

Getting started with IBM Spectrum Scale for Linux on IBM Z

*Built upon General Parallel File System
(GPFS)*

*Susanne Wintenberger
swinten@de.ibm.com
IBM R&D Boeblingen*

*Hendrik Brueckner
brueckner@de.ibm.com
IBM R&D Boeblingen*

Table of Contents

| | |
|---|----|
| 1 Abstract..... | 2 |
| 2 Introduction to IBM Spectrum Scale technology | 3 |
| 2.1 IBM Spectrum Scale cluster configurations..... | 3 |
| 2.2 Shared-disk model..... | 3 |
| 2.3 Network shared disk (NSD) client/server model..... | 4 |
| 3 Storage considerations | 6 |
| 3.1 Setup and usage of ECKD devices..... | 6 |
| 3.1.1 Hints and tips | 8 |
| 3.2 Setup and usage of ECKD devices on z/VM..... | 8 |
| 3.2.1 Full-pack minidisks..... | 10 |
| 3.2.2 Dedicated devices..... | 11 |
| 3.2.3 Hints and tips | 11 |
| 3.3 Setup and usage of SCSI devices | 11 |
| 3.3.1 Setup of FCP devices on z/VM..... | 11 |
| 3.3.2 Setup of SCSI devices | 12 |
| 3.3.3 Multipathing..... | 13 |
| 3.3.4 Hints and tips | 15 |
| 3.3.4.1 Blacklisted devices | 15 |
| 3.3.4.2 SCSI disk path failover | 15 |
| 4 Network considerations..... | 16 |
| 5 Memory considerations | 17 |
| 6 Setting up IBM Spectrum Scale..... | 17 |
| 6.1 Prerequisites | 17 |
| 6.2 Additional kernel parameters | 18 |
| 6.3 Communication without a password | 19 |
| 6.4 System time coordination in an IBM Spectrum Scale cluster | 20 |
| 6.5 Structure of IBM Spectrum Scale components | 20 |
| 6.6 Installing the IBM Spectrum Scale packages and building the IBM Spectrum Scale kernel modules | 21 |
| 6.7 Planning the IBM Spectrum Scale cluster | 22 |
| 6.8 Creating the IBM Spectrum Scale cluster, Network Shared Disks, and an IBM Spectrum Scale file system | 23 |
| 6.9 Hints and tips..... | 27 |
| 6.9.1 Cluster node connection issues..... | 27 |
| 6.9.2 Start of the IBM Spectrum Scale daemon fails | 28 |
| 7 References | 28 |

1 Abstract

IBM Spectrum Scale for Linux on IBM Z™ provides superior resiliency, scalability, high-performance data and file management - build upon IBM General Parallel File System (GPFS™). IBM Spectrum Scale provides system scalability, very high availability and reliability with no single point of failure. Administrators can configure the file system so that it automatically remains available if a disk or server fails. IBM Spectrum Scale can speed time-to-results and maximize utilization by providing parallel access to data and shared disks, improving scalability for high-performance workloads. In addition, IBM Spectrum Scale can automatically spread data across multiple storage devices—optimizing available storage utilization and delivering high performance where needed.

IBM Spectrum Scale is intended for workloads where performance, reliability, and availability of data are essential to the business; you can manage files through both Linux instances running in logical partition (LPAR) mode or on z/VM®.

IBM Spectrum Scale for Linux on Z V5.1.0 is available in the Data Management (Advanced) Edition and the Standard Edition. For more details about features supported on Linux on Z, please refer to the Frequently Asked Questions:

<http://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.html>

The paper covers the following topics:

- How to set up Linux on Z for Spectrum Scale
- How to install Spectrum Scale, how to set up and configure a Spectrum Scale cluster, and how to create a file system
- Best practices, hints, and tips

The setup of the features is not described in this paper, refer to chapter “7 References”.

Acknowledgements

The authors would like to thank the following people: Henry Calzaretta, Steve Duersch, Maria Eisenhaendler, Eugen Haas, Kenneth Hall, Gerald Hosch, Richard Lewis, Shuzheng Liu, Peter Muench, Aleksandra Pavic, Monika Schmidt and Frederick Stock

2 Introduction to IBM Spectrum Scale technology

This document can only give a short introduction to the Spectrum Scale technology (formerly GPFS). If you are not familiar with it, first read the Concepts parts of the *Spectrum Scale Concepts, Planning and Installation Guide* available from:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.1.0/com.ibm.spectrum.scale.v5r10.doc/

The Linux on Z instances can vary:

- The distribution can be either Red Hat Enterprise Linux and SUSE Linux Enterprise Server.
- The instances can run in logical partition (LPAR) mode or as guest operating systems of z/VM.
- They can run within the same mainframe system or they can be distributed across multiple mainframe systems.

2.1 IBM Spectrum Scale cluster configurations

Spectrum Scale for Linux on Z supports the following models of cluster configuration:

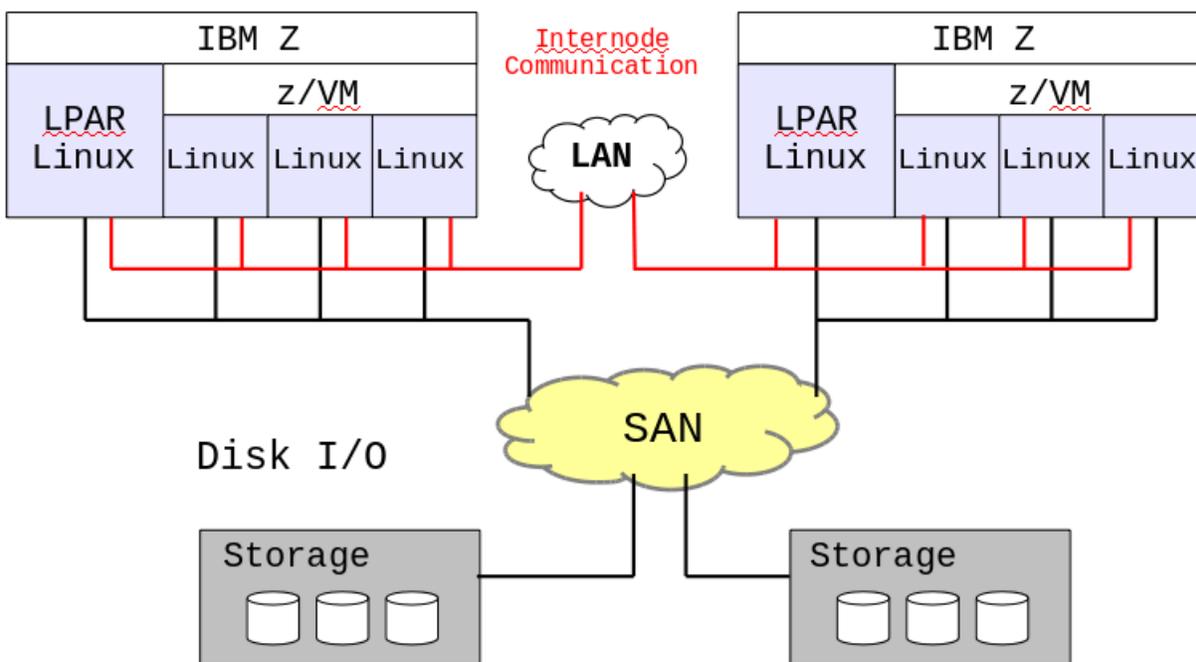
- In the shared-disk model the cluster nodes, the single operating system images, can access a set of disks directly.
- In the Network Shared Disk (NSD) client/server model, the shared-disk model is extended by mixing direct SAN access with network-attached cluster nodes. Network-attached cluster nodes access storage over a network connection.

