Redbooks
ibm.com/redbooks

Service Procedures for Linux on IBM Power Systems Servers

Luciano Chavez

Brahadambal Srinivasan

Michael Bringmann

Janani Janakiraman

IBM

Redpaper

Introduction

Collecting data on first occurrence of the problem can aid in problem determination and timely resolution of defects. At IBM®, this process of collecting data on first occurrence is often referred to as *First Failure Data Capture* (FFDC). Gathering this data before reporting a defect helps to understand the problem more quickly and thoroughly, which saves time analyzing data and reduces the time and mission affects in fixing defects.

Several diagnostic capabilities are built into the Linux operating system that enable you to determine the application level problems and system level problems. Collecting FFDC logs early, even before opening a defect report, helps to quickly determine whether:

- Symptoms match known problems (rediscovery)
- A report can be identified and resolved as a not-a-defect problem
- A workaround to reduce severity exists

Configuring systems to collect logs

In the ideal scenario, we expect nothing to fail but that expectation should not stop us from preparing for failure. In this section, we discuss how to set up specific tools and configure your system to collect the right logs in the eventuality of a failure.

IBM ServiceReport tool

Download and use the IBM ServiceReport configuration tool to validate and repair the system configuration for First Failure Data Capture.

This plug-in-based framework features different plug-ins that perform different types of validation tasks. For an example, the package plug-in validates whether certain packages are installed on the system.

This tool features the following modes:

- Validate mode: Performs system configuration validation.
- Repair mode: Attempts to auto-correct the system configuration, if possible.

This tool runs in the validate mode by default. It loads all of the validate plug-ins that are applicable to the current system environment and runs them. After all the plug-ins are run, the tool aggregates the result of all the validations that are performed by the applicable plug-ins and generates a validation report to display on the console.

The default report format is simple. It displays the short description of the plug-in and the validation status of that plug-in.

Helpful checks

Consider the following questions before opening the defect and run any other workloads as needed. Narrowing the scope of the defect can be helpful during the diagnosis process:

- Is the problem seen in previous releases (of this distro)?
- Is the problem seen in other distros?
- Is the problem an installation issue? Is it seen when other installation methods are used? For example:
  - If it is a scripted installation, is the problem seen interactively?
  - If it is a netboot installation, is the problem seen by CDROM/DVD installation?
  - If it is storage-related, is the problem seen with other storage adapters or devices? Other storage options, for example, non-LVM, multipath or RAID?
  - If it is network-related, is the problem seen with other network adapters? Are there any differences between IPv4 and IPv6? Static address versus DHCP?

The following utilities are available from Linux products to collect system configuration information and logs to diagnose system problems:

- Console logs
- Sosreport (for RedHat and Ubuntu)
- Supportconfig (for SuSE)
- Apport (for Ubuntu)

In addition to setting up and gathering relevant data, in cases where the system can be booted, collect the results after running these distro-specific utilities. Provide the resulting compressed archive file to the defect report.

Guidelines for logs

This section provides instructions for how to set up or configure the system to save all relevant logs.

Console logs

Critical diagnostics often appear on the console during failures. If those diagnostics are not captured, diagnosis of the problem is severely impeded. It should be standard practice to always run a capture program, such as `screen / script / tmux` on the system through which you connect to the server to collect the console logs during all problem workload runs.

If problems occur, the captured output must be preserved and made available to IBM Support engineers. In addition, the `silent` or `quiet` modes should be removed from the kernel command line, and `printk.time=1` and `loglevel=7` added. If increasing the log levels is too verbose for normal runs, try with a lower log level setting.

If yaboot is used, remove the `silent` or `quiet` and append the options for higher verbosity: `printk.time=1` and `loglevel=7` to the kernel line in `/etc/yaboot.conf` and then, restart.

Complete the following steps to enable higher verbosity in the logs with grub2:

1. Open the /etc/default/grub file by using a text editor.
2. Find the entry for the GRUB_CMDLINE_LINUX_DEFAULT attribute, and replace the silent and quiet modes with splash "printk.time=1 loglevel=7":

   ```
   GRUB_CMDLINE_LINUX_DEFAULT=
   GRUB_CMDLINE_LINUX_DEFAULT="splash  printk.time=1  loglevel=7"
   ```

3. Save the edits to the /etc/default/grub file.
4. Run **update-grub** (on Ubuntu) or **grub2-mkconfig -o /boot/grub2/grub.cfg** (on RHEL) to update the grub2.cfg file.
5. Restart the system.

For more information about the screen utility, see this web page.

For more information about the tmux utility, see the following web pages:

- Sharing Linux Terminal Sessions With Tmux and Screen
- Use tmux for a more powerful terminal
- Tmux Command Examples To Manage Multiple Terminal Sessions
- GitHub.com/tmux/tmux

Configuring other logs

Set up the following Linux configurations to ensure that more information is collected:

- For issues that occur over long durations, provide multiple Linux log files as many distros by enabling logrotate.

  The logrotate utility maintains a log file rotation policy and retains copies of log files to assist in establishing patterns that are related to system usage. Configure the logrotate options in /etc/logrotate.conf file and set up a script for every file that you require to logrotate in the /etc/logrotate.d/syslog file. A cronjob should be set up by using /etc/cron.<frequency>/logrotate; for example: /etc/cron.weekly/logrotate.

- For issues that occurred on a previous boot, it might be necessary to ensure that the Linux kernel logs are persistent.

  For more information about enable persistent journaling, see these resources:

  - SUSE
  - RHEL

- For Dynamic Logical Partitioning (DLPAR) and Live Partition Mobility (LPM) operations to function properly, you must install several system management packages.

  Various tools, packages, and libraries must be installed to properly manage an LPAR from an HMC. For more information about how to install the official packages for the distros, see the following IBM Knowledge Center pages:

  - IBM Linux on Power tools repository
  - Installing the tools repository
  - Installing packages with the tools repository

If your Linux system has a firewall running, ensure that you add firewall rules such that port 657 is open for TCP and UDP from any source (or at least from HMC IP addresses).

