Port ID Virtualization



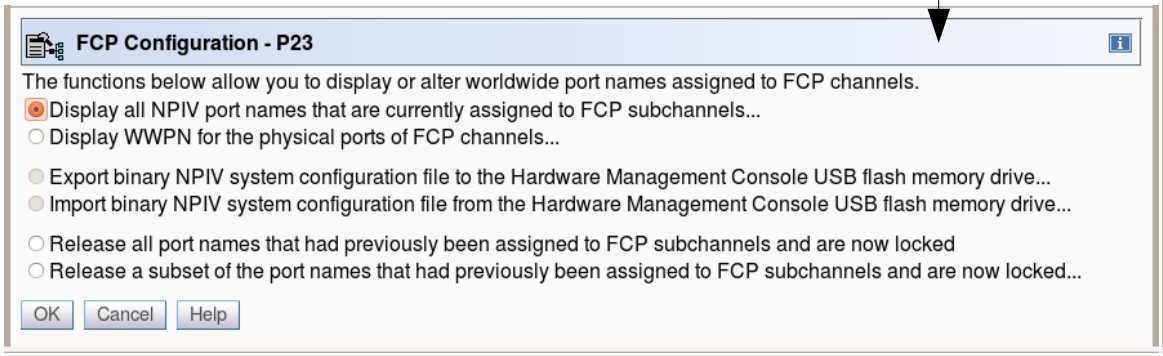
- Each virtual HBA uses FDISC with virtual WWPN to log into fabric and get its own N_Port ID [T11 FC-LS]
- Enable NPIV on the SAN switch before enabling it on the Z server.
- Switches typically limit the number of NPIV-enabled FCP devices per switch.
- Some switches limit the number of NPIV-enabled FCP devices per switch port.
- Each port login from an NPIV-enabled FCP device into a storage target counts as a separate host login, which are limited at storage.

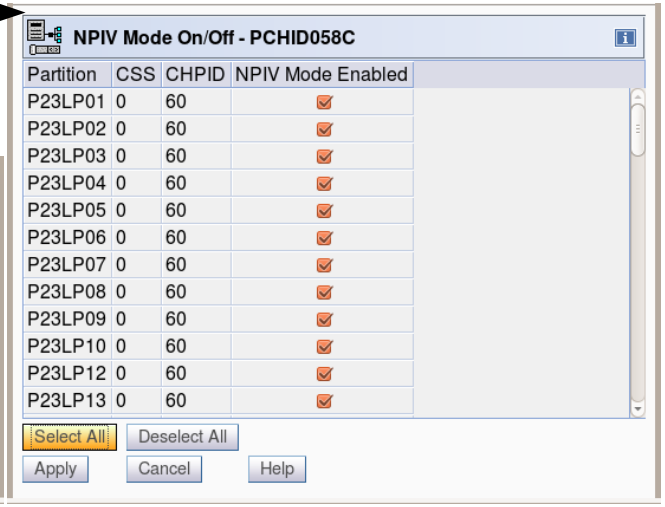
NPIV: Enable for all FCP Devices

- On the service element, for each FCP PCHID for each LPAR:
  1) Configure off its CHPID on LPAR
  2) Enable NPIV mode for LPAR
  3) Configure on its CHPID on LPAR if desired



- Manage FCP Configuration on the SE:





NPIV: ZFCP Point of View

- Is NPIV enabled for a certain FCP device?:

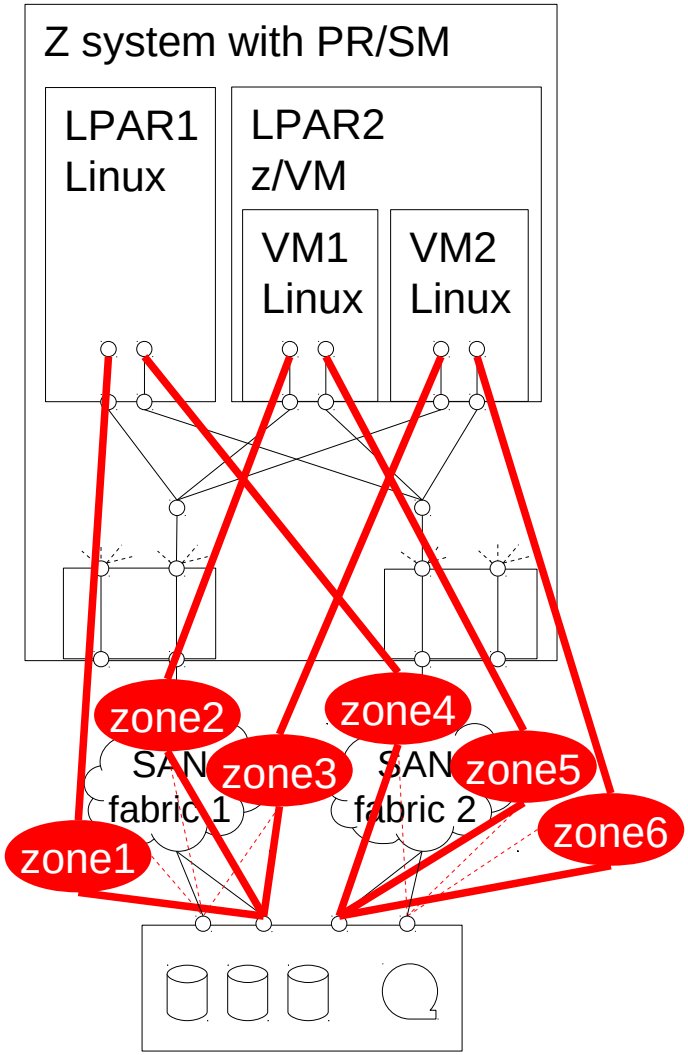
```
# lszfcp -Ha | grep -e port_type -e ^0
0.3.5a00 host0
    port_type               = "NPIV VPORT"
```

