Troubleshooting
Linux on z Systems and LinuxONE

Troubleshooting
Before using this document, be sure to read the information in "Notices" on page 37.

This edition applies to all Linux distributions that are supported on z Systems and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2012, 2017.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Summary of changes ........................................... v
SC34-2612-05 .................................................. v
SC34-2612-04 .................................................. v
SC34-2612-03 .................................................. v

About this document ........................................... vii

Chapter 1. Troubleshooting for Linux on z Systems .... 1
Techniques for troubleshooting Linux on z Systems problems ........................................... 1
Troubleshooting checklist ........................................ 3
Collecting data for general Linux on z Systems problems ........................................... 4
Collecting data for performance problems ................................................ 4
Collecting data for network problems ........................................... 5
Collecting data for hung system problems ........................................... 5
Collecting data for middleware problems ........................................... 5

Chapter 2. Tools for troubleshooting ....................... 7
Assumptions ...................................................... 7
Authority ......................................................... 7
sysfs and procfs .............................................. 7
debugfs ......................................................... 7
General tools ..................................................... 7
dbginfo.sh - Collect information for debugging ........................................... 7
supportconfig - SUSE Linux Enterprise Server troubleshooting ........................................... 8
sosreport - Generate debugging information for Red Hat Enterprise Linux systems ........................................... 9
Performance tools .............................................. 10
sadc - System activity data collector ........................................... 10
iostat - Monitor input/output device load ........................................... 13
z/VM MONWRITE - Collect CP *MONITOR data ........................................... 14
Collecting data using DASD statistics ........................................... 14
Displaying DASD performance data ........................................... 15
ziomon - Collect FCP performance data ........................................... 16
ziorep - Create FCP performance report ........................................... 18
Obtaining QDIO performance statistics ........................................... 21
Special tools ..................................................... 21
s390dBF traces - Use the kernel debug feature ........................................... 21
scsi_logging_level - Use the SCSI logging feature ........................................... 23
top - See resource usage ........................................... 24
ps - Report a snapshot of the current processes ........................................... 25
netstat - Show information about the Linux networking subsystem ........................................... 25
tcpdump - Collect traffic information for a network interface ........................................... 26
oprofile - Profiling of all running code on Linux systems ........................................... 26
Dump tools ......................................................... 26

Chapter 3. Contacting IBM Support ......................... 29

Chapter 4. Exchanging information with IBM ............ 31
Sending information to IBM Technical Support ........................................... 31
Receiving information from IBM Support ........................................... 31

© Copyright IBM Corp. 2012, 2017
Summary of changes

Changes to the troubleshooting information for the latest releases are listed.

SC34-2612-05

This edition contains the following changes compared to SC34-2612-04.

New information

- Information on how to detect if a Linux instance has been moved to another guest virtual machine or LPAR is included, see “How to detect guest relocation,” on page 33.

Changed information

- None

This revision also includes maintenance and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Deleted information

- None

SC34-2612-04

This edition contains the following changes compared to SC34-2612-03.

New information

- Information for Ubuntu 16.04 was added.
- An example of how to stop the SCSI logging feature has been added.

Changed information

- None

This revision also includes maintenance and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Deleted information

- None

SC34-2612-03

This edition contains the following changes compared to SC34-2612-02.

New information

- You can use the SCSI logging feature to debug SCSI problems, see “scsi_logging_level - Use the SCSI logging feature” on page 23.
- You can use the kdump feature to create kernel dumps, see “Dump tools” on page 26.
Changed information

- Information for Red Hat Enterprise Linux 7 and SUSE Linux Enterprise Server 12 was added.

This revision also includes maintenance and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Deleted information

- “Collecting data using SCSI statistics” has become obsolete and was removed.
About this document

This document describes troubleshooting of Linux for z Systems™ instances. It contains troubleshooting checklists, describes what tools to use for what problem, as well as how to contact IBM® support and transfer log files.

As of July 2017, IBM z Systems is re-branded to IBM Z. In this document, Linux on z Systems refer to Linux running on LinuxONE or an IBM mainframe, including IBM Z and zSeries in 64- and 31-bit mode.

Unless stated otherwise, all z/VM® related information in this document assumes a current z/VM version, see [www.vm.ibm.com/techinfo/lpmigr/vmleos.html](http://www.vm.ibm.com/techinfo/lpmigr/vmleos.html).


Your Linux distribution might provide utilities for working with mainframe devices that are not described in this publication. See the documentation that is provided with your distribution to find out what utilities you can use.

For Linux on z Systems documents that are adapted to a particular distribution, see one of the following web pages:


Other Linux on z Systems publications

You can find Linux on z Systems publications on developerWorks® and on the IBM Knowledge Center.

These publications are available on developerWorks at [www.ibm.com/developerworks/linux/linux390/documentation_dev.html](http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html)

- Device Drivers, Features, and Commands, SC33-8411
- Using the Dump Tools, SC33-8412
- How to Improve Performance with PAV, SC33-8414
- How to use FC-attached SCSI devices with Linux on z Systems, SC33-8413
- How to use Execute-in-Place Technology with Linux on z/VM, SC34-2594
- How to Set up a Terminal Server Environment on z/VM, SC34-2596
- Kernel Messages, SC34-2599
- libica Programmer’s Reference, SC34-2602
- Secure Key Solution with the Common Cryptographic Architecture Application Programmer’s Guide, SC33-8294
- Exploiting Enterprise PKCS #11 using openCryptoki, SC34-2713
- Linux on z Systems Troubleshooting, SC34-2612
These publications are available on the IBM Knowledge Center at

www.ibm.com/support/knowledgecenter/linuxonibm/lnz_r_lib.html

- *libica Programmer’s Reference, SC34-2602*
- *Secure Key Solution with the Common Cryptographic Architecture Application Programmer’s Guide, SC33-8294*
- *Exploiting Enterprise PKCS #11 using openCryptoki, SC34-2713*
- *Linux on z Systems Troubleshooting, SC34-2612*
- *Kernel Messages, SC34-2599*
Chapter 1. Troubleshooting for Linux on z Systems

To isolate and resolve problems with Linux on z Systems, you can use the troubleshooting information. This information contains instructions for using the problem-determination resources that are provided with Linux on z Systems.

Techniques for troubleshooting Linux on z Systems problems

Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem. Certain common techniques can help with the task of troubleshooting.

The first step in the troubleshooting process is to describe the problem completely. Problem descriptions help you and the IBM technical-support representative know where to start to find the cause of the problem. This step includes asking yourself basic questions:

• What are the symptoms of the problem?
• Where does the problem occur?
• When does the problem occur?
• Under which conditions does the problem occur?
• Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, which can then lead you to a problem resolution.