Sosreport utility

The **sosreport** command is a tool that collects configuration and diagnostic information from an RHEL Linux system. For example, the running kernel version, loaded modules, and system and service configuration files. The command also runs external programs to collect more information and stores this output in the resulting compressed .tar archive file.

To run sosreport, the sos or sosreport package must be installed. After the package is installed, run the # **sosreport** command.

For more information about Red Hat Enterprise Linux, see this Red Hat publication.

Supportconfig utility

The **supportconfig** command is SUSE's tool for collecting detailed system information and log files. It is similar to the sosreport utility in that it can be run from a shell and it collects system information, logs, and configuration information and creates a compressed .tar archive file that can be supplied to IBM Support.

The **supportconfig** command is provided by the supportutils package, which must be installed. Run the **#supportconfig** command to capture all of the logs.

Apport utility

The **apport** command is Ubuntu's tool for collecting detailed information about the failing component. This utility intercepts program crashes, collects debugging information about the crash and the operating system environment, and sends it to defect trackers in a standardized form.

Although the **apport** command is present by default on the Ubuntu system, ensure that the Whoopsie package is installed before you run **apport**. Otherwise, **apport** cannot upload the crash report to the Ubuntu Launchpad.

In many cases, it is wanted that the system firmware version is reported for crashes, hangs, or other problems. The /proc/device-tree/openprom directory includes useful information about the specific model machine's firmware levels.

For example, the following information is from an IBM POWER9™ server:

```
# cd /proc/device-tree/openprom
      # lsprop *
      ibm,fw-next-bank "T"
      ibm,fw-vernum_encoded
      "FW930.00 (VM930_024)"
      ibm,phandle      fffffffe (-2)
      linux,phandle    00d05c90 (13655184)
      model            "IBM,FW930.00 (VM930_024)"
      name             "openprom"
      relative-addressing
```

Configuring Crash Memory Dump Capture

The system must include a crash memory dump capture (kdump or fadump) setup that is configured. The feature also should be tested to verify it is operating correctly. The ServiceReport tool should help validate and enable kdump on your system.

Configuring kdump

Instructions for enabling or configuring the kdump facility often is documented by the distribution. For example, for RHEL 7 kdump configuration, see Red Hat's Kernel Crash Dump Guide.

For more information about the amount of memory that is required for kdump, see this web page.

See this Ubuntu web page for more information about the amount of memory that is required for kdump.

Firmware Assisted Dump

In this section, Firmware Assisted Dump (FADump) is described.

Overview

FADump uses the following flow:

1. Linux registers with firmware for FADump.
2. Linux crashes.
3. F/W backs up memory and CPU register data and resets hardware, except memory.
4. IPLs with special DT node to indicate active dump.
5. Linux starting process begins.
6. Linux checks for special DT node to find if memory dump is available.
7. Linux exports the memory dump as a special file: /proc/vmcore.
8. Linux system filters the /proc/vmcore file and saves it to disk.
9. Linux invalidates the memory dump and requests for restart.

Configuring FADump on different distros

This section explains how to configure or enable the fadump utility in different distros, such as RHEL, SLES, and Ubuntu:

1. Install and configure KDUMP.

   – RHEL:

   ```
   yum install kexec-tools
   yum install kdump
   yum install crash
   ```

   – SLES:

   ```
   zypper install kexec-tools
   zypper install kdump
   zypper install crash
   ```

- Ubuntu:

  ```
  apt-get install kexec-tools
  apt-get install kdump
  apt-get install crash
  ```

2. Change the kernel command line to pick up `fadump=on`:

   - RHEL:

     Add `fadump=on` to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`:

     ```
     GRUB_CMDLINE_LINUX="crashkernel=4096M fadump=on rd.lvm.lv=rhel_peplp25/root
     rd.lvm.lv=rhel_peplp25/swap"
     ```

   - SLES

     - Edit `/etc/sysconfig/kdump` and make changes as”

       ```
       KDUMP_FADUMP="yes"
       ```

     - Restart kdump service as:

       ```
       systemctl status kdump.service
       ```

       Newer SLES version does provide yast GUI support for configuring FADump:

       ```
       yast kdump
       ```

   - Ubuntu:

     Add `fadump=on` to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`:

     ```
     GRUB_CMDLINE_LINUX="crashkernel=4096M fadump=on rd.lvm.lv=rhel_peplp25/root
     rd.lvm.lv=rhel_peplp25/swap"
     ```

3. Reserve crash memory value.

   Crash kernel value can be setup in any of following methods:

   - `crashkernel=auto`
   - `crashkernel=2G-4G:1024M,4G-32G:2048M,32G-64G:4096M,64G-128G:8192M,128G-:16384M`
   - `crashkernel=1024M`

   > **Note:** The `-crashkernel=auto` option is supported only on RHEL as a best effort estimation.

   For RedHat, confirm that the steps that are described at this web page were used in configuring FADump.

   For SUSE, confirm that the steps that are described at this web page were used in configuring FADump.

   - RHEL

     If you want to specify reserved boot memory instead of accepting the default settings, add `crashkernel=xxM` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`, where *xx* is the amount of the memory that is required in megabytes:

     ```
     GRUB_CMDLINE_LINUX="crashkernel=4096M fadump=on rd.lvm.lv=rhel_peplp25/root
     rd.lvm.lv=rhel_peplp25/swap"
     ```

     > **Note:** Remove the quiet option from the command line.

- SLES

  If you want to specify reserved boot memory instead of accepting the default settings, add `crashkernel=xxM` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`, where *xx* is the amount of the memory that is required in megabytes:

  ```
  GRUB_CMDLINE_LINUX_DEFAULT="splash=silent showopts crashkernel=4096M
  fadump=on "
  ```

  > **Note:** If only the `fadump=on` parameter is passed to the kernel without setting the crashkernel parameter, memory reservation for the fadump kernel defaults to 5% of total memory.
  >
  > Also, remove quiet option from the command line.