- alternatively for older Linux version (< SLES 11 SP1, < RHEL 6.0, < 2.6.30):

```
# lszfcp -Ha | grep -e port_name -e ^0
0.3.5a00 host0
    permanent_port_name = "0xc05076ffe5005611"
    port_name           = "0xc05076ffe5005350"
```

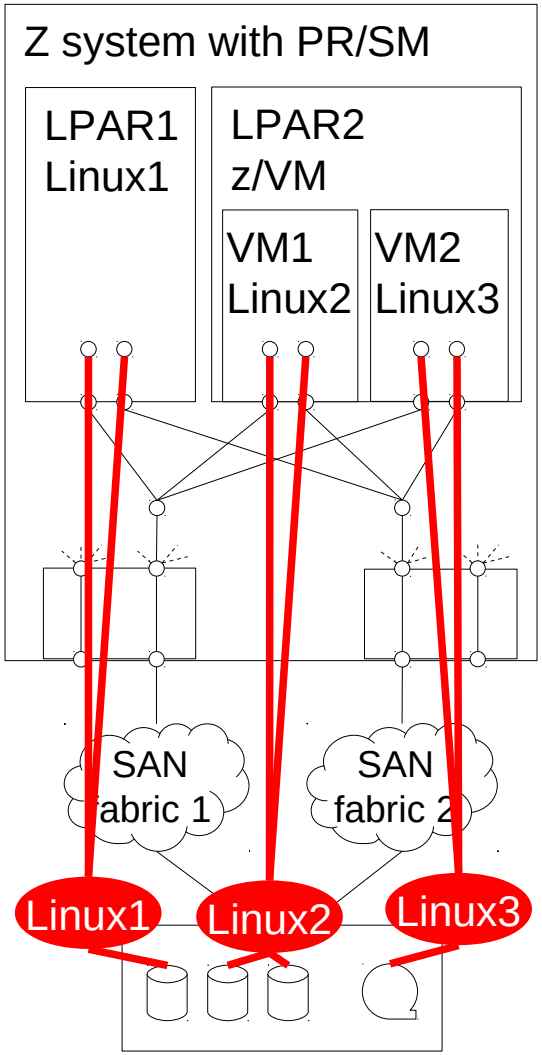
  - "permanent_port_name" is the WWPN assigned to the FCP channel
  - "port_name" is the WWPN used by the FCP device
  - if both port names differ NPIV is enabled, otherwise not

Zoning



- **Single Initiator Zoning based on WWPN** (as opposed to based on switch port):
  Have individual zone for each NPIV WWPN (FCP device), to avoid storms of change notifications and unnecessary recoveries.
- Since usually >1 initiator per target port, zones overlap at target ports
- Depending on storage recommendations, a zone can include multiple target ports
- If impossible: zFCP Auto Port Scan Resiliency