Depending on the cluster configuration, you must plan for the underlying hardware (storage and network) and the appropriate configuration to gain, for example, the needed performance and reliability.

2.2 Shared-disk model

The characteristics of the shared-disk model are:

- Every cluster node has direct access to the storage disks through the SAN.
- Internode communication is enabled through network connections.



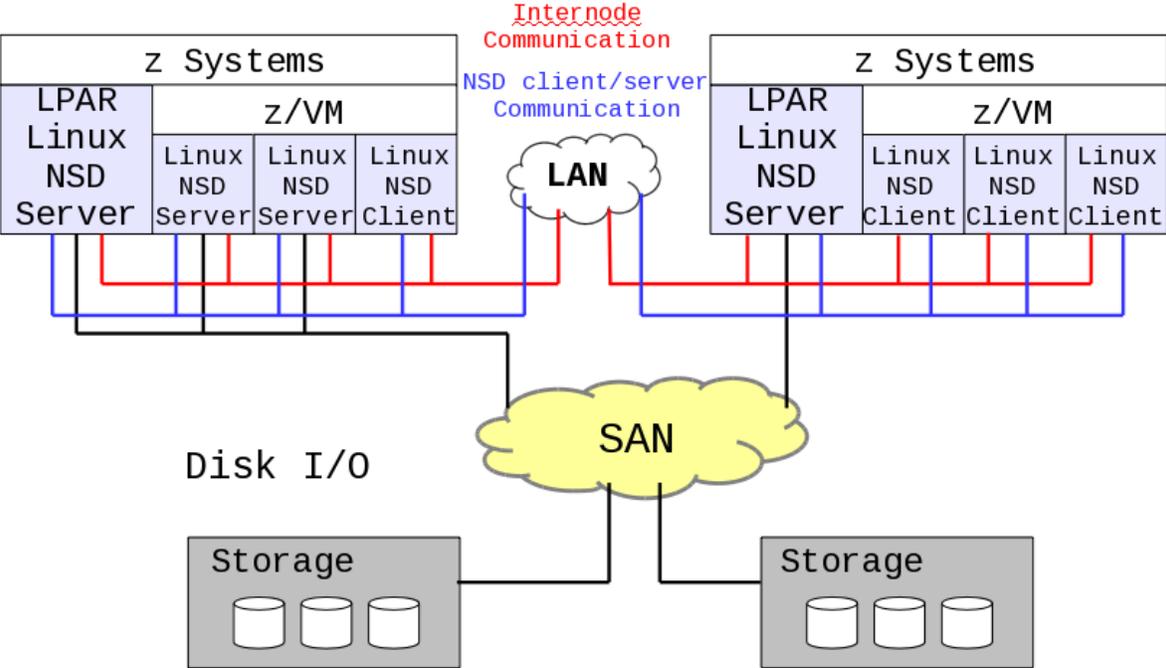
The main advantage of using a disk configuration that directly attaches and accesses the data is that Spectrum Scale does not need to use the network to send disk data from one cluster node to another. Hence this architectural model achieves a better performance than the NSD client/server model.

2.3 Network shared disk (NSD) client/server model

The characteristics of the network shared disk client/server model are:

- Some of the cluster nodes have direct access to the storage disks through the SAN and act as an NSD server to serve disk I/O operations for NSD clients.
- Other cluster nodes are defined as NSD clients and access data over a network connection and through an NSD server cluster node. In this case, besides the disk I/O through the SAN, the cluster nodes have network access through the NSD network protocol.
- Internode communication is still through network connections.

Getting started with IBM Spectrum Scale for Linux on IBM Z



Compared to the shared-disk model, only a couple of cluster nodes have direct access to the data disks. When a NSD client node needs to read or write to a disk, Spectrum Scale transfers the I/O request to an NSD server node. The node will then, for example, write it directly to the disk. The network in an NSD client/server configuration is heavily used for each read and write request of NSD client nodes. Thus a very high-speed network with low latency should be used for the best performance.

If an NSD client cannot access an NSD server due to a server failure, and thereby cannot access a disk, Spectrum Scale suspends using the disk. Use disk connectivity on multiple NSD server nodes and specify multiple NSD servers for each disk to guard against loss of NSD server availability.

In this case Spectrum Scale dynamically switches to the next available NSD server and continues to provide data.

You can also set up an NSD client node with both direct access to the data disks through the SAN and access to the disks through an NSD server node. If direct access is lost, for example due to a SAN problem, the data access is assured through the available NSD server.

Heterogeneous clusters are limited to the following: You can set up a heterogeneous cluster with x86 servers and POWER® servers running Red Hat Enterprise Linux or SUSE Linux Enterprise Server. These cluster nodes must be configured as NSD clients to access storage on Linux on Z cluster nodes. Do not share storage among different platforms, for example, by using local storage access.

Getting started with IBM Spectrum Scale for Linux on IBM Z

As of Spectrum Scale V4.2.2, some limitations of the heterogeneous clusters are lifted: You can set up a heterogeneous cluster with x86 servers, POWER servers and z servers running AIX, Red Hat Enterprise Linux and SUSE Linux Enterprise Server. The cluster nodes can serve as NSD servers to access their local platform-dependent storage. However, for Linux on Z SAN-attached disks can only be accessed from Linux on Z cluster nodes. Mainframe storage cannot be shared among different platforms.

3 Storage considerations

Linux on Z uses two different types of devices and device drivers respectively.

The DASD device driver provides access to real or emulated Direct Access Storage Devices (DASD) that can be attached to the channel subsystem of a mainframe. This device driver supports the Extended Count Key Data (ECKD™) and Fixed Block Access (FBA) devices.

Note: As of Spectrum Scale V4.2.2, DASDs that are recognized in Linux as FBA devices, for example emulated FBA devices, are supported. CMS-formatted devices are not supported for use by Spectrum Scale.

The zfcps device driver supports virtual SCSI-over-Fibre Channel adapters (FCP devices) and attached SCSI devices (LUNs).

For more details, see *Device Drivers, Features, and Commands*, available from

https://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_ddcnt.html

3.1 Setup and usage of ECKD devices

Prerequisites:

The ECKD devices are shared among Linux instances. While you set up shared ECKD devices from Linux, the devices must be offline to all other nodes. To ensure that the devices are offline, you can shut down the cluster nodes and boot them later.

To prepare ECKD devices for use by Spectrum Scale, complete these steps on a single cluster node:

1.) Ensure that each ECKD device is visible to the Linux instance. Verify the device exclusion list using the following command:

```
# cat /proc/cio_ignore
```

If one or more ECKD devices are listed in the output, the devices are not visible to the Linux instance. In this case, issue the command `cio_ignore` to remove each ECKD device of the output from the device exclusion list.

```
# cio_ignore -r <eckd-devno>
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

Verify the device exclusion list again by using the command above.

2.) Ensure that each ECKD device is online. To set a device online, issue the following command:

```
# chccwdev -e <eckd-devno>
```

3.) Low-level format each ECKD device.

Note: Spectrum Scale supports ECKD devices in either the compatible disk layout (CDL) format or the Linux disk layout (LDL) format. The compatible disk layout (CDL) format is the default and preferred format. In both cases the ECKD device must be formatted with a block size of 4096.

To format an ECKD disk with the CDL format, issue the following command:

```
# dasdfmt -d cdl -b 4096 /dev/dasd<x>
```

4.) For CDL-formatted devices only: Partition each ECKD device with the `fdasd` command and create, for example, a single partition that spans the complete device.