- Ubuntu

  If you want to specify reserved boot memory instead of accepting the default settings, add `crashkernel=xxM` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`, where *xx* is the amount of the memory that is required in megabytes:

  ```
  GRUB_CMDLINE_LINUX_DEFAULT="splash=silent showopts crashkernel=4096M
  fadump=on "
  ```

  > **Note:** If `fadump=on` parameter is passed only to the kernel without crashkernel parameter set, memory reservation for the fadump kernel defaults to 5% of total memory.
  >
  > Also, remove quiet option from the command line.

4. Update the boot loader:

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

5. Restart the machine to reflect the changes.
6. Verify that FADump memory is reserved:

   ```
   dmesg | grep "Res"
   ```

   If memory is not reserved, repeat step 2 and restart the machine so that the changes take effect.
7. Check whether FADump is enabled. If not, enable it:

   ```
   cat /sys/kernel/fadump_enabled
   ```

   Whether fadump is enabled (that is, `fadump=on`) is indicated in `/proc/cmdline`. The value should be 1. If the node has a value of 0, FADump is *not* enabled. FADump *must* be enabled.
8. Check whether FADump is registered. If it is not, register it:

   ```
   cat /sys/kernel/fadump_registered
   ```

   Whether FADump is registered is indicated by a value of 1. If it is 0, register FADump by using the following command:

   ```
   systemctl restart kdump.service
   ```

9. Check that FADump is ready to capture dump:

   ```
   systemctl status kdump.service
   ```

   or

   ```
   kdumpctl status
   ```

If all of the steps are successful, we are ready for FADump crash dump capture. If not, repeat the steps.

Run the following command to trigger the crash:

```
echo c > /proc/sysrq-trigger
```

For RedHat, confirm that the steps that are described at this web page were used in configuring FADump.

For SuSE, confirm that the steps at this web page were used in configuring FADump.

For Ubuntu, confirm that the recommended steps at this web page were followed for configuring FADump.

Additional settings

You must also enable the sysrq facility that allows for dumping specific information from the console without the need to issue a dump restart from the Hardware Management Console (HMC). To enable the facility while the system is running, enter the following command at the shell prompt:

```
echo 1 > /proc/sys/kernel/sysrq
```

However, to avoid the need to enter this command each time, edit the /etc/sysctl.conf configuration file by adding the following line to automatically enable sysrq at each start:

```
kernel.sysrq = 1
```

For more information about expectations when submitting a defect report and some basic general requisite information, such as, processor revision, firmware levels, and the operating system level, see “Defect submission guidelines ” on page 8.

Defect submission guidelines

In this section, we present a few helpful tips for providing information to the support engineers.

Defect report template

Use the template that is shown in Figure 1 to report defects.

```
1. Problem Statement:
    a. Problem description:
    b. System down? (Y/N):
    c. Reproducible? (Y/N):
      Details: (for example, Problem recreation steps. Has this ever worked on a
      different system or distro? Is it readily re-creatable on the system where the
      issue has been reported?)
    d. Frequency / Probability of occurrence: (H/M/L/Unknown)
    e. Impact:
    f. What is the current state of the system:
    g. Workaround / Mitigation implemented (if available):

2. System Configuration:
    a. System type:  Baremetal / PowerVM LPAR / KVM Guest /
       PowerNV Host?
    b. Power system details (for example: p8, p9.):
    c. Problem on Host/ KVM Guest/ LPAR?
    d. Build information:
        Guest OS & Kernel Build:
        Host OS & Kernel Build:
        Host Name:
        BMC:
        PNOR:
        OPAL:
        Adapter Firmware:
     Switch details:
     Storage details:
        HMC+VIOS+FSP+LPAR(s):
    e. Workload:
        System in Production (Y/N)?
        What is the workload being run on the system?

3. Debug Data Collected:
    a. Data collected from system (vmcore/ sosreport/other logs):
       sosreport / supportconfig provided (yes/no/na):
       /var/log/messages provided (Y/N):
       Alternatively, ‘journalctl’ output (Y/N):
       console logs provided (Y/N):
       dmesg provided (Y/N):
    b. Dump Config:
       Kdump configured on Host (Y/N):
       Kdump/Virsh Dump available (Y/N):

    c. Make debug data / dump available to IBM Support Engineers
```

Figure 1   Defect report template

Other guidelines

Along with the template, provide the following information where relevant and possible:

- The exact commands with arguments, or workloads that were being run. If the workload procedure uses a GUI, such as the HMC, describe the commands that were selected and the options that were chosen or entered.
- Show the installation of support tools, or, if the installation occurred before login, list the tools that the application depends upon.
- Provide time stamps for commands to aid comparison of events; for example:

  ```
  date; <command>
  ```
- Mention the time stamp at which problem is believed to have occurred, if possible.
- For log output, include only the pertinent bits in the defect report; or, if providing a large file, highlight exactly where the issue is seen (for example, “See log entries around this event or time”). Compressing the log files before uploading to a defect is recommended.
- For DLPAR and LPM to function correctly, you must install several system management packages. Run the following command to show the packages:

  ```
  rpm -qa | grep -i -E "src|rsct|DynamicRM"
  ```

Recommendations for more information to gather for specific classes of problems or hardware configurations are described next.

System crash and hang issues

The system must have a crash dump capture (`kdump` or `fadump`) setup that is configured and tested for its correct working, as described in “Firmware Assisted Dump” on page 5. The crash files should be made available to IBM Support engineers.

Guidelines for crashes

If a system crashes, the entire `/var/crash/127.0.0.1-YYYY-MM-DD-HH:MM:SS` subdirectory should be made available. This subdirectory includes the essential `vmcore-dmesg.txt` file that contains the console output at the time of the crash. The entire directory can be copied by using the following command:

```
scp -r `hostname`:/var/crash/127.0.0.1-YYY-MM-DD-HH:MM:SS
login@destination:/path/to/store
```

Linux installation issues

This information must be included when opening Linux installation-related defects.

The required template for reporting issues that are observed during installation is provided in "Installation issue reporting template" on page 11. This template is required along with the other general "Defect report template" on page 9.