LUN Masking / Host Mapping on Storage



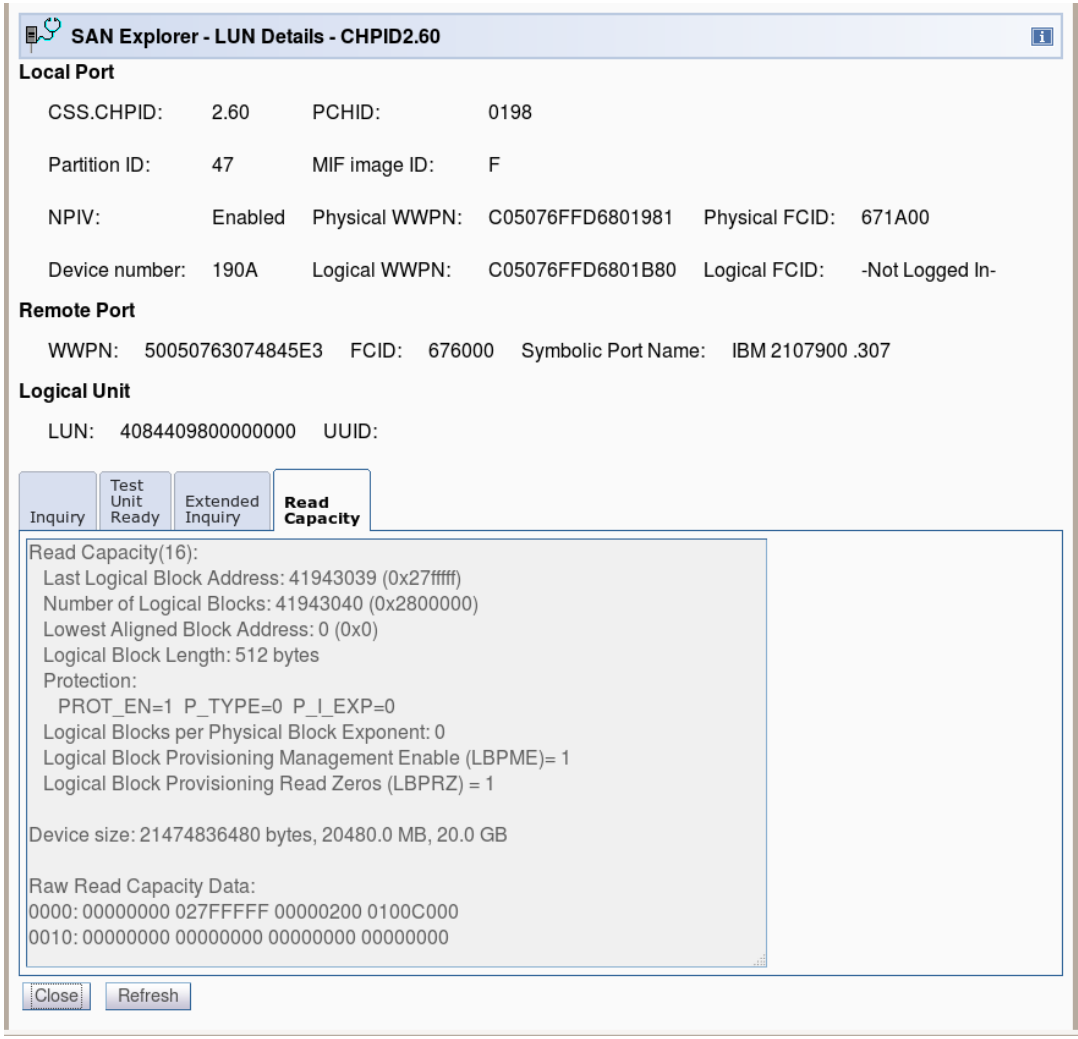
- In the storage target,
  use virtual initiator WWPNs of NPIV-enabled FCP devices to let each VM only access:
  - Its own exclusive logical units.
  - Logical units shared with other VMs (potentially on other physical machines).
    NOTE: Sharing requires OS support such as clustering file system!
- Depending on storage target type,
  this might require individual volume groups.

NPIV-Assist for Zoning and LUN Masking

- Needed login resources of NPIV-enabled FCP devices:
  - A set of NPIV-enabled FCP devices; log into the fabric to see host NPIV WWPN(s)
    during zoning on the SAN switches
  - A set of LUNs; log into a set of target port WWPNs to see host NPIV WWPN(s) during LUN masking / host mapping on the storage
- z/VM 6.4 (e.g. for bring-up preparation without running guest (OS))
  - FCP devices must be “free”, i.e. not dedicated / attached to a guest
  - `CP EXPlore FCP`
- any Linux
  - for FCP devices dedicated / attached to a guest / LPAR with Linux running, enable FCP devices with e.g. “chccwdev -e …” or “chzdev zfcp-host -ae …”

FCP SAN Explorer: Check I/O Configuration, Zoning, LUN Masking

- New function with z13 on the Hardware Management Console (HMC) / Service Element (SE)
- Machine must have completed IML
- Activate LPAR of interest
- Operating system ***not*** required, concurrent if OS runs in LPAR
- Select LPAR and FCP CHPID, “Channel Problem Determination”, “SAN explorer”
- Drill down:
  FCP devices,
  remote WWPNs (zone members),
  LUNs
- Since z13 GA2 also:
  - Diagnostic Data (RDP ELS)
  - Affinity (Active Zone Set)