```
# fdasd -a /dev/dasd<x>
```

Note: This step is not required for LDL-formatted devices because the `dasdfmt` command automatically creates a single Linux partition on the disk.

5.) Ensure that the steps 1-4 are completed for each ECKD device that you want to use with Spectrum Scale.

6.) To ensure that your Linux instance comes up with the disk configuration after a reboot, you must make your configuration changes persistent across reboots.

On Red Hat Enterprise Linux:

Add the devices to the configuration file `/etc/dasd.conf`, for example:

```
# cat /etc/dasd.conf
0.0.<eckd-devno>
```

On SUSE Linux Enterprise Server:

Use either the graphical user interface YAST2 (Yet another Setup Tool) to activate all the previously defined and configured ECKD devices in the DASD Disk Management dialog, or use the command line tool `dasd_configure`. Use this command separately for each ECKD device.

```
# dasd_configure 0.0.<eckd-devno> 1
```

As soon as you activate an ECKD device, the appropriate `udev` rule will be written. This `udev` rule ensures that an ECKD device is set online after the next reboot.

Getting started with IBM Spectrum Scale for Linux on IBM Z

7.) As soon as you have finished the configuration of the ECKD devices on one cluster node, you can continue with the set up of all other cluster nodes.

- a.) Set the ECKD devices online on every cluster node (steps 1 and 2)
- b.) Make your configuration change persistent across reboots (step 6)

Note: There is no need to (re-)format or (re-)partition the devices again. Do not repeat the steps 3 – 5, when you set up the other cluster nodes.

For more details about how to set up ECKD devices, see *Device Drivers, Features, and Commands*, available from

https://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_ddcnt.html

3.1.1 Hints and tips

The DASD device driver of Linux on Z supports the reserve/release mechanism for ECKD devices. Applications or administrators can use reserve/release with the `tunedasd` command. Do not perform any reserve/release operations, for example, with the `tunedasd` command, on ECKD devices that are used for Spectrum Scale.

Data integrity for shared disks is handled by Spectrum Scale itself.

3.2 Setup and usage of ECKD devices on z/VM

When your Linux instance runs as a z/VM guest operating system there are a couple of ways you can share an ECKD device. Prior to any usage of the ECKD device in the Linux operating system, you have to set up how to share your ECKD devices in the z/VM guest virtual machine.

Note: CMS-formatted devices are not supported for use by Spectrum Scale.

There are two ways to access an ECKD device on z/VM:

- Using minidisks
- Using the ECKD device as a dedicated device

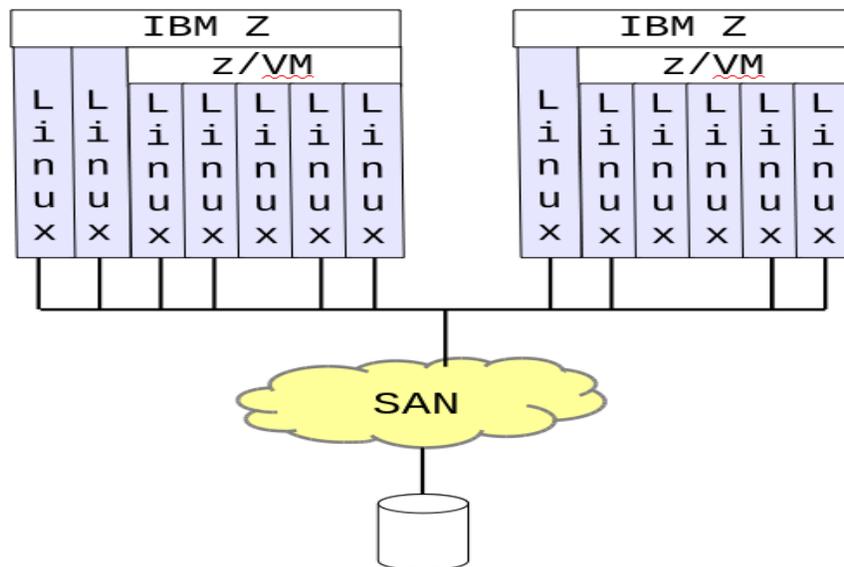
A minidisk is a contiguous range of cylinders on an ECKD device. Minidisks can span over a complete ECKD device, full-pack minidisk, or can be allocated on a portion of an entire ECKD device, non full-pack minidisk, or real minidisk. In the latter case, multiple minidisks can be allocated on the same ECKD device. Unlike dedicated devices, minidisks can be shared by multiple guest virtual machines of the same z/VM system and of other z/VM systems.

By creating a full-pack minidisk you can share the real ECKD device between guest virtual machines of the same and another z/VM system. You can also share it with Linux instances running in LPAR mode. In

Getting started with IBM Spectrum Scale for Linux on IBM Z

regard to Spectrum Scale, whenever it is possible, prefer full-pack minidisks over non full-pack minidisks that only span over a portion of the ECKD device.

In the following an example of sharing an ECKD device among Linux instances in LPAR mode and Linux guest virtual machines on multiple z/VM systems. In this case, you must define the ECKD device as a full-pack minidisk.



The other way to share a disk is to dedicate the whole ECKD device to a particular guest virtual machine. Dedicating a disk means that within the z/VM system, only this guest virtual machine has access to the ECKD device. Dedicated ECKD devices can be shared by logical partitions. For example, by sharing across LPARs and dedicating within z/VM systems you can configure access to the ECKD for one Linux instance per LPAR.

Both full-pack minidisks and dedicated devices can also be shared among z/VM systems that are members of the same z/VM Single System Image (SSI) cluster. They can also be shared across z/VM SSI clusters.

It is not possible to provide examples for all possible configurations discussed above. With Spectrum Scale for Linux on Z, it is most likely that the entire ECKD devices will be used rather than portions of an ECKD device. Hence the steps that follow for defining ECKD devices use entire ECKDs.

To define an ECKD device that is concurrently shared among multiple logical partitions, you must make the device shareable to z/VM. To make it shareable, share the ECKD device physically among the logical partitions in the IOCDs. To share an ECKD device among multiple mainframes, make it shareable in your Storage Area Network.

Getting started with IBM Spectrum Scale for Linux on IBM Z

More details: z/VM: CP Planning and Administration

https://www.ibm.com/support/knowledgecenter/SSB27U_7.2.0/com.ibm.zvm.v720.hcpa5/dshar.htm

3.2.1 Full-pack minidisks

To share an ECKD device as a full-pack minidisk among Linux instances running on the same z/VM system, a specific configuration is required.

The entire ECKD device must be accessible to other guest virtual machines in the same z/VM system. To achieve this, you must define the device in the user directory of one of the guest virtual machines. You can use an `MDISK` statement that covers all cylinders of the device (`0` to `END`). Alternatively, use an `MDISK` statement with the `DEVNO` operand.

To define an entire ECKD device as a shareable full-pack minidisk in multiple-write (MW) mode, use one of the following statements:

```
MDISK <virtual address> 3390 000 END <volser> MW <passwords>
MDISK <virtual address> 3390 DEVNO <real address> MW <passwords>
```

Note: If your Spectrum Scale cluster includes minidisks, do not use virtual reserve/release for minidisks. There is no reserve/release support at all for ECKD devices in the context of Spectrum Scale. Data integrity for shared ECKD devices is handled by Spectrum Scale itself.

Other guest virtual machines of the same z/VM system can access the same full-pack minidisk by using the `LINK` directory statement in their user directory:

```
LINK <guest1> <virtual address> <target virtual address> MW <password>
```

Do not use the minidisk cache when you share the full-pack minidisk that is defined with an `MDISK` statement that covers all cylinders of the device (`0` to `END`). The disk can be shared among multiple z/VM systems, or across z/VM SSI clusters, or with Linux instances in LPAR mode. To ensure that minidisk cache is not used, define the particular real ECKD device in the `SYSTEM CONFIG` file of the z/VM system by using an `RDEVICE` statement with the `MDC` operand. For example:

```
RDEVICE <real device number> type DASD MDC OFF
```

This full-pack minidisk can now also be shared with Linux instances running in LPAR mode, or with Linux guest virtual machines of multiple z/VM systems, or across z/VM SSI clusters.

As soon as the device is defined for your Linux guest operating system, continue with the same procedures as described in the previous section.

To display the guest virtual machines that have access to an ECKD device through minidisk links, issue the following command – be aware that only active guest virtual machines are listed:

Getting started with IBM Spectrum Scale for Linux on IBM Z

```
# CP query links <virtual device number>
```

```
[command output:
```

```
GUEST1 960A R/W, GUEST2 960A R/W, GUEST3 960A R/W, GUEST4 960A R/W]
```

3.2.2 Dedicated devices

Entire ECKD devices that are dedicated to a single guest virtual machine are for exclusive use of that guest virtual machine. They are not available to any other guest virtual machine of this particular z/VM system. For example, use the following `DEDICATE` directory statement to add the device to the user directory of the guest virtual machine and then (re-) log in to your guest virtual machine:

```
DEDICATE <virtual device number> <real device number>
```

As soon as the device is defined for your guest virtual machine, continue with the same procedures as described in the previous section.

3.2.3 Hints and tips

When you plan the administration of your z/VM systems or z/VM SSI clusters, note the following:

- Data integrity is handled by Spectrum Scale. You do not need to use real reserve/release for dedicated ECKD devices or full-pack minidisk. Spectrum Scale uses a sophisticated token-management system to provide data consistency while it allows simultaneous shared read and write access, with multiple independent paths to the same file from anywhere in the system.
- Defining a full-pack minidisk with the `DEVNO` statement avoids concerns with minidisk cache.
- It is possible to relocate a guest virtual machine as long as the virtual machine meets all relocation eligibility requirements.

For more details, see *z/VM: Getting started with Linux on z Systems*, available from

https://www.ibm.com/support/knowledgecenter/SSB27U_7.2.0/com.ibm.zvm.v720.hcpl0/preplgr.htm

3.3 Setup and usage of SCSI devices

To use SCSI devices on Linux on Z, you must connect the SCSI devices to an FCP adapter.

3.3.1 Setup of FCP devices on z/VM

When your Linux instance runs on z/VM, you must update the z/VM user directory. Use the following `DEDICATE` directory statement to add the FCP device to the guest virtual machine:

Getting started with IBM Spectrum Scale for Linux on IBM Z

```
DEDICATE <virtual FCP device number> <real FCP device number>
```

3.3.2 Setup of SCSI devices

The setup procedure depends on the Linux distribution.

If you are using Red Hat Enterprise Linux:

1.) Ensure that the FCP adapter used for Spectrum Scale is visible to the Linux instance. Verify the device exclusion list using the following command:

```
# cat /proc/cio_ignore
```

If the FCP adapter is listed in the output, the FCP adapter is not visible to the Linux system. In this case, use the `cio_ignore` command to remove the FCP adapter from the device exclusion list.

```
# cio_ignore -r <FCP-devno>
```

Verify the device exclusion list again with the command above.

2.) Ensure that the FCP adapter is online. You can issue the following command:

```
# chccwdev -e <FCP-devno>
```

3.) Define any SCSI devices that are not used to boot the Linux on Z instance in the `/etc/zfcp.conf` file.

```
# cat /etc/zfcp.conf
0.0.1900 0x5005076303000104 0x4021400000000000
```

4.) Activate any changes by using the following script.

```
# /sbin/zfcpconf.sh
```

For more details, see *Red Hat Enterprise Linux Installation Guide*, available from

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/

If you are using SUSE Linux Enterprise Server:

1.) Ensure that the FCP adapter used for Spectrum Scale is visible to the Linux instance. Verify the device exclusion list using the following command:

```
# cat /proc/cio_ignore
```

If the FCP adapter is listed in the output, the FCP adapter is not visible to the Linux system. In this case, use the `cio_ignore` command to remove the FCP adapter from the device exclusion list.

Getting started with IBM Spectrum Scale for Linux on IBM Z

```
# cio_ignore -r <FCP-devno>
```

Verify the device exclusion list again with the command above.

2.) Use either the graphical user interface YAST2 (Yet another Setup Tool) to remove the FCP adapter and to activate the new boot configuration, or edit the boot loader configuration file `/etc/default/grub`. Remove the FCP adapter from the list of devices to be ignored during the boot process.

```
# cat /etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT=" ... cio_ignore=all, ..., !<FCP-devno> ..."
```

Always activate the new boot configuration with the corresponding grub command after changing the boot loader configuration file.

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Then reboot your Linux on Z instance.

3.) Ensure that the FCP adapter is online. You can issue the following command:

```
# chccwdev -e <FCP-devno>
```

or use the command line tool `zfcplib`:

```
# zfcplib <FCP-devno> 1
```

4.) Use either the graphical user interface YAST2 from SUSE to configure the SCSI devices, or use the command line tools and `zfcplib`:

```
# zfcplib <FCP-devno> <port> <SCSI device> 1
```

For more details, see *SUSE Linux Enterprise Server 12 Administration Guide*, available from

https://www.suse.com/documentation/sles-12/pdfdoc/book_sle_admin/book_sle_admin.pdf

Or, see *SUSE Linux Enterprise Server 15 Administration Guide*, available from

https://www.suse.com/documentation/sles-15/pdfdoc/book_sle_admin/book_sle_admin.pdf

3.3.3 Multipathing

In Linux on Z it is mandatory to have multiple, completely disjoint paths to one SCSI disk due to the following reasons:

- Redundancy: to avoid single points of failure

Getting started with IBM Spectrum Scale for Linux on IBM Z

- Performance: to spread I/O requests across multiple paths
- Serviceability: to ensure that I/O can continue to run through other paths when one path is in maintenance mode

The coalescing of the paths to one disk is done by the kernel and the multipath daemon. As soon as the multipath daemon is started, the paths are coalesced to a new logical, multipathed device that is used for any further tasks.

Do not use the `/dev/sd*` device nodes. These nodes represent a single path to a target device.

Proceed with the following steps to set up your SCSI storage with multipathing.

- 1.) Define all the paths to a SCSI device and activate the configuration as shown in the previous chapter.
- 2.) Enable multipathing and start the multipath daemon.

If you are using [Red Hat Enterprise Linux](#):

Use the multipath configuration tool with the following syntax:

```
# mpathconf --user_friendly_names y --with_multipathd y --enable
```

If you are using [SUSE Linux Enterprise Server](#):

Enable the multipath daemon and then start the daemon using the following commands:

```
# systemctl enable multipathd.service
# systemctl start multipathd.service
```

- 3.) Check your multipath configuration with the following command.