Installation issue reporting template

The template that is shown in Figure 2 must be completed and provided in the defect description.

```
Boot type (installer was booted from):
- CDROM/ISO image
- Network boot
- QEMU direct boot kernel/initrd
- Other (specify):
Bootloader (netboot ony):
- grub
- yaboot
- PXELINUX
- Other (specify):
Bootloader protocol (netboot only):
- TFTP (in most cases)
- HTTP (optional, grub only)
- Unsure
Kernel cmdline used to launch install (if other than CDROM):
- List cmdline or provide full bootloader file, if available.
- HINT: You may be able to interrupt the bootloader at the menu
  screen and get the bootloader to display the kernel cmdline.
Install repository type:
- CDROM
- Local network repository (specify URL):
- Internet mirror (specify URL):
- Other (specify):
Was a response file used during install? (Yes/No):
 * If Yes, provide the file
 * If No, describe install steps
Point of failure:
1. Failed during boot of installer
2. Failed while accessing install media (for example, when you are unable to
access install repository or kickstart file)
3. Other failure during installation (stage 1)
4. Failed to reboot after installation
5. Problem during post-install (stage 2) configuration or other problem seen
after system reboot
```

Figure 2 Installation issue reporting template

Other required information

This section specifies the information that must be collected and made available to IBM Support engineers, depending on the point the failure occurred. Some steps request only that certain conditions be verified, while others provide information that should be made available to IBM Support engineers.

Failed during boot of installer

Provide the console output that shows the failure. For grub, you can add `insmod progress` or `set debug=all` to the `grub.cfg` for more verbose output, as shown in the following example:

```
# /boot/grub/grub.cfg - sample grub2 configuration file

# See the official grub documentation for more information.

set pager=1
set debug=all
```

For more information, see the GNU GRUB Manual 2.04.

If the `netboot` installation method is used, verify that bootloader components are available on the network, especially the bootloader image file and the bootloader config file (menu file). Typical image files are `grub2` or `grub2.arch`, `yaboot`, and `pxelinux.0`.

Typical config files are `grub.cfg`, `yaboot.conf`, `default`, or might feature a name that refers to the MAC or IP of the booting node:

```
   $ curl tftp://server/path/to/bootloader/image > /dev/null
   $ curl tftp://server/path/to/bootloader/bootloader.cfg > /dev/null
```

Also, verify the installer kernel and initrd files. Typical kernel names are `linux`, `vmlinux`, and `vmlinuz`. The initrd names can be `initrd`, `initrd.img`, `initrd.gz`:

```
   $ curl tftp://server/path/to/netboot/vmlinux > /dev/null
   $ curl tftp://server/path/to/netboot/initrd > /dev/null
```

You also can use:

```
   $ tftp <server IP> -c "get /path/to/target/file"
```

Failed while accessing installation media

If you receive errors that indicate that the installation media cannot be accessed, provide the installer logs that show the failure. For example: Unable to access installation repository or `preseed/kickstart/autoyast` file.

For more information, see “Saving debug logs post-installation” on page 15.

Verify that the repository is accessible from your server (see Table 1).

Table 1 Check Distro repository

Distro name	Command

Ubuntu	- Local repo:
	`$ curl http://server/path/to/Ubuntu/ISO/dists/vivid/Release`
	`$ curl http://server/path/to/RHEL/ISO/media.repo`
	`$ curl http://server/path/to/SLES/ISO/media.1/media`
	- Internet repo:
	`$ curl http://ports.ubuntu.com/ubuntu-ports/dists/vivid/Release`
	- Ubuntu specific (the installer does this):
	`$ wget -qO- http://ports.ubuntu.com/ubuntu-ports/dists/vivid/Release | grep -E '^(Suite|Codename):'`
	`$ wget -qO- http://server/path/to/Ubuntu/ISO/dists/vivid/Release | grep -E '^(Suite|Codename):'`

RedHat	Internet repo: `$ curl http://server/path/to/RHEL/ISO/media.repo`
SuSE	Internet repo: `$ curl http://server/path/to/SLES/ISO/media.1/media`

Other failure during installation (stage 1)

Provide the installer logs that show the failure. For more information, see “Saving debug logs post-installation” on page 15.

Provide pertinent screen captures or other installer output, installer backtrace file, and so on.

Failed to restart after installation

Provide the console output that shows the failure.

Attempt to differentiate the types of failures. Some are obvious, such as a kernel panic. Others are difficult to discern, such as when the system starts but does not provide a login prompt. This issue might mean that the system is hung or it might be a console problem (for example, `getty` is not started on that console) and the system can be accessed over the network by way of SSH. Ping the system and attempt to log in by using SSH.

Problem during post-install configuration (stage 2)

This section describes problems that are observed during any `first boot` configuration that is performed by the installer, or any other problem that occurs after system restart.

Provide `sosreport` or `supportconfig` file from the installed system. For more information, see “Guidelines for logs” on page 2

Also, clarify whether the issue is seen on the as-installed system or if any changes were made postinstallation.

Enabling SSH access during installation

This mechanism starts the `sshd` service during installation so that you can connect to the system during the Linux installation by using SSH, monitor the installation progress, or retrieve log files that show the events or failures that occurred during the installation process. For more information about SSH, see the `ssh(1)` man page. The options that are listed in Table 2 are boot options that are to be added to the installer's boot command line.

Table 2 Enable SSH

Distro name	Command

Ubuntu	Add the following code to your kernel cmdline. After the installation starts, you should be able to connect to `ssh installer@node`:
	`anna/choose_modules=network-console network-console/start=continue`
	`network-console/password=passw0rd network-console/password-again=passw0rd`

RedHat	Add the following code to your kernel cmdline. After the installation starts, you should be able to connect to `ssh root@node`: `inst.sshd=1`
SuSE	Add the following code to your kernel cmdline. After the installation starts, you should be able to connect to `ssh root@node`: `UseSSH=1 SSHPassword=passw0rd`

Saving debug logs during installation

If an installation failure occurs, you must extract the log files so that they can be uploaded as part of the defect report process. Some installers might provide this feature directly. However, in most cases, you must connect to the command-line interface of the installer by using SSH and runs commands to upload the files back to another system (see Table 3).

