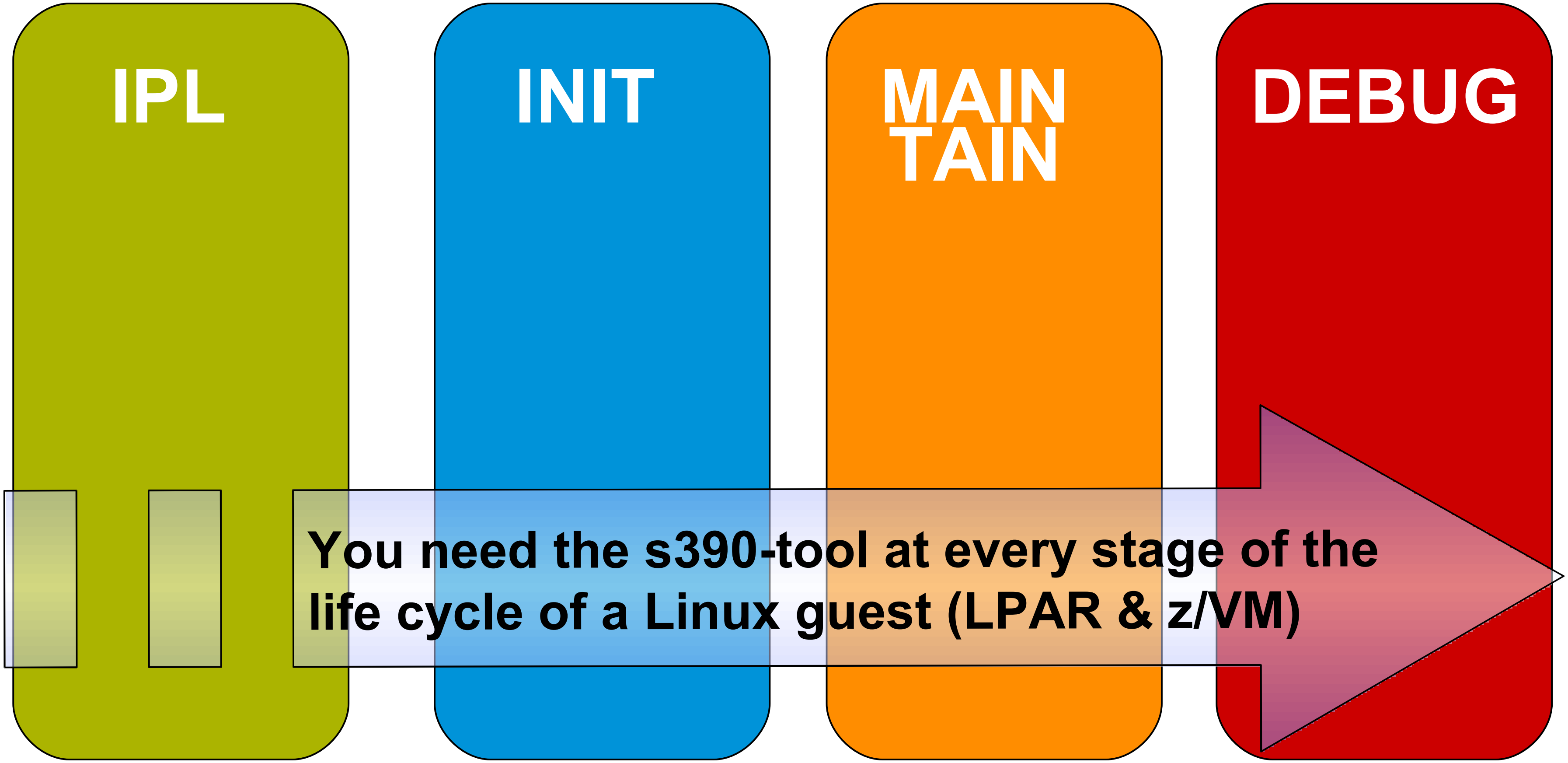
Hans-Joachim Picht, Linux on System z Initiatives & Linux Champion EMEA <hans@de.ibm.com>

s390-tools - In a Nutshell

The Swiss Army Knife for Linux on System z System Administration

Share Conference, Boston
Thursday, August 5, 2010: 3:00 PM-4:00 PM Room 306 (Hynes Convention Center)