```
# multipath -ll
```

```
[command output:
mpatha (36005076303ffc52a0000000000000560f) 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
  |- 0:0:14:1074741334 sde 8:64 active ready running
  `- 1:0:1:1074741334 sdf 8:80 active ready running]
```

As shown in the example above, the command response should show the pathgroups and the active SCSI devices.

3.3.4 Hints and tips

3.3.4.1 Blacklisted devices

If you have defined your multipath devices and started the multipath daemon, but the multipath user-space tool does not show any multipath devices, check the multipath configuration file `/etc/multipath.conf`.

The `/etc/multipath.conf` file contains a blacklist section where all devices that should not be managed by the multipath daemon are listed. Check in the blacklist section whether your multipath devices are blacklisted. No `/dev/sd*` or `/dev/mapper/*` devices should be blacklisted.

```
# cat /etc/multipath.conf
blacklist {
    devnode  "^(ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9]*"
    devnode  "^hd[a-z] [[0-9]*]"
    devnode  "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
    devnode  "^dasd[a-z]+[0-9]*"
}
```

As soon as you have changed the multipath configuration file, you must restart the multipath daemon with the following command:

```
# systemctl restart multipathd.service
```

3.3.4.2 SCSI disk path failover

In Linux on Z it is mandatory to have multiple paths to one SCSI disk. To ensure a path failover to an alternate path in a timely manner in the event of a path loss, change the multipath configuration file `/etc/multipath.conf`. This is important to ensure that Spectrum Scale can handle a path loss and subsequent path failover properly.

Set the following parameters in the multipath configuration file:

- features "0"
- failback immediate
- no_path_retry fail

The `/etc/multipath.conf` file contains a multipaths section where all multipath devices can be managed. Add a section entry for each multipath device as shown below.

```
# cat /etc/multipath.conf
multipaths {
    multipath {
        wwid          36005076303ffc52a0000000000000560f
        alias         mpatha
    }
}
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

```
        features          "0"  
        failback          immediate  
        no_path_retry     fail  
    }  
}
```

As soon as you have changed the multipath configuration file, you must restart the multipath daemon with the following command:

```
# systemctl restart multipathd.service
```

Finally, check your multipath configuration with the following command:

```
# multipath -ll
```

```
[command output:  
mpatha (36005076303ffc52a0000000000000560f) dm-0 IBM          ,2107900  
size=5.0G features='0' hwhandler='0' wp=rw  
`-+- policy='service-time 0' prio=50 status=active  
  |- 0:0:14:1074741334 sde 8:64 active ready running  
  `-- 1:0:1:1074741334 sdf 8:80 active ready running]
```

As shown in the example above, the command response should show the features of the multipath device. The feature "queue_if_no_path" must not be shown any longer.

Note: If you change the multipath configuration after you have created your Spectrum Scale cluster, you need to apply Spectrum Scale post processing steps. Please refer to the Frequently Asked Questions (Question 4.6 What considerations are there when setting up DM-MP multipath service?):

<http://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.html>

4 Network considerations

Spectrum Scale uses an IP network to connect all of the cluster nodes. This is typically a LAN with sufficient bandwidth (minimum 100MB/sec bandwidth) available for Spectrum Scale control traffic. Linux on Z cluster nodes within one IBM Z server can communicate with each other through HiperSockets™ devices. HiperSockets devices provide a fast and very reliable IP network connection. By using a HiperSockets Bridge Port configuration you can extend HiperSockets communication across multiple mainframes.

As the network in an NSD network-attached environment is heavily used for disk I/O (for each read, write, or token request), a very high-speed network with low latency should be used to achieve the best performance. Consider a mix of Gigabit Ethernet and HiperSockets devices.

5 Memory considerations

When you plan for a Spectrum Scale cluster, keep in mind that Spectrum Scale itself needs a considerable amount of memory. Most of the Spectrum Scale memory usage is for the Spectrum Scale page pool, but that is not the only memory used by Spectrum Scale. The page pool value should be one-third to one half of the memory on the node. As a result, the minimum memory size for a Linux instance should be 4 GB.

The actual used memory depends highly on the other workload running on a cluster node. When defining the Spectrum Scale page pool size, you must take into consideration other functions that might be running on the node, for example applications such as IBM WebSphere® Application Server or IBM WebSphere MQ®.

The pagepool value can be seen with the following command:

```
# mmlsconfig pagepool  
  
[command output:  
pagepool 1G]
```

To change the size of the pagepool, use the `mmchconfig` command.

For more details, see *Spectrum Scale Command and Programming Reference*, available from

https://www.ibm.com/support/knowledgecenter/STXKQY_5.1.0/com.ibm.spectrum.scale.v5r10.doc/pdf/

6 Setting up IBM Spectrum Scale

6.1 Prerequisites

The table shows the supported minimum Linux distributions for Spectrum Scale for Linux on Z V5.1.0, available in the Standard Edition and Data Management (Advanced) Edition.

| Distribution | Minimum Level | Kernel |
|---------------------------------|--|---|
| Red Hat Enterprise Linux 7 | Red Hat Enterprise Linux 7.7 Red Hat Enterprise Linux 7.9 | 3.10.0-1062 3.10.0-1160 |
| Red Hat Enterprise Linux 8 | Red Hat Enterprise Linux 8.1 Red Hat Enterprise Linux 8.2 Red Hat Enterprise Linux 8.3 | 4.18.0-147 4.18.0-193 4.18.0-240 |
| SUSE Linux Enterprise Server 15 | SUSE Linux Enterprise Server 15 SP1 SUSE Linux Enterprise Server 15 SP2 | 4.12.14-197.15-default 5.3.18-22-default |

Note: SUSE Linux Enterprise Server 12, Ubuntu Server 16 and Ubuntu Server 18 are no longer

Getting started with IBM Spectrum Scale for Linux on IBM Z

supported beginning with Spectrum Scale on Z V5.1.0. If you have to use Spectrum Scale versions prior to V5.1.0, for example V5.0.5, please refer to the Frequently Asked Questions:

<http://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.html>

6.2 Additional kernel parameters

If you are using Red Hat Enterprise Linux Version 7:

To prepare a Linux on Z instance as a Spectrum Scale cluster node, add additional kernel parameters to `/etc/zipl.conf`.

```
# cat /etc/zipl.conf
Parameters = "... vmalloc=4096G"
```

Changing the boot loader configuration file `/etc/zipl.conf` always implies that the new boot configuration must be activated by using the `zipl` command.

```
# zipl -V
```

Then reboot your Linux on Z instance.