Table 3 Save Debug Logs during installation

Distro name	Command

Ubuntu	From the Ubuntu installer main menu, select **Save debug logs**. A message is displayed that reads, "A simple web server has been started on this computer to serve log files".
	Retrieve the files and provide it to the support engineer:
	`wget http://<node>/hardware-summary`
	`wget http://<node>/syslog`

RedHat	From the RedHat installer (`anaconda`) shell, compress these files and provide the compressed file to the support engineer. You can complete this task by using SSH if you start your installation with `inst.sshd` on the kernel cmdline. You also can complete this task from your system console:
	`(cd /; tar -zcvf /root/installer-logs.tgz ./var/log/ ./tmp/{*conf,*log,*state})`
	`scp -p /root/installer-logs.tgz somenode:/tmp/`

SuSE	From SuSE installer (`autoinst`) shell, save the `Yast2` logs and provide it to the support engineer (for more information, see this web page):
	`save_y2logs /tmp/y2logs.tgz`
	`scp -p /tmp/y2logs.tgz somenode:/tmp/`

Saving debug logs post-installation

The sosreport, supportconfig, or apport utilities that were described in “Guidelines for logs” on page 2 produce archive files of one format or another and are made available to IBM Support engineers as needed.

DLPAR and LPM issues

This section describes information that is required to be collected when you work on issues that are related to DLPAR/Hotplug and Live Partition Mobility (LPM), or RMC connection issues when one of these commands is used. a sosreport

Before reporting a problem, be sure that you are running the latest firmware for the system and adapters that is suitable for your hardware.

In all cases of DLPAR or LPM issues, a sosreport/supportconfig archive that contains the system logs must be collected and made available to IBM Support engineers.

In addition, provide the /var/log/drmgr log file because the sosreport/supportconfig utilities might not provide it. If a /var/log/dmrgr.0 log exists, also provide that log. Finally, provide a full copy of the /var/log/messages file (or journalctl output) for the cases where a series of LPM events occurred over a period.

If the packages are installed and are at the latest levels, provide the output of the **lssrc** command to know the status of the systems management daemons:

```
$lssrc -a
 Subsystem          Group         PID     Status
 ctrmc              rsct         22511    active
 IBM.HostRM         rsct_rm      22631    active
 IBM.ServiceRM      rsct_rm      22666    active
 IBM.MgmtDomainRM   rsct_rm      22706    active
 ctcas              rsct         22731    active
 IBM.ERRM           rsct_rm      22738    active
 IBM.AuditRM        rsct_rm      22739    active
 IBM.SensorRM       rsct_rm      22740    active
 IBM.DRM            rsct_rm      22741    active
```

Commands, such as **lssrc -a** can list some of the RSCT process or daemons to be inactive. It is normal for some of the RMs or daemons, such as ctcas and IBM.HostRM, to be inactive. The RMs or daemons. such as ctrmc, IBM.MgmtDomainRM, and IBM.DRM must be active. Other RMs or daemons are brought online by RMC, if needed.

In addition, capture and provide the output of the commands:

```
lslocks
firewall-cmd --list-ports
ps ax | grep drmgr
```

If you see drmgr in the output of ps ax | grep drmgr, try to dump the call stack for it because this inclusion in the output means it is hung; for example:

```
# ps ax | grep drmgr
…
 7370 ?       SLl     0:53 drmgr …
 …
# cd /proc/7370
# cat stack
 or
# strace -p 7370
```

Include RSCT data collection from client LPARs that are running Linux kernels. Collect the following data:

- The `pedbg` logs from the source Hardware Management Console (HMC) and the destination HMCs if the source and destination CEC are managed by two different HMCs)
- The `snap -a` from all VIOS nodes (of both source and destination CEC)

If the client LPARs are running Linux, run the **`/opt/rsct/bin/ctsnap -xrunrpttr`** command (the snapshot is collected in `/tmp/ctsupt`).

Note: These logs must be collected only if the workload is being run on a distro level that does not include updates to `sosreport/supportconfig` to automatically gather this data.

Storage and I/O Issues

This section lists the required information to be collected if defects are observed while different storage I/O setups and configurations are used.

Multipath issues

When run at the shell prompt, the commands that are described here provide data that is required for working multipath storage-related issues. Much of this information is available already in the archive file that is produced by the corresponding system configuration and diagnostics collection utility. Providing that archive to the support engineer is a good starting point.

The `/etc/multipath.conf` configuration file should be supplied. The presence of this file is a good first indicator of whether multipath was configured. In addition, the output from the commands that are described here should be made available to IBM Support engineers:

```
multipath -ll
dmesg
multipath -v3
dmsetup table
ls -l /dev/mapper
multipathd -k'show config'
multipath -c /dev/sdX
lvm pvdisplay | grep Name
```

For multipath support to work correctly, the following packages must be installed:

```
device-mapper-multipath (RHEL)
multipath-tools (SLES)
multipath-tools and multipath-tools-boot (Ubuntu)
```

With current releases of Linux that us the `systemd init` system, run the commands that are listed in Table 4 on page 17 at the shell prompt (depending on the distribution) to check where `multipathd` is running.

Table 4 Check if multipathd is running

Distro name	Command

RHEL	`systemctl status multipathd.service`
	`service multipathd status`

SLES	`systemctl status multipathd`
Ubuntu	`systemctl status multipath-tools.service`

The following commands are specifically useful when debugging problems with the daemon process `multipathd`. To log to the console (in foreground mode or interactively),:

```
multipathd -d
multipathd -k'show config'
multipath -c /dev/sdX
```

To run the `multipathd` daemon process in the background as usual but provide a high level of verbose logging to `syslog`, run the following command:

```
multipathd -v4
```

To trace system calls that are made by the `multipathd` daemon process, run the following command:

```
strace -f multipathd
```

IBM POWER RAID (ipr) driver

In addition to collecting the required system information with the system configuration and diagnostics collection utility, the `iprutils` package provides a data collection utility of its own that is called used called `iprsos`, which is specific to IPR that can also be used.