What are the symptoms of the problem?

When starting to describe a problem, the most obvious question is “What is the problem?” This question might seem straightforward; however, you can break it down into several more-focused questions that create a more descriptive picture of the problem. These questions can include:

• Who, or what, is reporting the problem?
• What are the error codes and messages?
• How does the system fail? For example, is it a loop, hang, crash, performance degradation, or incorrect result?

Where does the problem occur?

Determining where the problem originates is not always easy, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few of the components to consider when you are investigating problems.

The following questions help you to focus on where the problem occurs to isolate the problem layer:

• Is the problem specific to one platform or operating system, or is it common across multiple platforms or operating systems?
• Is the current environment and configuration supported?
• Do all users have the problem?
• (For multi-site installations.) Do all sites have the problem?

If one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is understanding the environment in which it exists. Take some time to completely describe the problem environment, including the operating system and version, all corresponding software and versions, and hardware information. Confirm that you are running within an environment that is a supported configuration; many problems can be traced back to incompatible levels of software that are not intended to run together or have not been fully tested together.

**When does the problem occur?**

Develop a detailed timeline of events leading up to a failure, especially for those cases that are one-time occurrences. You can most easily develop a timeline by working backward: Start at the time an error was reported (as precisely as possible, even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log.

To develop a detailed timeline of events, answer these questions:
• Does the problem happen only at a certain time of day or night?
• How often does the problem happen?
• What sequence of events leads up to the time that the problem is reported?
• Does the problem happen after an environment change, such as upgrading or installing software or hardware?

Responding to these types of questions can give you a frame of reference in which to investigate the problem.

**Under which conditions does the problem occur?**

Knowing which systems and applications are running at the time that a problem occurs is an important part of troubleshooting. These questions about your environment can help you to identify the root cause of the problem:
• Does the problem always occur when the same task is being performed?
• Does a certain sequence of events need to happen for the problem to occur?
• Do any other applications fail at the same time?

Answering these types of questions can help you explain the environment in which the problem occurs and correlate any dependencies. Remember that just because multiple problems might have occurred around the same time, the problems are not necessarily related.

**Can the problem be reproduced?**

From a troubleshooting standpoint, the ideal problem is one that can be reproduced. Typically, when a problem can be reproduced you have a larger set of tools or procedures at your disposal to help you investigate. Consequently, problems that you can reproduce are often easier to debug and solve.

However, problems that you can reproduce can have a disadvantage: If the problem is of significant business impact, you do not want it to recur. If possible,
re-create the problem in a test or development environment, which typically offers you more flexibility and control during your investigation.

- Can the problem be re-created on a test system?
- Are multiple users or applications encountering the same type of problem?
- Can the problem be re-created by running a single command, a set of commands, or a particular application?

### Troubleshooting checklist

When you report a problem, provide as much information as possible about the circumstances.

Answering the following questions can help you or IBM support to determine the cause for problems that occur with Linux on z Systems:

1. **How does the problem manifest itself? What are the symptoms?**
   - When this problem occurs, is a specific error message or error code issued?
   - Is trace output of the operation available?

2. **How long has the problem been occurring?**
   - Is it a first time occurrence? When did it happen? (Date and time help to analyze the logs.)
   - How frequently does it occur?
   - Is there any pattern?

3. **If the problem occurred subsequent to some period of normal operation, did anything change in the environment?**
   - Was an operating system patch applied?
   - Did the network environment change? For example, was a server moved or a domain migrated?
   - Did the system recently fail or abnormally terminate?

4. **If you know (for example, based on message prefixes or error codes), where does the problem occur? On one or more systems, production or test environment?**

5. **Can you reproduce the problem on a test system (so that you do not negatively affect the production environment)? What steps are required to reproduce the problem?**

6. **How many users are impacted?**
   - Does this problem affect one, some, or all users?
   - Does the problem occur only for a user who was recently added to the environment, such as a new employee?
   - Do differences exist between the users who are affected and the users who are not affected?

7. **How many applications or business processes are impacted?**
   - Does this problem affect one, some, or all applications or business processes?
   - Does the problem occur only for a new application or business process?
   - Do differences exist between the applications or business processes that are affected and the applications or business processes that are not affected by the problem?

In your report, describe the server and storage infrastructure in as much detail as possible:

- Machine setup, for example, IBM z13™ or IBM z13s.
- Storage server, for example, DS8000®.
- Storage attachment, for example FICON®, or FCP.
- Disk configuration.
- Network, for example OSA (type, mode), HiperSockets™.
- Network topologies.
- Middleware setup (databases, web servers, SAP, or Tivoli Storage Manager. Include version information, if relevant).

You can now collect additional diagnostic data that is required for an IBM technical-support representative to effectively troubleshoot the problem.

**Collecting data for general Linux on z Systems problems**

Collect diagnostic data when a problem occurs. Then submit the diagnostic data to IBM Support. Whatever the problem, start with this general collection of data.

**About this task**

Collecting data before opening a problem management record (PMR) can help you to answer the following questions:
- Do the symptoms match any known problems? If so, has a fix or workaround been published?
- Can the problem be identified and resolved without a code fix?
- When does the problem occur?

The diagnostic data that you collect, and the sources from which you collect that data, are dependent on the type of problem that you are investigating. A base set of information is typically always required. For specific symptoms, you might need to collect additional problem-specific data.

When you submit a problem to IBM Support, you must provide a base set of information.

**Procedure**

To collect general diagnostic data:
1. Collect the base set of diagnostic information by using `dbginfo.sh`.
2. Depending on your distribution, also collect distribution-specific information:
   - On SUSE Linux Enterprise Server, run `supportconfig`.
   - On Red Hat Enterprise Linux, run `sosreport`.

**Collecting data for performance problems**

If performance is a problem, collect diagnostic data that you can use to diagnose and resolve the problem.

**Procedure**

To collect diagnostic data for performance diagnostics:
1. Start sadc (System Activity Data Collection) and provide sar files.
2. If running as guest under z/VM, collect z/VM MONWRITE data.
3. Attach the data files to the opened problem report.
Collecting data for network problems

If the network has a problem, collect diagnostic data that you can use to diagnose and resolve the problem.

**Procedure**

To collect diagnostic data for network diagnostics:
1. Provide a diagram of your network setup.
2. Use `netstat` to collect diagnostic data.
3. Attach the data files to the opened problem report.

Collecting data for hung system problems

If the system hangs, collect diagnostic data that you can use to diagnose and resolve the problem.