If you are using Red Hat Enterprise Linux Version 8:

To prepare a Linux on Z instance as a Spectrum Scale cluster node, use the `grubby` utility to add additional kernel parameters for all boot loader configuration files on your node as shown in the following example.

```
# grubby --update-kernel=ALL --args="vmalloc=4096G"
```

Changing the boot loader configuration always implies that the new boot configuration must be activated by using the `zipl` command.

```
# zipl -V
```

Then reboot your Linux on Z instance.

If you are using SUSE Linux Enterprise Server:

To prepare a Linux on Z instance as a Spectrum Scale cluster node, add an additional kernel parameter to `/etc/default/grub`.

```
# cat /etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT=" ... vmalloc=4096G "
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

Changing the boot loader configuration file `/etc/default/grub` always implies that the new boot configuration must be activated by using the following command.

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Then reboot your Linux on Z instance.

6.3 Communication without a password

Spectrum Scale requires password-less root access between all nodes in the Spectrum Scale cluster to issue Spectrum Scale administration commands. Spectrum Scale requires both a remote copy and remote shell command with password-less root access. For this purpose Spectrum Scale supports `rcp/rsh` by default, but can be configured to support `scp/ssh`, or any remote copy/shell commands that support the same command line interface as `rcp/rsh`. Generally `scp/ssh` are used because they offer a more secure environment.

You can restrict the use of password-less root access via `scp/ssh` to the network used in the cluster for Spectrum Scale inter-node communication by editing the `sshd` configuration file. You might find the article at this link helpful for making these changes:

ibm.com/developerworks/aix/library/au-aix-modifying-ssh-configuration/

As of IBM Spectrum Scale for Linux on Z V4.2.1, avoid allowing remote root logins to your Spectrum Scale nodes by using use sudo wrapper scripts to run Spectrum Scale administration commands.

For more details about how to set up Spectrum Scale without remote root login, see the *Spectrum Scale Administration Reference*, available from

ibm.com/support/knowledgecenter/STXKQY_5.0.3/com.ibm.spectrum.scale.v5r03.doc/pdf

What now follows are the steps necessary to configure password-less root access for Spectrum Scale using the `scp/ssh` commands.

1.) Login with the root-user ID. An SSH authorized key must be generated on every cluster node that is used for administration purposes and must be placed in the `/root/.ssh/authorized_keys` file.

2.) Check whether a SSH authorized key, in this case RSA, already exists:

```
# cd .ssh
# ls id_rsa.pub
```

If an RSA key is already created, proceed to step 3. Otherwise generate the SSH authorized key in the same directory: (Press Enter at the prompts for the passphrase).

```
# ssh-keygen -t rsa
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

This process creates a public key, which is placed in the `.ssh/id_rsa.pub` file.

3.) Copy the entire contents of the file to the `authorized_keys` file of the directory `/root/.ssh`.

```
# cat id_rsa.pub >> authorized_keys
```

4.) You will need to log in to the current host and each remote host with SSH. Be sure to include the local system.

When you log in, you will be prompted for a “yes” or “no” to accept the RSA fingerprint of the remote host. After you have verified the host key, reply “yes”.

```
# ssh localhost
# ssh node1
# ssh node1.domain
```

5.) Copy the `id_rsa.pub` file to all other nodes.

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub root@node2
```

Now you should be able to log in to each remote host from `node1` with SSH from the command line without a prompt for the RSA fingerprint and without a prompt for a password.

6.4 System time coordination in an IBM Spectrum Scale cluster

Because all the nodes in a Spectrum Scale cluster work on common data, it is critical that time in the cluster is coordinated across the cluster. To synchronize the time in the cluster, you can use NTP, STP, or other network time coordination functions.

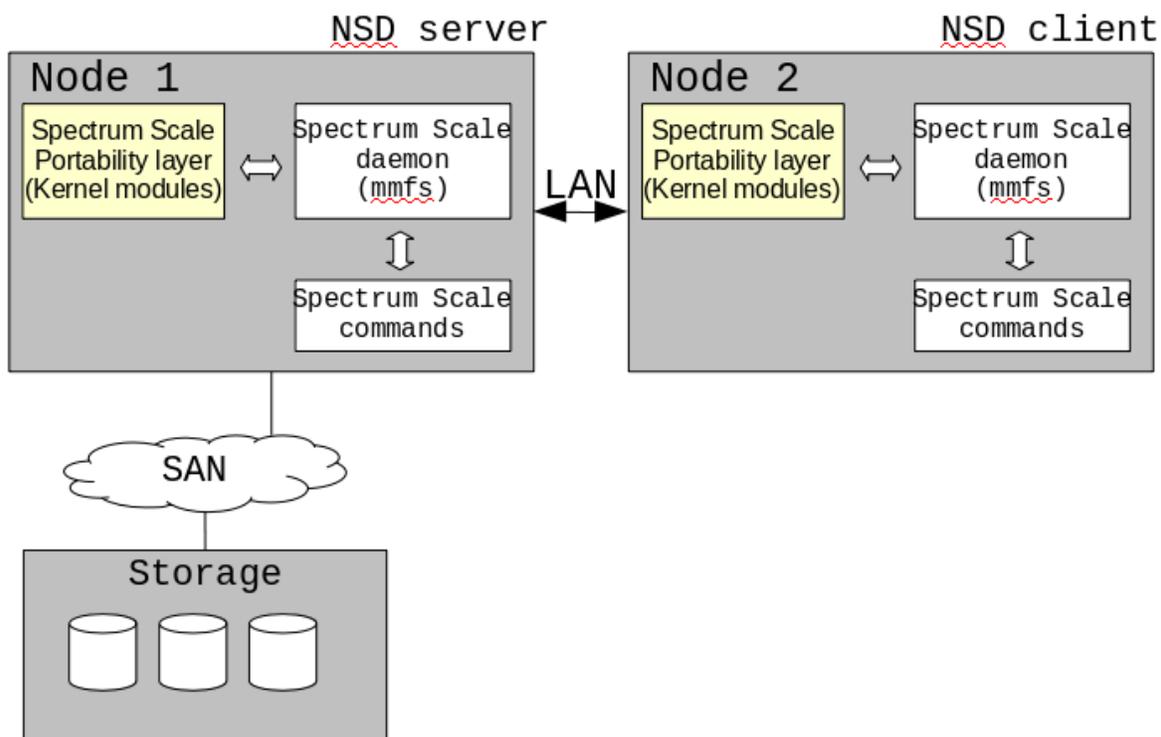
6.5 Structure of IBM Spectrum Scale components

Each cluster node has the following three basic components:

- Spectrum Scale (formerly GPFS) administration commands: to manage the cluster, file system, etc.
- Spectrum Scale (formerly GPFS) daemon: to perform all I/O operations and buffer management for Spectrum Scale
- Spectrum Scale (formerly GPFS) open source portability layer: includes loadable kernel modules that enables the Spectrum Scale daemon to interact with the Linux kernel

During the installation, you can build the portability layer on one Linux instance, for example, on a development system, and install the created package on your other Linux instances.

Note: You need to rebuild and reinstall the portability layer if your Linux kernel version changes, for example, after a system update.



The structure of IBM Spectrum Scale

6.6 Installing the IBM Spectrum Scale packages and building the IBM Spectrum Scale kernel modules

The Spectrum Scale product packages must be installed on all nodes of the cluster independently of the later use of the cluster node. All package names are prefixed with `gpfs.*`, for example, `gpfs.*.rpm` and can be found `/usr/lpp/mmfs/5.1.0.0`.

1) Install the `gpfs.*` packages dependent on the Spectrum Scale edition and the desired features. After the installation has successfully completed, the Spectrum Scale product files can be found in `/usr/lpp/mmfs`

2) Build the Spectrum Scale portability layer. The layer includes kernel modules, which allow the Spectrum Scale daemon to interact with Linux on Z. Be aware that the Spectrum Scale portability layer has to be rebuilt and installed every time the Linux kernel or Spectrum Scale is updated.

Build the Spectrum Scale portability layer with the `mmbuildgp1` command:

```
# /usr/lpp/mmfs/bin/mmbuildgp1
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

You must perform this build step on each node in the cluster. Alternatively, if all the cluster nodes are running the same version of Linux, you can use `mmbuildgpl` to generate the `gpfs.gplbin` package after successful compilation of the Spectrum Scale portability layer. Then, install the generated `gpfs.gplbin` package on the other cluster nodes. To build the RPM package, issue:

```
# /usr/lpp/mmfs/bin/mmbuildgpl -build-package
```

3) After all Spectrum Scale packages including the Spectrum Scale portability layer are installed on all cluster nodes, reboot all the cluster nodes.

Note: To avoid working with a direct path, add the binary library directory of Spectrum Scale to the `PATH` environment variable of your shell profile, for example the bash profile `.bash_profile` in the `/root` directory. Then save the shell profile. For example:

```
export PATH=$PATH:/usr/lpp/mmfs/bin
```

6.7 Planning the IBM Spectrum Scale cluster

When you plan for a Spectrum Scale cluster, you have to consider two aspects:

- The cluster nodes and their roles
- The data disks used with the Spectrum Scale file system

The cluster nodes and their roles can be described in a node descriptor file that is later used to create the cluster. This node descriptor file contains a list of every node to be included in your cluster. Each node is listed on its own line and must be specified in the following format:

```
NodeName:NodeType:AdminNodeName
```

For example:

```
node1:manager-quorum
node2:manager-quorum
node3:quorum
node4:
```

This example shows that there are four nodes in the cluster:

- node1 and node2 have both file system manager and token manager roles and are used to derive the quorum
- node3 is only counted as a quorum node
- node4 is a client node, which is the default

Getting started with IBM Spectrum Scale for Linux on IBM Z

The data disks used with the Spectrum Scale file system are described in the NSD stanza file. The file provides information about each disk in your cluster. You use this file to define the usage of each NSD of your Spectrum Scale file system. Each NSD entry must be specified in the following format:

```
%nsd: device=<device interface name>
      nsd=<alias>
      servers=<NSD server list>
      usage=<data and/or metadata>
```

For example:

```
%nsd: device=/dev/mapper/mpatha
      nsd=disk1
      servers=node1,node2
      usage=dataAndMetadata
%nsd: device=/dev/mapper/mpathb
      nsd=disk2
      servers=node1,node2
      usage=dataAndMetadata
%nsd: device=/dev/dasda1
      nsd=disk3
      servers=node1,node2
      usage=dataOnly
```

This example creates three NSDs, which can be used to create a Spectrum Scale file system.

- Disk1 and disk2 are multipath SCSI devices that can be accessed by the NSD cluster servers node1 and node2 and can store both data and metadata.
- Disk3 is an ECKD device that can be accessed by the NSD cluster servers node1 and node2 and can only store data – when you use an ECKD device in your Spectrum Scale file system, use the partition as NSD rather than the whole device.

The planning and the setup of the node descriptor file and the NSD stanza file need to be done only once on one of the cluster nodes.

6.8 Creating the IBM Spectrum Scale cluster, Network Shared Disks, and an IBM Spectrum Scale file system

After you have finished the planning for the Spectrum Scale cluster, you can start creating the Spectrum Scale cluster and your Spectrum Scale file system by completing the following steps:

1.) Create the Spectrum Scale cluster by using the `mmcrcluster` command:

```
# mmcrcluster -N <nodes.file> -C <cluster1> -r /usr/bin/ssh
-R /usr/bin/scp -A
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

```
[command output:
mmcrcluster -N nodes.file -C cluster1 -r /usr/bin/ssh -R /usr/bin/scp -A
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Warning: Not all nodes have proper GPFS license designations.
    Use the mmchlicense command to designate licenses as needed.
mmcrcluster: Propagating the cluster configuration data to all
    affected nodes. This is an asynchronous process.]
```

where the `-A` option starts the Spectrum Scale daemons automatically when the nodes come up.

2.) Check the creation of the Spectrum Scale cluster using the `mmlscluster` command:

```
# mmlscluster
```

```
[command output:
GPFS cluster information
```

```
=====
```

```
GPFS cluster name:      cluster1
GPFS cluster id:        18000255686092070264
GPFS UID domain:        cluster1.domain.com
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        CCR
```

| Node | Daemon node name | IP address | Admin node name | Designation |
|------|------------------|-------------|-------------------|----------------|
| 1 | node1.domain.com | 10.20.80.86 | node1.domain.com | quorum-manager |
| 2 | node2.domain.com | 10.20.80.87 | node2.domain.com | quorum-manager |
| 3 | node3.domain.com | 10.20.80.88 | node3.domain.com | quorum |
| 4 | node4.domain.com | 10.20.80.89 | node4.domain.com] | |

3.) Each node in the Spectrum Scale cluster must possess a license. Set the type of the Spectrum Scale license associated with the node according to the functional roles that the node has been designated to perform.

```
# mmchlicense server --accept -N node1,node2,node3
# mmchlicense client --accept -N node4
```

```
[command output:
The following nodes will be designated as possessing GPFS server licenses:
    node1.domain.com
    node2.domain.com
    node3.domain.com
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all affected
nodes. This is an asynchronous process.]
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

Note: A Spectrum Scale client node acting as an NSD server requires a server license.

4.) To check the license assignment, use the command `mmlslicense`.

```
# mmlslicense
```

```
[command output:
```

```
Summary information
```

```
-----  
Number of nodes defined in the cluster:           4  
Number of nodes with server license designation:  3  
Number of nodes with client license designation:  1  
Number of nodes still requiring server license designation: 0  
Number of nodes still requiring client license designation: 0  
This node runs IBM Spectrum Scale Standard Edition]
```

5.) Start the Spectrum Scale cluster on all nodes:

```
# mmstartup -a
```

```
[command output:
```

```
Tue Oct 21 17:31:58 CEST 2014: mmstartup: Starting GPFS ...]
```

where the `-a` option causes Spectrum Scale to be started on all nodes in the Spectrum Scale cluster.

6.) To get information about the status of the Spectrum Scale cluster, use the `mmgetstate` command:

```
# mmgetstate -a
```

```
[command output:
```

```
Node number  Node name      GPFS state  
-----  
1            node1         active  
2            node2         active  
3            node3         active  
4            node4         active]
```

where the `-a` option obtains the state information from all nodes in the Spectrum Scale cluster.