Multipathing for Disks – LVM on Top

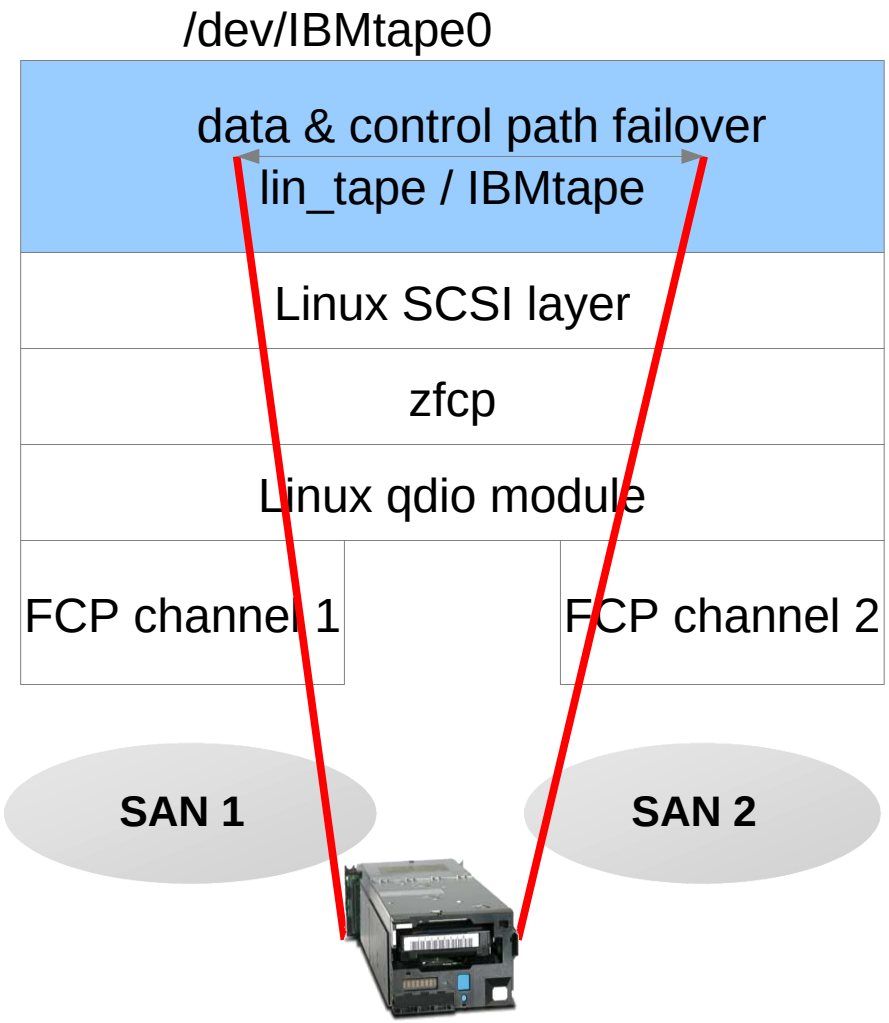
- explicitly ensure that all LVM PVs are **assembled** from multipath devices (/dev/mapper/…) instead of single path scsi devices (/dev/sd…)
  NOTE: pvcreate on multipath devices is necessary but not sufficient!
- otherwise PVs can randomly use only a single path anytime → lack of redundancy
- use a white list of explicitly allowed PV base device names in */etc/lvm/lvm.conf*:

```
global_filter = [ "a|^/dev/mapper/.*$|", "a|^/dev/dasd.*$|",
                  "a|^/dev/scm.*$|", "a|^/dev/dcssblk.*$|", "r|.*|" ]
```

- as of S12/R7/K/U apply config change once: `# systemctl restart lvm2-lvmetad`
- verify the correct filter for every SCSI disk device node using pvscan,
  “Skipping (regex)” must be shown:

```
# pvscan -vvv 2>&1 | fgrep '/dev/sd'
...
  /dev/sda: Added to device cache
  /dev/block/8:0: Aliased to /dev/sda in device cache
  /dev/disk/by-path/ccw-0.0.50c0-zfcp-0x1234123412341234:\
    0x0001000000000000: Aliased to /dev/sda in device cache
  ...
  /dev/sda: Skipping (regex)
```

Multipathing for IBM Tapes



Use Case:

- Backup with
  IBM Spectrum Protect /
  Tivoli Storage Manager (TSM)
  (client & server for Linux on Z)

Setup:

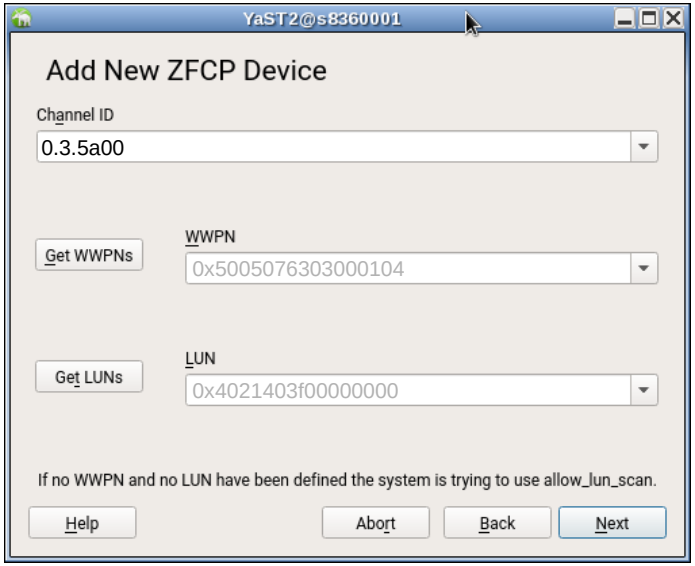
- enable via lin_tape module parameter
  e.g. in /etc/modprobe.conf.local:
  options lin_tape alternate_pathing=1
- attach all paths to tape drive