In some rare situations, it might be helpful to increase the IPR driver logging level. The valid value for the `log_level` options is 0 - 4. For example, add the following input to the kernel command line to increase the logging level to 3:

```
ipr.log_level=3
```

The trace output can be recovered from the kernel log buffer by using the **`dmesg`** command or, if the system is hung or halted by a kernel crash, the **`xmon`** debugger or **`kdump`**.

Fibre Channel issues

When encountered with Fibre Channel issues, check if the Fibre Channel ports are online in the host. The data can be obtained through `sysfs` by running the **`lspci -nn`** command.

Run the following command to list the adapters and their PCI addresses in the box:

```
ls -ld /sys/bus/pci/drivers/*/*... /sys/bus/pci/drivers/lpfc/0001:02:00.0/
```

This command lists the drivers that are used by each PCI address and thus the adapter.

Run the following commands to list the SCSI/FC host numbers that are associated with each PCI address and thus the adapters:

```
ls -ld /sys/class/scsi_host/host*
ls -ld /sys/class/fc_host/host*
```

Run the following command to check the port status:

```
grep . /sys/class/fc_host/host*/port_state
```

Run the following command to check the port status for storage side:

```
grep . /sys/class/fc_remote_ports/rport-*/port_state
```

Run the following command to list each SD block device, its rport, SCSI address, and major:minor:

```
ls -ld /sys/class/fc_remote_ports/rport-*/device/target*/*/block/sd*/bdi
```

Enabling tracing for the Emulex Fibre Channel (lpfc) driver

To enable more verbose logging that involves the LPFC driver, add the following option to the kernel command line:

```
lpfc.lpfc_log_verbose=0x9a
```

Where, the kernel options for the kernel command line are in the standard format of `<module>.<option>=<value>`.

If you load the module from the shell by using the **`modprobe`** command, the format resembles the following example:

```
modprobe <module>  <option>=<value> <option>=<value>
```

Enabling verbose logging for the QLogic Fibre Channel (qla2xxx) driver

To enable more verbose logging that involves the LPFC driver, add the following input as an option to the kernel command line:

```
qla2xxx.ql2xextended_error_logging=1
```

NVMe

As with other issues, the `sosreport` or `supportconfig` report includes information, such as `lsblk` and `mount` information that often is needed for NVMe issues. Along with that data, issue the following commands and collect the output to be provided to the IBM Support engineers:

```
nvme --version
nvme list
nvme id-ctrl <dev>
nvme fw-log <dev>
nvme id-ns <ns>
```

Serial and USB/xhci

For USB and serial port device issues, provide the following information:

- USB/xhci:

  ```
  lsusb -v
  ```

- For USB installation issues, the tool name and exact command line that are used to generate the bootable USB stick:

  ```
  lsblk
  ```

- Serial:

  ```
  setserial -g <dev>
  ```

- Information about the topology, how were the ports connected, and loopbacks
- Minicom configuration file that is used and the command line that is used to start minicom

IBM virtual SCSI (ibmvscsi) driver

Information that is required for diagnosing virtual Small Computer System Interface (SCSI) issues is similar to the information that is required for the other SCSI drivers. If multipath is involved, collect the information about multipath as described in "Multipath issues" on page 16.

An `sosreport/supportconfig` report covers most of the information that is required. Run the following commands to gather the information:

```
lsscsi
lspath
lsmap -all
lsnports
lsblk
cat /proc/scsi/*
```

LVM

Much the same information is needed for diagnosing LVM-related issues. If multipath is involved, you must collect that information as described in "Multipath issues" on page 16. Again, `sosreport` or `supportconfig` report covers most of what is needed.

Run the following commands as `root` and capture their output to be made available to the IBM Support engineers:

```
cat /proc/partitions
cat /proc/mounts/
cat /etc/lvm/lvm.conf
dmsetup ls
dmsetup table
dmsetup targets
dmsetup version
lvm version
```

Networking issues

This section covers information that is required to be collected when you encounter networking-related issues.

General networking issues

If general networking issues are observed, the following information must be collected:

- Provide `sosreport/supportconfig` data.

  When a KVM setup is used, provide the reports from the host and guest.
- Provide network topology.

- If the failure involves a workload between multiple systems, clearly identify the IP addresses involved (client/server) and any external network equipment that is involved in the communications between those two endpoints.
- Provide specific date and time details of any failures and errors.

Network debug or data gathering suggestions

Collect the output of the following commands to provide more debug information:

- `netstat -s`

  This useful command is run before and after any workloads that involve communication problems or potential packet loss. Ensure to capture the information to be made available to the support engineer; for example:

  ```
  netstat -s > /tmp/netstat-s.$(date '+%s')
  [run failure scenario]
  netstat -s > /tmp/netstat-s.$(date '+%s')
  ```

- `ethtool -S`

  This command is for the network device that is involved (assuming `eth0`, in this example).

  Running this command before and after workloads helps to identify any drops or errors that are specific to the failure timeline; for example:

  ```
  ethtool -S eth0 > /tmp/ethtool-S_eth0.$(date '+%s')
  [run failure scenario]
  ethtool -S eth0 > /tmp/ethtool-S_eth0.$(date '+%s')
  ```

- Network traces (sniffer on the network or tcpdump on the host)

  This debug step often requires information from the `sosreport/supportconfig` and topology to identify the correct interface and network players. When you attempt to track down potential packet loss, run traces on both ends of the connection (that is, the client and server).

  The following example command traces only the traffic between a client and server, and captures only enough of the packet to monitor the connection state and timing (assuming that the command is run on the client):

  ```
  tcpdump -i eth0 -s 92 -w /tmp/client_eth0.cap host {server name or IP}
  ```

InfiniBand issues

If defects are observed while InfiniBand is used, gather the output of the following commands along with the information as described in “General networking issues” on page 19:

- `ibstat`
- `ibstat -p`
- `ibv_devices`
- `ibswitches`
- `iblinkinfo`
- `ibhosts`
- `ibping`

VNIC issues

To ensure that we can gather information for VNIC issues, `kdump/fadump` should be enabled.

Provide information about the network and cabling configuration. If a peer system setup is used, also provide the peer system's information.