7.) Prepare each physical disk you intend to use with Spectrum Scale by first defining it as a Network Shared Disk (NSD) using the `mmcrnsd` command. This command writes the NSD layer to the disk. This step is independent of the architectural model you use, whether the disks are directly attached to a cluster node or are accessed through an NSD client/server configuration. The `mmcrnsd` command expects as input an NSD stanza file (for an NSD stanza file example, see 6.7)

```
# mmcrnsd -F <nsd.file>
```

```
[command output:
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

```
mmlcrnsd -F nsd.file
mmlcrnsd: Processing disk /dev/mapper/mpatha
mmlcrnsd: Processing disk /dev/mapper/mpathb
mmlcrnsd: Processing disk /dev/dasda1
mmlcrnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.]
```

where “-F <NSD stanza file>” specifies the file containing the NSD stanzas for the disks to be created.

Note: After successful completion, the NSD stanza file is modified for use by the next command `mmlcrfs`, which creates the file system.

8.) Check the creation of the Network Shared Disks by using the `mmlsnsd` command:

```
# mmlsnsd
```

[command output:

| File system | Disk name | NSD servers |
|-------------|-----------|------------------------------------|
| (free disk) | disk1 | node1.domain.com,node2.domain.com |
| (free disk) | disk2 | node1.domain.com,node2.domain.com |
| (free disk) | disk3 | node1.domain.com,node2.domain.com] |

9.) As soon as the Network Shared Disks are created, you can create the Spectrum Scale file system by using the `mmlcrfs` command.

```
# mmlcrfs <file system name> -F <nsd.file> -T /mnt -A yes
```

[command output:

```
mmlcrfs gpfs1 -F nsd.file -T /mnt -A yes
The following disks of gpfs1 will be formatted on node node1:
  disk1: size 10000 MB
  disk2: size 10000 MB
  disk3: size 7043 MB
Formatting file system ...
Disks up to size 101 GB can be added to storage pool system.
Creating Inode File
Creating Allocation Maps
Creating Log Files
Clearing Inode Allocation Map
Clearing Block Allocation Map
Formatting Allocation Map for storage pool system
Completed creation of file system /dev/gpfs1.
mmlcrfs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.]
```

where:

Getting started with IBM Spectrum Scale for Linux on IBM Z

-F <NSD stanza file>: specifies the file containing the NSD stanzas for the disks to be created

-A option: indicates that the Spectrum Scale file system will be mounted on each cluster node as soon as the Spectrum Scale daemon is started on the node.

10.) Finally, mount all the Spectrum Scale file systems in all nodes of the cluster using the `mmmount` command

```
# mmmount all -a
```

```
[Command output:  
mmmount: Mounting file systems ...]
```

where:

`all` option: mounts all Spectrum Scale file systems

`-a` option: causes Spectrum Scale to execute the command on all nodes in the Spectrum Scale cluster

11.) Check the file systems that are mounted by using the command `mmismount`.

```
# mmismount all -C all
```

```
[command output:  
File system gpfs1 is mounted on 4 nodes.]
```

where:

`all` option: lists all mounted Spectrum Scale file systems for a particular Spectrum Scale cluster

`-C` option: specifies the clusters for which mount information is requested.

For more details about how to install Spectrum Scale, see *Spectrum Scale Concepts, Planning and Installation Guide*, available from

https://www.ibm.com/support/knowledgecenter/STXKQY_5.1.0/com.ibm.spectrum.scale.v5r10.doc/pdf/

6.9 Hints and tips

6.9.1 Cluster node connection issues

Symptom: Connecting to the node fails with error messages similar to the following:

```
CCR: failed to connect to node 10.0.0.1:1191 (sock 3 err 79)  
node1.domain: authentication refresh for node 2 (node2.domain) failed (err 79)  
node1.domain: authentication initialization failed (err 79):
```

Getting started with IBM Spectrum Scale for Linux on IBM Z

Action: Try one of the following:

- Enable the port used for communication between the cluster nodes – the default port is 1191
- Disable the firewall.

6.9.2 Start of the IBM Spectrum Scale daemon fails

Symptom: The Spectrum Scale daemon fails to start. Check the message log file `/var/log/messages` of the failing cluster node for the following message:

```
GPFS mmfslinux for Linux on z Systems requires a region-second kernel address space. Use kernel parameter vmalloc=4096G to set it up.
```

Action: Check the kernel parameter in the boot loader configuration file. If you did not specify the `vmalloc` kernel parameter, change the boot loader configuration file accordingly, run the `zipl` command to update your boot configuration, and reboot Linux.

7 References

The following publications have been updated with information on Spectrum Scale for Linux on Z and are available from the web:

- IBM Spectrum Scale Concepts, Planning, and Installation Guide (GC28-3161)
- IBM Spectrum Scale Administration Reference (SC28-3162)
- IBM Spectrum Scale Command and Programming Reference (SC28-3164)
- IBM Spectrum Scale Problem Determination Guide (SC28-3163)

To access the above Spectrum Scale documentation, refer to the IBM Knowledge Center website. This website contains the most recent Spectrum Scale documentation in PDF and HTML formats and is located at

ibm.com/support/knowledgecenter/STXKQY/ibmspectrumscale_welcome.html

IBM Knowledge Center is a worldwide central repository of IBM technical publications hosted in a single application, located at a single URL

ibm.com/support/knowledgecenter

The Frequently Asked Questions provides you with the most up-to-date information on topics including how to order IBM Spectrum Scale, supported platforms and supported configuration sizes and capacities

<http://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.html>

Getting started with IBM Spectrum Scale for Linux on IBM Z

Note: this FAQ is updated regularly; read it before you make any system upgrades or major configuration changes to your IBM Spectrum Scale cluster.

You can find and navigate technical content more efficiently and easily with improved search and filtering. Create your own collections of IBM documents with PDF output on demand.

Red Hat Enterprise Linux publications

The principal source of documentation for Red Hat distributions is Red Hat at

https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.

IBM provides several Linux on Z publications for Red Hat Enterprise Linux, available on developerWorks®.

- Device Drivers, Features, and Commands on Red Hat Enterprise Linux
- Using the Dump Tools on Red Hat Enterprise Linux

You can find the latest versions of these publications on the IBM Knowledge Center at

ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_rh.html

SUSE Linux Enterprise Server publications

The principal source of documentation for SUSE Linux Enterprise Server is SUSE Linux GmbH at

<https://www.suse.com/documentation>

IBM provides several Linux on Z publications for SUSE Linux Enterprise Server, available on developerWorks.

- Device Drivers, Features, and Commands on SUSE Linux Enterprise Server
- Using the Dump Tools on SUSE Linux Enterprise Server
- Kernel Messages on SUSE Linux Enterprise Server

You can find the latest versions of these publications on the IBM Knowledge Center at

ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_suse.html

Getting started with IBM Spectrum Scale for Linux on IBM Z



©Copyright IBM Corporation 2020

IBM Systems

Route 100

Somers, New York 10589

U.S.A.

Produced in the United States of America,
12/2020

IBM, IBM logo, ECKD, GPFS, HiperSockets, MQ, POWER, WebSphere, z/OS, z/VM, z Systems and Z are trademarks or registered trademarks of the International Business Machines Corporation.

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.

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.

InfiniBand and InfiniBand Trade Association are registered trademarks of the InfiniBand Trade Association.

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.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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.

OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the [OpenStack website](#).

TEALEAF is a registered trademark of Tealeaf, an IBM Company.

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

Worklight is a trademark or registered trademark of Worklight, an IBM Company.

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

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates. It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.