Multipathing – Error Recovery on SCSI Layer

- the following applies if the lower FC transport layer could not detect/recover errors, typically due to dirty fibres or SAN switches suppress RSCNs ⇐ must fix reasons

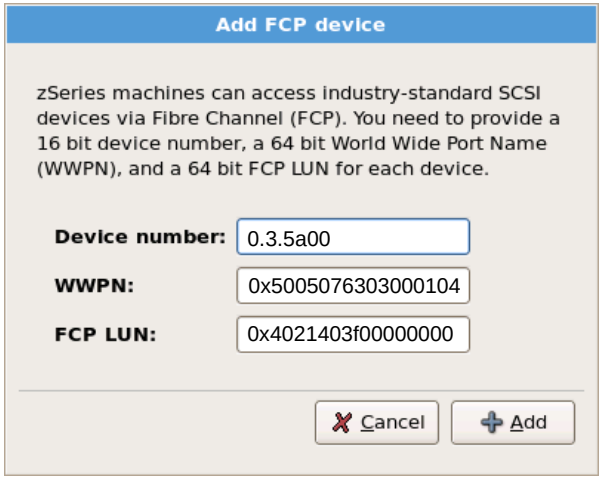
- on starting IO request: start SCSI command (=block request) timeout
- on timeout: start SCSI **E**rror **H**andling on SCSI host **as last resort**; multipathd can only see path failure once EH processed path checker IO request;
  - try to abort SCSI command (Upstream changed this to be outside of EH and handles it in a asynchronous fashion)
  - if above failed, try to reset device (=LUN), then TUR
  - if above failed, escalate and try to reset target, then TUR
  - if above failed, escalate and try to reset host (=FCP device recovery), then TUR
  - if above failed, finally give up: set SCSI device offline
- since above handling can take many minutes to complete, recent distros provide “eh_deadline” directly escalating to host reset after deadline

LUN Management with ZFCP: SLES Installation



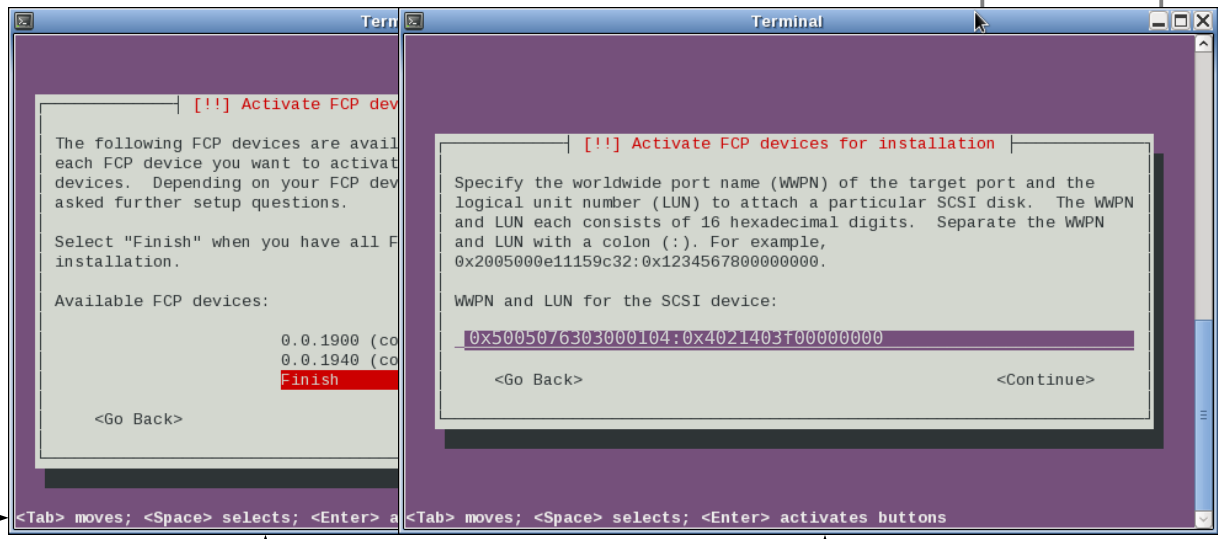
- interactive
  - GUI / TUI: YaST installer button “Configure ZFCP Disks”
  - GUI and TUI can discover available FCP devices, WWPNs, and LUNs
- unattended
  - AutoYaST: <zfcp> element
- auto LUN scan <SLES12SP2: specify just one valid path per FCP device.
  auto LUN scan ≥SLES12SP2: omit WWPN&LUN with YaST or AutoYaST.
- if you need to pass zfcp module parameters during installation via parm file [doc]:
  options="zfcp.*parameter1=value1 parameter2=value2*"

LUN Management with ZFCP: RHEL Installation



- interactive
  - GUI of installer (anaconda)
- unattended
  - kickstart: “zfcp” option
- both interactive and unattended
  - FCP_*n*='*device_bus_ID WWPN FCP_LUN*' in generic.prm or in a CMS conf file
  - RHEL7 also in generic.prm: rd.zfcp=*device_bus_ID*,*WWPN*,*FCP_LUN*
  - can also be used for e.g. install from SCSI LUN [doc1,doc2]
- RHEL5 installer boot parameter in generic.prm parmfile: “mpath”
- temp. workaround for auto LUN scan: specify just one valid path per FCP device