Collect the following logs and files:

- `cat /var/log/messages`
- `cat /var/log/warn`
- For SuSE, provide `supportconfig`
- For RHEL, provide `sosreport`
- For Ubuntu, provide `apport`
- `vmcore`, in case of a crash
- Output of `ethtool -S`
- Output of `tcpdump -i <interface> -s 65535 -w <some-file>`

kdump issues

To report issues in getting the kdump functionality working, the following information must be collected and the suitable prerequisite checks must be performed before defect submission.

Provide the `sosreport` (in the case of Red Hat and Ubuntu) and `supportconfig` (in the case of SuSE) along with the following information for problem isolation and resolution:

- To confirm whether crashkernel is used and how much memory is reserved, the output of the following command for crashkernel parameter value

  ```
  cat /proc/cmdline
  ```

- Provide the amount of memory that is reserved for kdump (on all distros)

  ```
  dmesg | grep Reser
  ```

- Provide the config files for kdump to understand the configuration.

> **Note:** Each distro features its own configuration file.

- To check status, start or stop kdump service:
  - Red Hat: `service kdump status/start/stop`
  - SLES: `service kdump status/start/stop`
  - Ubuntu: `kdump-config status/show/load/unload`
- Provide output of the following commands:
  - `free -m`
  - `cat /proc/cpuinfo | grep processor`
  - `ppc64_cpu -info`
  - `ppc64_cpu -smt`
  - `cat /proc/sys/kernel/panic`
  - `cat /proc/sys/kernel/panic_on_oops`
  - `cat /sys/kernel/kexec_crash_loaded`
  - `cat /sys/kernel/kexec_loaded`

- Provide the complete console log at the time of kdump failure. The sosreport/supportconfig does not capture this information.

Consider the following troubleshooting points to ensure that a valid configuration is, in fact, failing:

- If the console shows that the kdump kernel failed because of Out of Memory (OOM), increase crashkernel reservation and retry.

  > **Note:** Sometimes kdump kernel panics even before putting OOM message. First, retry dumping the vmcore with increased crashkernel memory.

- In the case of remote dumps, such as Network FileSystem (NFS)/Secure Shell (SSH) where you are copying the vmcore file to a remote machine, you can have the IP address of the crashed machine as prefix to the directory name. This configuration helps to find the directory easily on the remote machine if you have the IP address:
  - Configure kdump over NFS.

    Update kdump-config show results to check for the NFS entry.
  - List the NFS mount path from /etc/default/kdump-tools path.

    For example, NFS="9.3.111.11:/data/dumps/jumplp4"
  - Collect the output of the **ifconfig -a** or **ip -a** command.
  - In a SuSE system, if you see more than one network interface, such as eth0, and eth1, try the suggestions that are provided at this web page.
- If makedumpfile/kdump encounters with the kernel version, run the **makedumpfile -v** command.

Debugging fadump issues

For defects that are seen in fadump, apart from information that was described in the previous section, the following data also is required from the kernel where crash was triggered:

```
cat /sys/kernel/fadump_enabled
cat /sys/kernel/fadump_registered
```

Virtualization issues

Kernel-based virtual machines (KVM) can also report issues in almost all if the areas that are described in this document. In most cases, the kdump mechanism is sufficient for obtaining a memory dump from a VM after a crash or panic.

However, in some cases, it is necessary to work directly with the hypervisor to obtain a crash dump. Two mechanisms are available with libvirt to achieve a crash dump: pvpanic and virsh dump. Both of these methods are described in the Red Hat document *Virtualization Deployment and Administration Guide*.

For more information about the pvpanic mechanism. see Red Hat's *Virtualization Deployment and Administration Guide - Setting a Panic Device*.

For more information about the **virsh dump** command, see Red Hat's *Virtualization Deployment and Administration Guide - Creating a Dump File of a Domain's Core*.

The following example shows the use of the **virsh dump** command to create a dump of a guest virtual machine whose domain is "my-vm" use:

```
virsh dump my-vm my-vm.dump --memory-only
```

Debug logs

In general, enable more debugging from `libvirt`, and have it added to `messages/journalctl`, or a designated log file. For more information, see this libvirt web page.

virsh commands

For problems with **virsh** commands, set the following environment variables in your command shell:

```
export LIBVIRT_DEBUG=1
export LIBVIRT_LOG_OUTPUTS="1:file:/tmp/virsh.log"
```

Then, run **virsh** to accumulate the logs in the `/tmp/virsh.log` file. This process should be sufficient to at least get a precise idea of what is occurring and where things are going wrong, which helps you to then insert the correct breakpoints when running under a debugger.

For more information about logging information, see this libvirt web page.

Logging in the libvirt daemon

Similarly, the libvirt daemon logging behavior can be tuned by using three configuration variables that are stored in the `/etc/libvirt/libvirtd.conf` configuration file:

- `log_level`: accepts the following values:
  - 4: Only errors
  - 3: Warnings and errors
  - 2: Information, warnings, and errors
  - 1: Debug and everything
- `log_filters`: Defines logging filters
- `log_outputs`: Defines logging outputs

When starting the libvirt daemon, any logging environment variable settings overrides the settings in the config file. Command line options take precedence over all. If no outputs are defined for libvirtd, it attempts to use:

- 0.10.0 or later: systemd journal, if `/run/systemd/journal/socket` exists
- 0.9.0 or later: file `/var/log/libvirt/libvirtd.log` if running as a daemon
- before 0.9.0: syslog if running as a daemon
- all versions: to stderr stream if running in the foreground

Libvirtd does not reload its logging configuration when it is issued a SIGHUP. If you want to reload the configuration, you must perform a service libvirtd restart or manually stop and restart the daemon.

Commonly useful information

Provide the following information about the VM and its execution:

- Steps that are needed to reproduce the problem.
- QEMU command-lines (ps aux | grep qemu) or at least the guest XMLs that are used here.
- Configuration XML for the VM; for example: `virsh dumpxml <VM name>`.
- Console log that shows operations that are performed on the guest VM.
- The `libvirt/qemu` log from the latest problem invocation of the VM after an error is observed.