**Procedure**

To collect diagnostic data for hung system diagnostics:
1. Create a kernel dump.
2. Include `system.map`, `kerntypes`, `vmlinux (text)` and `vmlinux (debug)` for SUSE Linux Enterprise Server and `vmlinux (full)` for Red Hat Enterprise Linux and Ubuntu.

Collecting data for middleware problems

If middleware is the problem, collect diagnostic data that you can use to diagnose and resolve the problem.

**Procedure**

To collect data for problems with any middleware product (for example, databases):
1. Contact the product support organization.
2. Collect the appropriate debug data as instructed.
3. Attach the data files to the opened problem report.
Chapter 2. Tools for troubleshooting

A variety of troubleshooting tools are available to help you diagnose and resolve problems for Linux on z Systems. Assumptions that are used for all tools are summarized here.

Assumptions

You need the correct authorization to use the commands. The examples assume that the file system is set up in a certain way.

Authority

Most of the tasks described require a user with root authority.

In particular, writing to procfs, and writing to most of the described sysfs attributes requires root authority.

Throughout, it is assumed that you have root authority.

sysfs and procfs

Most of the tasks described assume certain mount points for the file systems.

The mount point for the virtual Linux file system sysfs is assumed to be /sys. Correspondingly, the mount point for procfs is assumed to be /proc.

debugfs

It is assumed that debugfs is mounted at /sys/kernel/debug.

To mount debugfs, you can use this command:

```
# mount none -t debugfs /sys/kernel/debug
```

To mount debugfs persistently, add the following to /etc/fstab:

```
debugfs /sys/kernel/debug debugfs auto 0 0
```

General tools

Tools that can be used in most cases when debugging Linux on z Systems problems.

dbginfo.sh - Collect information for debugging

The dbginfo.sh script collects various system-related files for debugging purposes. It captures the current system environment and generates a .tar file.

Note: The dbginfo.sh script also collects current debugging and trace information, which is saved in the virtual Linux file system sysfs. This information is not persistent across reboots. You must run the dbginfo.sh script before a Linux system is rebooted to avoid losing important debugging and trace information.
If the Linux system runs as z/VM guest operating system, dbginfo.sh also collects information about the z/VM guest setup.

The dbginfo.sh script is part of the s390-tools package in SUSE Linux Enterprise Server and Ubuntu, and the s390-utils package in Red Hat Enterprise Linux.

The service and development team continuously improve dbginfo.sh. You can download the latest version from the developerWorks website at http://www.ibm.com/developerworks/linux/linux390/s390-tools.html. The dbginfo.sh script is included in the s390-tools .tar file.

**Authorization**

- Running the script requires root authority.
- For z/VM guest operating systems you require privilege class B.

**Syntax**

```bash
$ dbginfo.sh
```

**Example**

To generate a diagnostic report with dbginfo.sh, issue `dbginfo.sh`. The output is similar to the following:

```
[root@system]# dbginfo.sh
dbginfo.sh: Debug information script version <version>
Copyright IBM Corp. 2002, 2016
Hardware platform = s390x
Kernel version = <version> (<version>)
Runtime environment = z/VM
1 of 7: Collecting command output
2 of 7: Collecting z/VM command output
3 of 7: Collecting procs
4 of 7: Collecting sysfs
5 of 7: Collecting log files
6 of 7: Collecting config files
7 of 7: Collecting osa oat output
Finalizing: Creating archive with collected data
Collected data was saved to:
Review the collected data before sending to your service organization.
```

**supportconfig - SUSE Linux Enterprise Server troubleshooting**

The `supportconfig` script gathers system troubleshooting information on SUSE Linux Enterprise Server systems. It captures the current system environment and generates a tar-archive.

The script file collects complementary information to the dbginfo.sh script. The `supportconfig` script is part of the Supportutils package.

**Authorization**

Running the script requires root authority.
**Syntax**

See the `supportconfig` man page for more details.

```
supportconfig
```

**Example**

To run `supportconfig`, issue:

```
root@system:~ # supportconfig
```

**Output**

The script produces a tar ball. The location of the tar ball is given in the script output:

```
=============================================================================  
Support Utilities - Supportconfig  
  Script Version: <version>  
  Script Date: 2017 06 01  
=============================================================================  
Gathering system information  
  Data Directory: /var/log/nts_<linux_sys>_170711_1621  
Basic Server Health Check...  
  Done  
[...]  
Creating Tar Ball  
==[ DONE ]===================================================================
Log file tar ball: /var/log/nts_<linux_sys>_170711_1621.tbz  
Log file size: 633K  
Log file md5sum: 453709e61ce570076b6ddfcd0327d110  
=============================================================================  
```

**sosreport - Generate debugging information for Red Hat Enterprise Linux systems**

The `sosreport` script gathers system troubleshooting information. It captures the current system environment and generates a tar file.

The script file collects complementary information to the `dbginfo.sh` script. The `sosreport` script is part of the `support-utils` package.

**Authorization**

Running the script requires root authority.

**Syntax**

See the `sosreport` man page for details.

```
sosreport
```
Example

To run sosreport, issue the command:

```
[root@system]# sosreport
```

Output

The script produces a .tar file. The location of the .tar file is given in the script output:

```
[root@system]# sosreport
sosreport (version 3.3)
This command will collect diagnostic and configuration information
[...]
No changes will be made to system configuration.
Press ENTER to continue, or CTRL-C to quit.
Please enter your first initial and last name [linux.com]: ABC
Please enter the case number that you are generating this report for: DEF
[...]
Creating compressed archive...
Your sosreport has been generated and saved in:
/var/tmp/sosreport-SW.1-20170711144958.tar.xz
[...]
```

Performance tools

Tools that can be used when debugging Linux on z Systems performance problems.

**sadc - System activity data collector**

The **sadc** command samples system data a specified number of times at a specified interval measured in seconds. It writes to the specified output file or the standard output in binary format. The **sadc** command is a backend to the **sar** command.

Data about, for example, the following areas is captured:

- CPU utilization
- Disk I/O overview and on device level
- Network I/O and errors on device level
- Memory usage and swapping

The tools report statistics data over time and create average values for each item.

**Starting sadc/sar as a service**

Start sadc/sar by using the sysstat service. When started as a service, the data files are written to the `/var/log/sa` directory. The files are named `sa<dd>` and `sar<dd>` respectively, where `<dd>` is the current day’s two-digit date. Both files are constantly updated during the day.