LUN Management with ZFCP: Ubuntu Installation



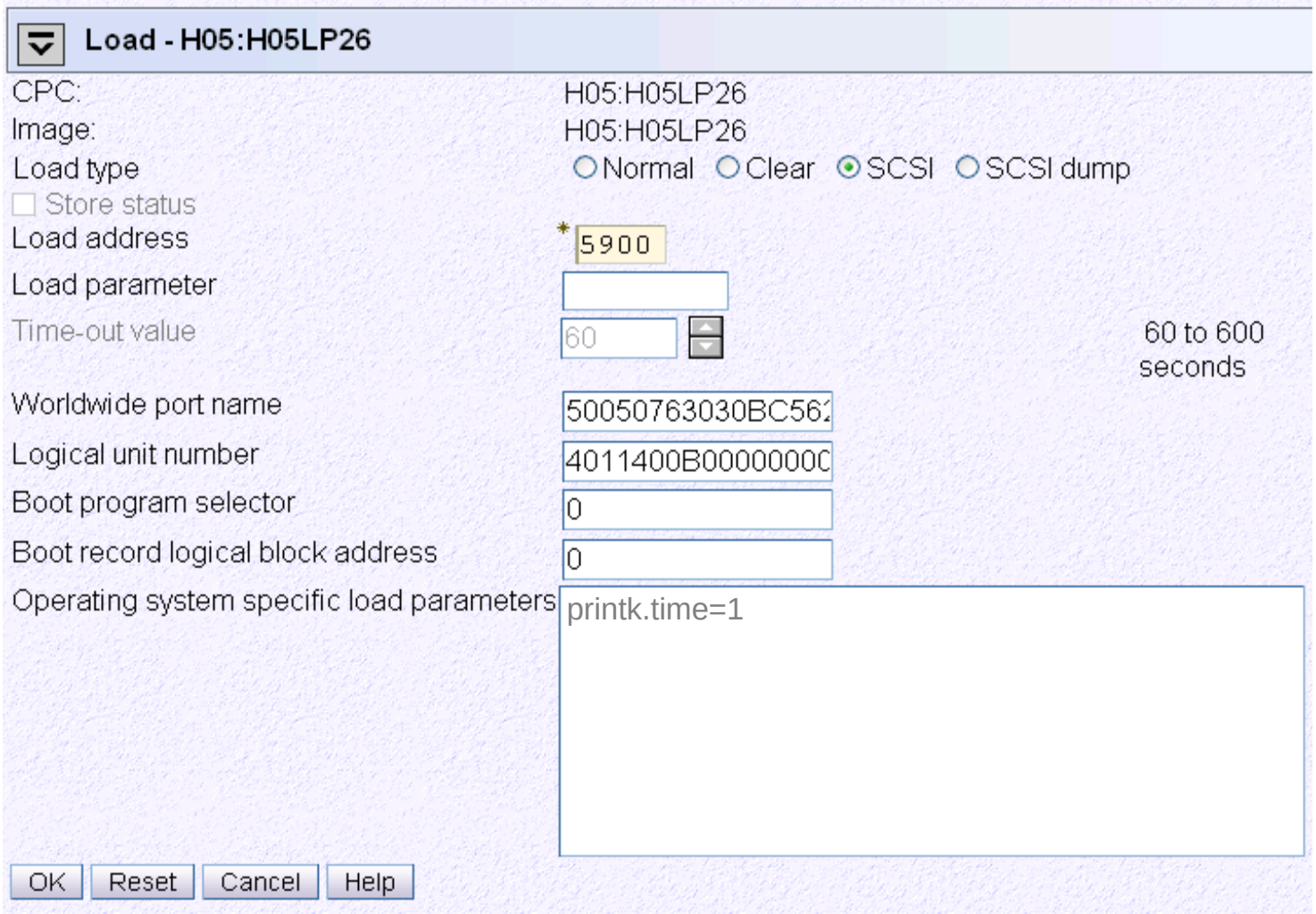
- interactive
  - TUI of installer
    - auto LUN scan active
    - auto LUN scan Inactive
  - preseeding: add one parameter for all paths to all disks required in installer to parm file used for booting installer; omit ':<*WWPN*>:<*LUN*>' if auto LUN scan active for this FCP device:

```
s390-zfcp/zfcp=0.3.5a00:0x5005076303000104:0x4021403f00000000,0.3.5b00:0x500507630300c104:0x4021403f00000000,0.3.fc00,0.3.fd00
```

- unattended
  - preseeding: add one statement for all paths to all disks to preseed file; omit ':<WWPN>:<LUN>' if auto LUN scan active for this FCP device:

```
d-i s390-zfcp/zfcp string 0.3.5a00:0x5005076303000104:0x4021403f00000000, \
                          0.3.5b00:0x500507630300c104:0x4021403f00000000, \
                          0.3.fc00,0.3.fd00
```