  These logs can be found in the `/var/log/libvirt/qemu/` directory where the log files are derived from the name of the VM; for example, `virsh start virt-tests-HSR --console` places the log file in cat `/var/log/libvirt/qemu/virt-tests-HSR.log`.

Conclusion

Providing all of the FFDC information at the time of opening the defect helps developers to work on problem isolation and arrive at the root-cause analysis (RCA), which leads to early problem resolution. This process also reduces the likelihood of defects going stale while waiting for submitters to provide the right logs and reduce the number of requests for recreation of the reported problem.

Firmware Assisted Dump

In this section, Firmware Assisted Dump (FADump) is described.

Overview

FADump uses the following flow:

1. Linux registers with firmware for FADump.
2. Linux crashes.
3. F/W backs up memory and CPU register data and resets hardware, except memory.
4. IPLs with special DT node to indicate active dump.
5. Linux starting process begins.
6. Linux checks for special DT node to find if memory dump is available.
7. Linux exports the memory dump as a special file: `/proc/vmcore`.
8. Linux system filters the `/proc/vmcore` file and saves it to disk.
9. Linux invalidates the memory dump and requests for restart.

Configuring FADump on different distros

This section explains how to configure or enable fadump in different distros, such as RHEL, SLES, and Ubuntu:

1. Install and configure KDUMP.
   - RHEL:
     ```
     yum install kexec-tools
     yum install kdump
     yum install crash
     ```
   - SLES:
     ```
     zypper install kexec-tools
     zypper install kdump
     zypper install crash
     ```
   - Ubuntu:
     ```
     apt-get install kexec-tools
     apt-get install kdump
     apt-get install crash
     ```
2. Change the kernel command line to pick up `fadump=on`:
   - RHEL:

     Add `fadump=on` to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`:
     ```
     GRUB_CMDLINE_LINUX="crashkernel=4096M fadump=on rd.lvm.lv=rhel_peplp25/root
     rd.lvm.lv=rhel_peplp25/swap"
     ```
   - SLES
     - Edit `/etc/sysconfig/kdump` and make changes as”
       ```
       KDUMP_FADUMP="yes"
       ```
     - Restart kdump service as:
       ```
       systemctl status kdump.service
       ```
       Newer SLES version does provide yast GUI support for configuring FADump:
       ```
       yast kdump
       ```
   - Ubuntu:

     Add `fadump=on` to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`:
     ```
     GRUB_CMDLINE_LINUX="crashkernel=4096M fadump=on rd.lvm.lv=rhel_peplp25/root
     rd.lvm.lv=rhel_peplp25/swap"
     ```
3. Reserve crash memory value.

   Crash kernel value can be setup in any of following methods:
   - `crashkernel=auto`
   - `crashkernel=2G-4G:1024M,4G-32G:2048M,32G-64G:4096M,64G-128G:8192M,128G-:16384M`
   - `crashkernel=1024M`

   For RedHat, confirm that the steps that are described at this web page were used in configuring FADump.

   For SuSE, confirm that the steps at this web page were used in configuring FADump.

- RHEL

  If you want to specify reserved boot memory instead of accepting the default settings, add `crashkernel=xxM` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`, where *xx* is the amount of the memory that is required in megabytes:

  ```
  GRUB_CMDLINE_LINUX="crashkernel=4096M fadump=on rd.lvm.lv=rhel_peplp25/root
  rd.lvm.lv=rhel_peplp25/swap"
  ```

  > **Note:** Remove the quiet option from the command line.

- SLES

  If you want to specify reserved boot memory instead of accepting the default settings, add `crashkernel=xxM` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`, where *xx* is the amount of the memory that is required in megabytes:

  ```
  GRUB_CMDLINE_LINUX_DEFAULT="splash=silent showopts crashkernel=4096M
  fadump=on "
  ```

  > **Note:** If the `fadump=on` parameter is passed to only the kernel without crashkernel parameter set, memory reservation for the fadump kernel defaults to 5% of total memory.
  >
  > Also, remove quiet option from the command line.

- Ubuntu

  If you want to specify reserved boot memory instead of accepting the default settings, add `crashkernel=xxM` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`, where *xx* is the amount of the memory that is required in megabytes:

  ```
  GRUB_CMDLINE_LINUX_DEFAULT="splash=silent showopts crashkernel=4096M
  fadump=on "
  ```

  > **Note:** If the `fadump=on` parameter is passed to only the kernel without crashkernel parameter set, memory reservation for the fadump kernel defaults to 5% of total memory.
  >
  > Also, remove quiet option from the command line.

4. Update the boot loader:

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

5. Restart the machine to reflect the changes.
6. Check whether FADump is enabled. If not, enable it:

   ```
   cat /sys/kernel/fadump_enabled
   ```

   Whether fadump is enabled (that is, `fadump=on`) is indicated in `/proc/cmdline`. The value should be 1. If the node has a value of 0, FADump is *not* enabled. FADump *must* be enabled.

7. Check whether FADump is registered. If it is not, register it:

   ```
   cat /sys/kernel/fadump_registered
   ```

   Whether FADump is registered is indicated by a value of 1. If it is 0, register FADump by using the following command:

   ```
   systemctl restart kdump.service
   ```

8. Verify FADump memory reserved:

   ```
   dmesg | grep "Res"
   ```

   If memory is not reserved, repeat step 2 and restart the machine so that the changes take effect.

9. Check that FADump is ready to capture dump:

   ```
   systemctl status kdump.service
   ```

   or

   ```
   kdumpctl status
   ```

   If all of the steps are successful, we are ready for FADump crash dump capture. If not, repeat the steps.

   Run the following command to trigger the crash:

   ```
   echo c > /proc/sysrq-trigger
   ```

For RedHat, confirm that the steps that are described at this web page were used in configuring FADump.

For SuSE, confirm that the steps at this web page were used in configuring FADump.

For Ubuntu, confirm that the recommended steps at this web page were followed for configuring FADump.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, 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 http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

Redbooks (logo) ®

IBM®

POWER9™

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

REDP-5611-00

ISBN 0738459062

Printed in U.S.A.