**Procedure**

To start the `sadc` command as a service:

Start the sysstat service.
- On Red Hat Enterprise Linux 7: To start the sysstat service with Red Hat Enterprise Linux 7 as a permanent service that persists across reboots, issue:

```
systemctl enable sysstat.service
```

To check that the service is started at boot time and to check that the service is active, issue:

```
systemctl status sysstat.service
```

If the service is not active, issue the following command to start it:

```
systemctl start sysstat.service
```

- On Red Hat Enterprise Linux 5 or 6: To start the sysstat service with Red Hat Enterprise Linux 5 or 6 as a permanent service that persists across reboots, issue:

```
chkconfig sysstat on
```

To check that the service is started at boot time, issue:

```
chkconfig --list |grep sysstat
```

To check that the service is up and running, use the following command:

```
service sysstat status
```

If the service is not running, issue the following command to start it:

```
service sysstat start
```

- On SUSE Linux Enterprise Server 12: You must configure and activate the service using YaST to make data collection persistent across reboots. Alternatively, use the `systemct1` command.

To ensure that the service is started at boot time, issue:

```
systemct1 enable sysstat.service
```

To check that the service is started at boot time and to check that the service is active, issue:

```
systemct1 status sysstat.service
```

If the service is not active, issue the following command to start it:

```
systemct1 start sysstat.service
```

- On SUSE Linux Enterprise Server 11: Configure the service using YaST to make data collection persistent across reboots.

To start the sysstat service directly, issue:
This is not persistent across reboots.
To check the status of the sysstat service, issue:

```
/etc/init.d/boot.sysstat status
```

- On Ubuntu 16.04: To start the sysstat service with Ubuntu 16.04 as a permanent service that persists across reboots, use the `sysv-rc-conf` command:

```
sysv-rc-conf on
```

To check that the service is started at boot time, issue:

```
sysv-rc-conf --list sysstat
```
To check that the service is up and running, issue:

```
service sysstat status
```
If the service is not running, issue the following command to start it:

```
service sysstat start
```

### Results

To report performance data, include both the sadc and the sar data files with the problem report.

### What to do next

After you collect the appropriate diagnostic data, you can complete the following tasks, as appropriate:
- [Chapter 3, “Contacting IBM Support,” on page 29](#)
- [Chapter 4, “Exchanging information with IBM,” on page 31](#)

### Starting sadc/sar directly

If your problem requires data collection that is not covered by the sar/sadc defaults, you can start the tools manually. Start the tools manually, for example, when you need a smaller sampling interval than the default.

### About this task

The sampling interval depends on the time period during which performance problems are seen. You can use a default sampling interval of 10 minutes. If performance problems occur for a couple of minutes occasionally, shorten the sampling interval to less than a minute.

### Procedure

1. To start the `sadc` command directly, issue a command of the following form:

   ```
   /usr/lib64/sa/sadc [options] [interval [count] > <sadc_outfile>
   ```

   See the `sadc` man page for details.
   For example:
Omit the `count` parameter to let sadc sample data until it is stopped. Use the `-S DISK` option to collect disk statistics. By default sadc does not report disks activity to prevent data files from growing too large.

**Note:** In Ubuntu 16.04, the sadc tool is located in the `/usr/lib/sysstat/sadc` directory. Hence, issue a command of the following form:

```
/usr/lib/sysstat/sadc [options] [interval] [count] > <sadc_outfile>
```

2. Extract data and write records by using the `sar` command. Use a command of the following form:

```
sar -A -f <sadc_outfile> > <sar_outfile>
```

For example:

```
[root@system:]# sar -A -f sadc_outfile > sar_outfile
```

where:

- `-A` reports all the collected statistics.
- `-f` specifies the binary input file.

The `sar` command creates a collection of performance reports from the collected `sadc` data and writes these reports to an output file.

**Results**

To report performance data, include both the sadc and the sar data files with the problem report.

**What to do next**

After you collect the diagnostic data, you can complete the following tasks, as appropriate:

- Chapter 3, “Contacting IBM Support,” on page 29
- Chapter 4, “Exchanging information with IBM,” on page 31

**iostat - Monitor input/output device load**

The `iostat` command monitors system input/output device load by observing the time that the devices are active in relation to their average transfer rates.

The `iostat` report shows:

- CPU utilization
- Device queue information
- Service time
Authorization

Root access is required on Linux operating systems.

Syntax

See the iostat man page for the complete syntax and all options.

![iostat](image)

Parameters

- **-c** Collects CPU statistics.
- **-d** Collects disk statistics.
- **-t** Prints a time stamp for each report.
- **-k** Displays statistics in kilobytes per second instead of blocks per second.
- **-x** Displays extended statistics, if available.

Examples

To generate a report with a sampling interval of 10 seconds, collecting disk statistics in KB per second, including a time stamp, and extended statistics, issue the command:

```
[root@system]# iostat -dkx 10
```

Red Hat Enterprise Linux 4

For disk I/O problems, iostat is preferred over sadc/sar, because the sadc/sar version on these distributions does not include appropriate disk I/O statistics.

z/VM MONWRITE - Collect CP *MONITOR data

If your Linux system runs as a guest operating system under z/VM and encounters performance problems, use the MONWRITE utility and include CP *MONITOR data in the problem report.

The z/VM monitor records are in binary format. Make sure that:

- The records are packed and tersed correctly.
- The record size settings are correct.
- The binary to ASCII conversion is made correctly.

For more information about how to collect and upload z/VM MONWRITE data, see [www.ibm.com/vm/perf/tips/collect.html](http://www.ibm.com/vm/perf/tips/collect.html)

Usage notes

- The sadc and sar files must cover the same time interval as the z/VM MONWRITE data.
- Use the default sampling time interval of 1 minute.
Collecting data using DASD statistics

The DASD statistics kernel function monitors the activities of the DASD device driver and the storage subsystem. It mainly records processing time of I/O operations within a given time interval.

Procedure

To collect diagnostic data by using DASD statistics:

1. Start DASD statistics with the following command:

   ```
   # echo set on > /proc/dasd/statistics
   ```

2. Summarized histogram information is available in `/proc/dasd/statistics`, and can be extracted with the following command:

   ```
   # cat /proc/dasd/statistics
   ```

3. Stop DASD statistics with the following command:

   ```
   # echo set off > /proc/dasd/statistics
   ```

Results

DASD statistics creates a summary for all devices.

An IOCTL interface is available to collect the statistics for individual devices. To get DASD statistics for an individual DASD, use the `tunedasd` command:

```
# tunedasd -P /dev/dasd<xx>
```  

What to do next

After you collect the appropriate diagnostic data, you can complete the following tasks:

- Chapter 3, “Contacting IBM Support,” on page 29
- Chapter 4, “Exchanging information with IBM,” on page 31

Displaying DASD performance data

The `dasdstat` command reports the statistics over time and gather data for individual devices or across all devices. The statistics include performance data about Parallel Access Volume (PAV) and High Performance FICON.