SCSI IPL example LPAR



SCSI IPL example z/VM

WWPN

LUN

in hexadecimal format with a blank between the first 8 from the final 8 digits

```
set loaddev port 50050763 03000104 lun 40214000 00000000
set loaddev bootprog 3 scpdata 'printk.time=1'

query loaddev
PORTNAME 50050763 03000104    LUN  40214000 00000000    BOOTPROG 3
BR_LBA   00000000 00000000
SCPDATA
     0----+----1----+----2----+----3----+----4----+----
0000 PRINTK.TIME=1

i 1900
00: HCPLDI2816I Acquiring the machine loader from the processor controller.
00: HCPLDI2817I Load completed from the processor controller.
00: HCPLDI2817I Now starting the machine loader.
00: MLOEVL012I: Machine loader up and running (version v2.4.4).
00: MLOPDM003I: Machine loader finished, moving data to final storage location.
…
Linux version 3.0.101-0.29-default (geeko@buildhost) (gcc version 4.3.4 [gcc-4_3-branch
  revision 152973] (SUSE Linux) ) #1 SMP Tue May 13 08:40:57 UTC 2014 (9ec28a0)
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
setup.dae2e8: Reserving 128MB of memory at 896MB for crashkernel (System RAM: 1024MB)
…
Kernel command line: root=/dev/mapper/36005076303ff0104000000000000002100
                     TERM=dumb crashkernel=256M-:128M BOOT_IMAGE=0 printk.time=1
```

device number of FCP device with access to SCSI boot disk (zipl target, typically /boot/(zipl/))

SCSI IPL Select Boot Menu Entry

- RHEL, SLES≤11, Ubuntu
  - Interactive: not available
  - Non-interactive: use “bootprog” IPL parameter to select zipl boot menu entry
- SLES12
  - Interactive: select grub menu entry during boot, key bindings for IBM Z
  - Non-interactive: requires ≥ grub2-2.02~beta2-***54.1***

```
# grub2-once --enum
0       SLES12
1>0     Advanced options for SLES12>SLES12, with Linux 3.12.43-52.6-default
1>1     Advanced options for SLES12>SLES12, with Linux 3.12.43-52.6-default (recovery mode)
1>2     Advanced options for SLES12>SLES12, with Linux 3.12.39-47-default
1>3     Advanced options for SLES12>SLES12, with Linux 3.12.39-47-default (recovery mode)
```

E.g. to boot 3.12.39-47, use the following value for “loadparm”: `g1.2`
(“bootprog” must be empty, 0, or 1)

SCSI IPL Dynamically Pass Kernel Parameters

- RHEL6, SLES11≥SP1, Ubuntu
  - Interactive: not available
  - Non-interactive: “operating system specific load parameters” / “scpdata”
- SLES12
  - Interactive:edit grub menu entry during boot, key bindings for IBM Z
  - Non-interactive: “operating system specific load parameters” / “scpdata”
    BUT these also affect the grub2-s390x-emu bootstrap environment!
- For persistent mechanism, see slide "Linux kernel parameters and ..."

- Methods to pass bootprog / scpdata / loadparm:
  - LPAR: HMC/SE Load task (Boot program selector / Operating system specific load parameters / Load parameter)
  - z/VM guest: #CP SET LOADDEV (bootprog / scpdata), #CP IPL (loadparm)
  - Linux: chreipl (--bootprog / --bootparms / --loadparm) from s390-tools

Troubleshooting

Troubleshooting: scsi_logging_level

- More SCSI output in kernel messages
- Default is: 0
- Higher levels can create lots of messages and slow down system due to synchronous output of kernel messages on the console → undesired errors!
  → low level and/or filter console kernel messages with /proc/sys/kernel/printk
- Find issues with LUN discovery and SCSI error handling (recovery) such as dirty fibres but only negligible impact on regular I/O
- Can be added to kernel parameters: "scsi_mod.scsi_logging_level=4605"

```
# scsi_logging_level -s \
  --mlcomplete 1 -T 7 -E 5 \
  -S 7 -I 0 -a 0
New scsi logging level:
dev.scsi.logging_level = 4605
SCSI_LOG_ERROR=5
SCSI_LOG_TIMEOUT=7
SCSI_LOG_SCAN=7
SCSI_LOG_MLQUEUE=0
SCSI_LOG_MLCOMPLETE=1
SCSI_LOG_LLQUEUE=0
SCSI_LOG_LLCOMPLETE=0
SCSI_LOG_HLQUEUE=0
SCSI_LOG_HLCOMPLETE=0
SCSI_LOG_IOCTL=0
```