Before you begin

Before you can collect DASD performance statistics, the debug file system must be mounted, see “debugfs” on page 7.

About this task

The `dasdstat` command is available as of Red Hat Enterprise Linux version 6.3, SUSE Linux Enterprise Server 11 SP3, and Ubuntu 16.04. Use `dasdstat` to gather DASD performance statistics and to display them.
Examples

The command can be used to get DASD performance statistics across all devices or for individual ones.

- To start gathering data for a summary of all available DASDs, issue:

  ```
  # dasdstat -e global
  ```

- To start gathering data for a selected device 0.0.b223, issue:

  ```
  # dasdstat -e dasda 0.0.b223
  ```

- To stop gathering data for a single device, issue:

  ```
  # dasdstat -d dasda
  ```

- To reset statistic counters for device 0.0.b223, issue:

  ```
  # dasdstat -r 0.0.b223
  ```

- To read data statistics for all devices and for a single device respectively, issue:

  ```
  # dasdstat global
  # dasdstat dasda
  ```

Instead of using the `dasdstat` command, you can access the raw DASD performance data in debugfs. This data can be used, for example, to write scripts that evaluate dedicated DASD performance data.


What to do next

After you collect the appropriate diagnostic data, you can complete the following tasks:

- Chapter 3, “Contacting IBM Support,” on page 29
- Chapter 4, “Exchanging information with IBM,” on page 31

ziomon - Collect FCP performance data

As of SUSE Linux Enterprise Server 11, Red Hat Enterprise Linux 6, Red Hat Enterprise Linux 5.8, and Ubuntu 16.04, use the `ziomon` tool to gather FCP performance data.

The monitor tool `ziomon` collects information and details about:

- The FCP configuration
- The system I/O traffic through FCP adapters
- The overall I/O latencies, adapter latencies, and fabric latencies
- The usage of the FCP resources

Use the `ziorep` tools to analyze the reports created by `ziomon`. The process is illustrated in Figure 1 on page 17.
Authorization

Root access is required on Linux operating systems.

ziomon syntax

See the ziomon man page for the complete syntax and all options.

►►ziomon◄◄

►►<size limit of output file>◄◄

►►-i <interval>◄◄

Parameters

-i <interval>
  Specifies the elapsed time between writing data to disk in seconds. Defaults to 60 seconds.

-d <duration>
  Specifies the monitoring duration in minutes. Must be a multiple of the interval length.

-l <size limit of output file>
  Defines the upper limit of the output files. Must include one of the suffixes M (megabytes), G (gigabytes), or T (terabytes). This limit is only a tentative value that can be slightly exceeded.

-o <output file>
  Specifies the prefix for the log file, configuration file, and aggregation file.
<device>
Denotes one or more device names that are separated by blanks.

Examples

To generate a diagnostic report for devices /dev/sda and /dev/sdb, issue the command:

```bash
[root@system]# ziomon -i 20 -d 5 -l 50M -o trace_data /dev/sda /dev/sdb
```

Output

The ziomon tool creates two output files in the directory where it was started:

- `<output file>`.cfg holds various configuration data from the system
- `<output file>`.log holds the raw data samples that are taken during the data collection phase in a binary format
- `<output file>`.agg aggregates old sample data when the `.log` file grows larger than the allowed limit, thus freeing the log file for more recent data.

Usage notes

- Needs vmalloc space for each device node and CPU.
- The ziomon tool can be stopped with CTRL+C before the time period expires.

ziorep - Create FCP performance report

After you collect FCP performance data using the ziomon tool, use the ziorep tool to create appropriate FCP performance reports.

Three reporting tools are available as of SUSE Linux Enterprise Server 11, Red Hat Enterprise Linux 6, Red Hat Enterprise Linux 5.8, and Ubuntu 16.04.

- "ziorep_config - Configuration report"
- "ziorep_utilization - Report on utilization details" on page 19
- "ziorep_traffic - Traffic report" on page 20

For more information about the FCP performance reports, see How to use FC-attached SCSI devices with Linux on z Systems, SC33-8413.

For a discussion about using the reports with examples, see the z/Journal article Investigating SCSI Devices on System z: How to Use the Ziomon Utilities to Enhance Performance available from [http://enterprisesystemsmedia.com](http://enterprisesystemsmedia.com)

ziorep_config - Configuration report

Use the ziorep_config to report the configuration of the attached SCSI storage and to visualize the interconnection between the different layers that are involved in the SCSI attachment.

The configuration data is reported according to configuration report type (adapter report, device report, or multipath report).

ziorep_config syntax

See the ziorep_config man page for the complete syntax and all options.
-i or --input <src_file>
   Specifies the configuration file that is created by ziomon as source.

-a or --adapter <device_bus_id>
   Limits the output to the list of FCP devices specified.

-A or --Adapter
   Prints the adapter (FCP device) report, this report is the default.

-D or --Device
   Prints the SCSI device report.

-M or --Map
   Prints the multipath mapper report.

Example

To generate a device report for a specific SCSI device from a previous ziomon run that created the configuration file myfcp.cfg, issue the following command:

[root@system]# ziorep_config -D -t -l 0x4021400f00000000 -i myfcp.cfg

ziorep_utilization - Report on utilization details

Use the ziorep_utilization command to produce a report on the usage of FCP resources.

There are two different reports available. The first report provides information on the physical adapters. This report includes usage statistics of the card’s bus, CPU, and overall utilization.

The second report provides various metrics on the virtual adapter. This report includes statistics on the Queued Direct I/O (QDIO) queue that is used to transfer data between the Linux system and the FCP adapter. Statistics that are given are average and maximum utilization in each interval and the number of instances when the queue was full.

ziorep_utilization syntax

See the ziorep_utilization man page for the complete syntax and all options.

-s or --summary
   shows a summary of the data.
-c or --chpid <chpid>
limits the data the specified FCP adapters. The format is a 2-byte hexadecimal
number. You can specify multiple FCP channels by using multiple -c command
line switches.

Example

To generate a utilization report for the specific physical adapter (CHPID) 50, issue
the following command:

```
[root@system]# ziorep_config -c 50 myutil.log
```

ziorep_traffic - Traffic report

Use the ziorep_traffic command produces a report about the systems I/O traffic
through FCP channels and traffic latency.

There are two reports available, varying by the level of detail:

- The default report shows traffic information on a summary level.
- A report that is limited to certain devices that gives detailed traffic information.

Each device is identified by WWPN or LUN and gives information about I/O rate
and throughput and latencies in the I/O subsystem, channel, and fabric. See the
ziorep_traffic man page for the complete syntax and all options.

ziorep_traffic syntax

See the ziorep_config man page for the complete syntax and all options.

```
ziorep_traffic [-i <time>] [-D] [-C <val>] <out_file>
```

- i <time>or --interval <time>
  Sets the aggregation interval to <time> in seconds. Must be a multiple of the
  interval size of the source data. Set to 0 to aggregate over all data.

- C or --collapse <val>
  Specifies on what level you want to aggregate data. See the ziorep_traffic
  man page for more details about possible aggregation levels.

- D or --Device
  Gives detailed information about the traffic.

Example

To generate a traffic report that is called mytraffic.log for all devices with the
data aggregated to a 60-second interval, issue the following command:

```
[root@system]# ziorep_traffic -i 60 mytraffic.log
```
Obtaining QDIO performance statistics

For SUSE Linux Enterprise Server 11 SP2, Red Hat Enterprise Linux 6.2, and Ubuntu 16.04, use the QDIO performance statistics to obtain information of QDIO devices. These statistics apply to FCP devices and to qeth devices.

About this task

To look at these debug logs use the Linux file system debugfs, which must be mounted. See “debugfs” on page 7 for details.

These statistics are located in `<debugfs_mount>/qdio/<device_bus_id>/statistics` where `<debugfs_mount>` is the mount point for debugfs and `<device_bus_id>` is the bus ID of an FCP or qeth device.

Procedure

- To collect QDIO performance statistics for the device fc00, issue the command:

  ```
  # echo 1 > /sys/kernel/debug/qdio/0.0.fc00/statistics
  ```

- To stop collecting QDIO performance statistics, issue:

  ```
  # echo 0 > /sys/kernel/debug/qdio/0.0.fc00/statistics
  ```

- To read the collected data, issue:

  ```
  # cat /sys/kernel/debug/qdio/0.0.fc00/statistics
  ```

What to do next

After you collect the appropriate diagnostic data, you can complete the following tasks:

- Chapter 3, “Contacting IBM Support,” on page 29
- Chapter 4, “Exchanging information with IBM,” on page 31

Special tools

Tools for special circumstances that can be used when debugging Linux on z Systems problems.

s390dbf traces - Use the kernel debug feature

All device drivers and other kernel components write debug log records. These records are available after a system crash. You can also read and save them on a running system.

To look at these debug logs use the Linux file system debugfs, which must be mounted. See “debugfs” on page 7 for details.

Below the s390dbf directory each registered component is represented by a subdirectory with the name of that component. The subdirectories contain files that represent different views of the debug log. Available views are: hex_ascii, sprintf, flush, pages, and level.

The debug information that is written to the logs depends on the debug level that is set for that log. The debug level ranges from 0 for the least detail to 6 for the
most detail. The default level is 2. Only debug entries with a level that is lower or equal to the actual level are written to the log.

To set or change a debug level, from the s390dbf subdirectory for the component you want to work with, issue:

```
  echo <value> > level
```

**Examples**

- To collect the maximum amount of debug information, issue:

  ```
  echo 6 > level
  ```

- To flush the debug log buffer for the component, issue:

  ```
  echo - > flush
  ```

- The kernel debug feature uses wraparound memory buffers. To increase the buffer size, read it first and then enter a higher value with the following command:

  ```
  echo 10 > pages
  ```
**scsi_logging_level - Use the SCSI logging feature**

Use the `scsi_logging_level` command to create, set, or get the SCSI logging level.

The SCSI logging feature prints SCSI messages of the various log areas to the kernel message log file `/var/log/messages`. This feature is used to get information about:

- Issues with the LUN discovery.
- Issues with SCSI error handling and recovery, for example, problems with dirty fibers.

**scsi_logging_level syntax**

See the `scsi_logging_level` man page for the complete syntax and all options.

```plaintext
scsi_logging_level [-a <level>] [-E <level>] [-T <level>] [-S <level>] [-M <level>] --mlqueue <level> --mlcomplete <level> -L <level> --llqueue <level> --llcomplete <level> -H <level> --hlqueue <level> --hlcomplete <level> -I <level>
```

**Parameters:**

- `-a` or `--all <level>`
  specifies value for all SCSI_LOG fields.

- `-E` or `--error <level>`
  specifies SCSI_LOG_ERROR.

- `-T` or `--timeout <level>`
  specifies SCSI_LOG_TIMEOUT.

- `-S` or `--scan <level>`
  specifies SCSI_LOG_SCAN.

- `-M` or `--midlevel <level>`
  specifies SCSI_LOG_MLQUEUE and SCSI_LOG_MLCOMPLETE.

- `--mlqueue <level>`
  specifies SCSI_LOG_MLQUEUE.

- `--mlcomplete <level>`
  specifies SCSI_LOG_MLCOMPLETE.

- `-L` or `--lowlevel <level>`
  specifies SCSI_LOG_LLQUEUE and SCSI_LOG_LLCOMPLETE.

- `--llqueue <level>`
  specifies SCSI_LOG_LLQUEUE.

- `--llcomplete <level>`
  specifies SCSI_LOG_LLCOMPLETE.
--llcomplete <level>
specifies SCSI_LOG_LLCOMPLETE.

-H or --highlevel <level>
specifies SCSI_LOG_HLQUEUE and SCSI_LOG_HLCOMPLETE.

--hlqueue <level>
specifies SCSI_LOG_HLQUEUE.

--hlcomplete <level>
specifies SCSI_LOG_HLCOMPLETE.

-I or --ioctl <level>
specifies SCSI_LOG_IOCTL.

-s or --set
creates and sets the logging level as specified on the command line.

g or --get
gets the current logging level.

c or --create
creates the logging level as specified on the command line.

-v or --version
displays version information.

-h or --help
displays help text.

You can specify several SCSI_LOG fields by using several options. When multiple options specify the same SCSI_LOG field, the most specific option has precedence.

Examples

• To display the logging word of the SCSI logging feature and each logging level, use the following command:

```bash
# scsi_logging_level -g
```

• To set an appropriate log level for the most important log areas with only negligible impact on the performance, use the following command:

```bash
# scsi_logging_level --hlcomplete 1 -T 7 -E 5 -S 7 -I 0 -a 0 -s
```

• You can add the SCSI logging level to the kernel boot parameters. To add it, specify for example:

```
"scsi_mod.scsi_logging_level=4605"
```

• To switch off the logging feature, use the following command:

```bash
# scsi_logging_level -s -a 0
```

**top - See resource usage**

The `top` command provides a dynamic, real-time view of a running system and shows resource usage on a thread level. It can show, for example, CPU usage and detailed memory usage.
Syntax

See the top man page for the complete syntax and all options.