Troubleshooting: debug data

- Check kernel messages that are possibly related to FCP with Linux on Z:
  - "device-mapper: multipath"
  - sd (SCSI disk)
  - lin_tape* (IBM tape)
  - scsi (common SCSI code)
  - rport (common SCSI code FC remote port messages)
  - zfcp
    - See "Kernel Messages" book on http://www.ibm.com/support/knowledgecenter/linuxonibm/com.ibm.linux.l0kmsg.doc/l0km_r_zfcp_container.html (for RHEL/Ubuntu, chose book from development stream with matching kernel version, there are no message IDs so you have to find by matching a message substring)
  - qdio (communication between zfcp and FCP device)
- Other syslog messages
  - multipathd (path management daemon for disks)
  - lin_taped (path management daemon for IBM tapes)
- zfcp driver traces available in /sys/kernel/debug/s390dbf/
- Collect data with **dbginfo.sh** (s390-tools) when reporting a problem to capture configuration, messages, and traces

Troubleshooting: performance

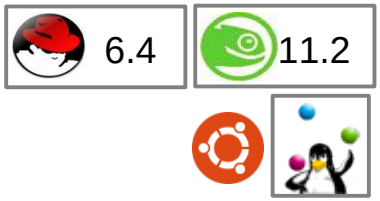
- zFCP Performance Analysis with ziomon
  http://www.vm.ibm.com/education/lvc/zlinlvc.html
  or
  https://share.confex.com/share/120/webprogram/Session13112.html
- more details in Linux on Z and LinuxONE documentation by IBM
  http://www.ibm.com/developerworks/linux/linux390/distribution_hints.html, or
  http://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_distlibs.html
  - How to use FC-attached SCSI devices with Linux on Z

Individual zFCP Features

FCP Hardware Data Router Support

- FCP hardware data router reduces path length and improves throughput depending on workload
- If not default, enable the hardware data router feature in zfcp with kernel parameter "zfcp.datarouter=1"
- check whether the zfcp module parameter datarouter was enabled or disabled:
  `# cat /sys/module/zfcp/parameters/datarouter`
  Y
- under z/VM: show if datarouter is active per FCP device: **`#CP Q V FCP`**
- Note: The hardware data routing feature becomes active only for FCP devices that are based on adapter hardware with hardware data routing support.
- Hardware data router requirements:
  - at least: zEnterprise 196 GA2 or zEnterprise 114; FICON Express8S
  - LPAR. z/VM: guest support available beginning with z/VM 6.3.
  - RHEL 6.4, SLES 11SP3; enabled by default: RHEL7, SLES12, KVM, Ubuntu

End-to-end (E2E) data integrity (T10 DIF)



- End-to-end data integrity checking is used to confirm that a data block originates from the expected source and has not been modified during the transfer between the storage system and the FCP device
- To turn end-to-end data integrity checking on
  set the kernel parameter "zfcp.dif=1"
- check whether the FCP device supports end-to-end data integrity checking, use the lszfcp command and limit the query to a specific FCP device

```
# lszfcp -b 0.0.1700 -Ha |grep prot_capabilities
1
```

  - 0 means: FCP device does not support end-to-end data integrity.
  - 1 means: FCP device supports DIF type 1.
- E2E data integrity checking requirements:
  - at least: zEnterprise 196 GA2 or zEnterprise 114; FICON Express8
  - LPAR. z/VM: guest support since 5.4 & 6.1 (both with PTFs for APAR VM64925)
  - T10 DIF support for SCSI disk only (e.g. DS8000 since release 6.3.1)
  - RHEL 6.4 & 7, SLES 11SP2 & 12, KVM, Ubuntu

EXPERIMENTAL: End-to-end (E2E) data integrity extension (DIX)

7.0 11.2

- Data integrity extension (DIX) builds on DIF to extend integrity checking, e.g. to the operating system, middleware, or an application.
- SCSI devices for which DIX is enabled must be accessed as raw block device with direct I/O (unbuffered I/O bypassing the page cache) or through a file system that fully supports stable page writes, e.g. XFS. Expect error messages on invalid checksums with other access methods.
- Find out about end-to-end data integrity support of an FCP device:

```
# lszfcp -b 0.0.1700 -Ha |grep prot_capabilities
17
```

  - 0 means: FCP device does not support end-to-end data integrity.
  - 1 means: FCP device supports DIF type 1.
  - 16 means: FCP device supports DIX type 1.
  - 17 means: FCP device supports DIF type 1 with DIX type 1.

Zoning: Limited automatic port rescan on events

- Based on slide about Zoning: Implement single initiator zones (based on (virtual) WWPNs)
- If single initiator zones are impossible:
- Proper solution is zfcp auto port scan resiliency [R6.7,R7.2,S12SP1,KVM,U]
- For older distros [as of RHEL6.4,SLES11SP3], as a workaround, disable automatic port rescanning by setting kernel parameter:
  `zfcp.no_auto_port_rescan=1`
  - Ports are still unconditionally scanned when the adapter is set online and when user-triggered writes to the sysfs attribute “port_rescan” occur.
  - On fabric changes, manually trigger a port rescan by running:
    `# echo 1 > /sys/bus/ccw/drivers/zfcp/0.0.1700/port_rescan`
  - Automatic port rescanning is enabled by default.