Parameters

- -b Writes the output for each interval into a file.
- -d Specifies the delay time interval in seconds.
- -n Indicates that the maximum number of iterations top should produce before it ends.
- -p Limits the output to the specified processes.

In the running top program, use the F key to configure the displayed columns. Use the W key to write the current configuration to ~/.toprc This is the default directory.

Example

To write 180 iterations 1 second apart into a file, issue:

```
[root@system]# top -b -d 1 -n 180 >top.log 2>&1
```

ps - Report a snapshot of the current processes

The ps command gives comprehensive statistics data on process level and reports a snapshot of the current processes.

See the ps man page for the complete syntax and all options.

Example

The following sample command shows every process in an easily readable format:

```
[root@system]# ( DELAY=10; while [ true ]; do echo "*** "`date`; ps -eLo pid,user,%cpu, %mem,wchan:15,nwchan,stat,time,flags,etime,command:50; sleep $DELAY; done ) 1>>psinfo.out 2>&1
```

netstat - Show information about the Linux networking subsystem

The netstat command shows information about the Linux networking subsystem.

In particular netstat shows:

- Summary information of each protocol
- Detailed information about each connection and interface statistics
- Various error states, for example TCP segments retransmitted
- Information about routing tables
See the netstat man page for the complete syntax and all options.

**Example**

The following sample command shows summary statistics for each protocol:

```
[root@system]# netstat -s
```

Where:

- **-s** shows summary statistics that includes the number of incoming and outgoing packages.

**tcpdump - Collect traffic information for a network interface**

The tcpdump network analysis tool dumps traffic collected for a given network interface.

**tcpdump syntax**

See the tcpdump man page for the complete syntax and all options.

```
tcpdump -s <length> -X -i <interface>
```

**Parameters**

- **-s <length>**
  Writes <length> of data from each packet rather than the default 65535 bytes.

- **-X**
  Writes each packet in hexadecimal and in ASCII format.

- **-i <interface>**
  Identifies the network interface.

**Example**

To dump network traffic for interface eth0, issue the command:

```
[root@system]# tcpdump -s 65000 -X -i eth0
```

**oprofile - profiling of all running code on Linux systems**

The oprofile tool offers profiling of all running code on Linux systems, providing various statistics.

For more information, see [http://public.dhe.ibm.com/software/dw/linux390/perf/Linux_system_monitoring.pdf](http://public.dhe.ibm.com/software/dw/linux390/perf/Linux_system_monitoring.pdf)

**Dump tools**

When the system hangs, create a memory dump.

The following dump tools are available:

- The DASD dump tool writes the memory dump directly to a DASD partition. It supports both ECKD™ and FBA DASDs.
• The tape dump tool writes the memory dump directly to an ESCON/FICON tape device.
• The SCSI dump tool writes the memory dump into file system. It is supported for LPAR and as of z/VM 5.4.
• The kdump feature is a two stage dump tool. This feature is made available through a Linux kernel and initial RAM disk that are preloaded in memory, along with a production system. It is available as of Red Hat Enterprise Linux 6.4, SUSE Linux Enterprise Server 11 SP3, and Ubuntu 16.04.
• VMDUMP (for z/VM guest operating systems) writes the memory dump to z/VM spool space (VM reader). VMDUMP uses a dump format specific to z/VM, the dump must be converted. Do not use VMDUMP to dump large VM guests; the dump process is very slow.

For more information about dump tools, see Using the Dump Tools available at [www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_main.html](http://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_main.html)

In particular, the Using the Dump Tools book contains a chapter about handling large dumps that describes how to split large dumps, for example, with makedumpfile.
Chapter 3. Contacting IBM Support

IBM Support provides assistance with product defects, answers FAQs, and helps users resolve problems with the product.

Before you begin

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Support. Before contacting IBM Support, your company or organization must have an active IBM software maintenance agreement (SWMA), and you must be authorized to submit problems to IBM. For information about the types of available support, see the Support portfolio topic in the “Software Support Handbook”.

Procedure

To contact IBM Support about a problem:

1. Define the problem, gather background information, and determine the severity of the problem. For more information, see the Getting IBM support topic in the Software Support Handbook.

2. Gather diagnostic information.

3. Submit the problem to IBM Support in one of the following ways:
   - Using IBM Support Assistant (ISA):
   - Online through the IBM Support Portal. You can open, update, and view all of your service requests from the Service Request portlet on the Service Request page.
   - By phone. For the phone number to call in your region, see the Directory of worldwide contacts web page.

Results

If the problem that you submit is for a software defect, IBM Support creates a software patch. Missing or inaccurate documentation is normally corrected in the next documentation update. The patch is sent to the Linux distributor for inclusion. Whenever possible, IBM Support provides a workaround that you can implement until the patch is available. For a subscription service for Linux operating system software updates, see the Linux support site available at http://www.ibm.com/systems/z/os/linux/support/
Chapter 4. Exchanging information with IBM

To diagnose or identify a problem, you might need to provide IBM Support with data and information from your system. In other cases, IBM Support might provide you with tools or utilities to use for problem determination.

Sending information to IBM Technical Support

To reduce the time that is required to resolve your problem, you can send trace and diagnostic information to IBM Technical Support.

Procedure

To submit diagnostic information to IBM Technical Support:

1. Open a problem management record (PMR).
2. Collect the diagnostic data that you need. Diagnostic data helps reduce the time that it takes to resolve your PMR. See the following topics:
   - “Collecting data for general Linux on z Systems problems” on page 4
   - “Collecting data for performance problems” on page 4
   - “Collecting data for network problems” on page 5
   - “Collecting data for hung system problems” on page 5
   - “Collecting data for middleware problems” on page 5
3. Compress the files by using the .zip or .tar file format.
4. Transfer the files to IBM. You can use one of the following methods to transfer the files to IBM:
   - Standard data-upload methods: FTP and HTTP
     There are two servers available for uploading data:
     - testcase.boulder.ibm.com (US only)
     - ecurep.ibm.com (international)
   - Secure data upload methods: FTPS, SFTP, and HTTPS
   - IBM Support Assistant
   - The Service Request tool
   All of these data exchange methods are explained on the [IBM Technical Support website](http://www.ibm.com/de/support/ecurep/index.html).

Receiving information from IBM Support

Occasionally an IBM technical-support representative might ask you to download diagnostic tools or other files. You can use FTP to download these files.

Before you begin

Ensure that your IBM technical-support representative provided you with the preferred server to use for downloading the files and the exact directory and file names to access.
Procedure

To download files from IBM Support:

1. Use FTP to connect to the site that your IBM technical-support representative provided and log in as anonymous. Use your email address as the password.

2. Change to the appropriate directory:
   a. Change to the /fromibm directory.
      cd fromibm
   b. Change to the directory that your IBM technical-support representative provided.
      cd nameofdirectory

3. Enable binary mode for your session.
   binary

4. Use the get command to download the file that your IBM technical-support representative specified.
   get filename.extension

5. End your FTP session.
   quit
Appendix. How to detect guest relocation

Information about guest relocations are stored in the s390 debug feature (s390dbf). You can access this information in a kernel dump or from a running Linux instance.

About this task

You can detect if a Linux instance has been moved to another guest virtual machine or LPAR. One available mechanism for guest relocation is z/VM Single System Image (SSI).

You can access the s390 debug feature lgr from a live system or with the crash tool from a kernel dump. When the debug feature contains only one entry, no relocation has been detected and the entry identifies the boot virtual guest machine. For each detected relocation one additional entry is written.

Procedure

Choose the method that suits your purpose:

- Use the crash tool to read from a kernel dump. Issue a command of this form:

  ```
  # crash <vmlinux files> <dump file>
  crash> s390dbf lgr hex_ascii
  ```

- Use the cat command on a live system to read the debugfs entry for lgr. Issue a command of this form:

  ```
  # cat /sys/kernel/debug/s390dbf/lgr/hex_ascii
  ```

Example

Assume that one relocation of the guest virtual machine ZVMGUEST has been detected from a z/VM in LPAR VM000A to a z/VM in LPAR VM000B. You can see this in the kernel dump:

```
# crash vmlinux dump
crash> s390dbf lgr hex_ascii
00 01317816806:277332 3 - 00 .. | IBM2817000000000000EAA1402 ... VM000A ... ZVMGUEST
00 01317866806:277332 3 - 00 .. | IBM2817000000000000EAA1402 ... VM000B ... ZVMGUEST
```

Alternatively, you can see such a relocation from a running system:

```
# cat /sys/kernel/debug/s390dbf/lgr/hex_ascii
00 01317816806:277332 3 - 00 .. | IBM2817000000000000EAA1402 ... VM000A ... ZVMGUEST
00 01317866806:277332 3 - 00 .. | IBM2817000000000000EAA1402 ... VM000B ... ZVMGUEST
```

For more information about the complete s390dbf record, see the struct lgr_info definition in the Linux kernel source code.

Accessibility

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Documentation accessibility

The Linux on z Systems and LinuxONE publications are in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when you use the PDF file and want to request a Web-based format for this publication, use the Readers’ Comments form in the back of this publication, send an email to eservdoc@de.ibm.com, or write to:

IBM Deutschland Research & Development GmbH
Information Development
Department 3282
Schoenaicher Strasse 220
71032 Boeblingen
Germany

In the request, be sure to include the publication number and title.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility at

www.ibm.com/able
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

© Copyright IBM Corp. 2012, 2017
This information is for planning purposes only. The information herein is subject to change before the products described become available.

**Trademarks**

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 "Copyright and trademark information" at


Adobe is either a registered trademark or trademark of Adobe Systems Incorporated in the United States, and/or other countries.

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

Special characters
/debug, mount point 7
/proc, mount point 7
/sys, mount point 7
/sys/kernel/debug, mount point 7

A
accessibility 35
assumptions 7
authority
  root 7

C
checklists
  troubleshooting
    Linux on z 3
collecting data for general Linux on z problems 4
commands, Linux
  scsi_logging_level 23
costensions 7

D
DASD dump tool 26
DASD statistics 15
dasdstat
  troubleshooting tool 15
dbginfo.sh 7
diagnostic data
  collecting for hung system problems 5
collecting for middleware problems 5
collecting for network problems 5
collecting for performance problems 4
  starting sadc/sar directly 12
  collecting with sadc 10
  sending to IBM 31
dump tools
  DASD 26
  kdump 26
  SCSI 26
tape 26
  VMDUMP 26

G
general Linux on z problems
  collecting data for 4
guest relocation 33

H
hung system
  troubleshooting
    collecting diagnostic data 5

I
iostats
  troubleshooting tool 13

K
kdump 26

L
Linux on z
  troubleshooting checklist 3

M
middleware
  troubleshooting
    collecting diagnostic data 5
MONWRITE 14
mount point
  debugfs 7
  procs 7
  sysfs 7

N
network
  troubleshooting
    collecting diagnostic data 5

O
oprofile 26

P
performance
  statistics, QDIO 21
performance problems
  collecting data with sadc 10
  collecting diagnostic data 4
problem determination
  exchanging information with IBM Support 31
ps 25

Q
QDIO performance
  statistics 21

R
relocation
  guest 33
  root
  authority 7
Troubleshooting

S
s390dbf 21
sadc
  starting as service 10
troubleshooting tool 10
sar
  starting as service 10
SCSI dump tool 26
scsi_logging_level, Linux command 23
sending data to IBM 31
starting sadc/sar directly
troubleshooting
collecting diagnostic data 12
statistics
  QDIO performance 21
supportconfig 8, 9

T
tape dump tool 26
tcpdump 26
top 25
troubleshooting
checklist
  Linux on z 3
collecting data for general Linux on z problems 4
collecting data for hung system problems 5
collecting data for middleware problems 5
collecting data for network problems 5
collecting data for performance problems
  starting sadc/sar directly 12
collecting data with sadc 10
collecting performance data 4
general tools 7
identifying problems, techniques for 1
Linux on z 1
performance tools 10
sending data to IBM 31
special tools 21
tools 7
  sadc 10
troubleshooting and support
  contacting IBM Support 29
  exchanging information with IBM Support 31
troubleshooting techniques 1
troubleshooting tools
  DASD statistics 15
dasdsym 15
dbginfo.sh 7
dump tools 26
iostats 13
MONWRITE 14
oprofile 26
ps 25
s390dbf 21
supportconfig 8, 9
tcpdump 26
top 25
ziomon 16
ziorep 18
ziorep_config 18
ziorep_traffic 20
ziorep_utilization 19

V
VMDUMP 26

Z
ziomon 16
ziorep 18
ziorep_config 18
ziorep_traffic 20
ziorep_utilization 19
Readers’ Comments — We'd Like to Hear from You

Linux on z Systems and LinuxONE
Troubleshooting

Publication No. SC34-2612-05

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.
Submit your comments using one of these channels:
• Send your comments to the address on the reverse side of this form.
• Send your comments via email to: eservdoc@de.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

Email address
Readers’ Comments — We'd Like to Hear from You

IBM Deutschland Research & Development GmbH
Information Development
Department 3282
Schoenaicher Strasse 220
71032 Boeblingen
Germany