The Linux on System z Life Cycle



IPL

INIT

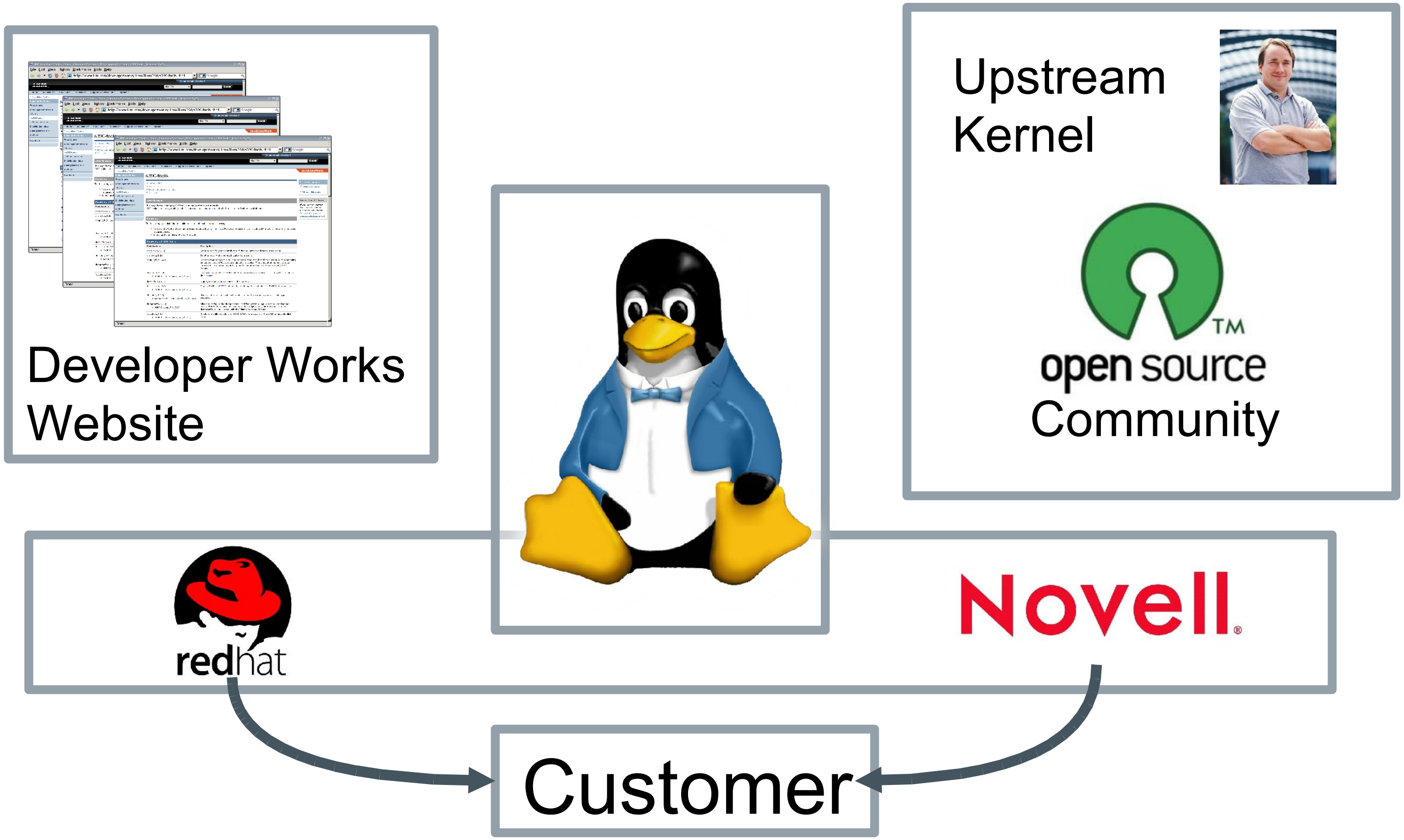
MAIN
TAIN

DEBUG

You need the s390-tool at every stage of the life cycle of a Linux guest (LPAR & z/VM)

IBM Linux on System z Development

IBM Linux on System z Development contributes in the following areas: Kernel, s390-tools, Open Source Tools (e.g. eclipse, ooprofile), GCC, GLIBC, Binutils



Developer Works Website

Upstream Kernel

open source Community

redhat

Novell.

Customer

....the code you use is the result of the efforts of an anonymous army of blue penguins involved in developing, testing, documenting, ...

What is the s390-tools package?

s390-tools is **the** essential set of Linux on System z system administration tools:

It contains everything from the boot loader to dump related tools for system crash analysis .

This software package is contained in all major (and IBM supported) distributions which support s390

RedHat Enterprise Linux 4 (s390-tools-1.3.2)
RedHat Enterprise Linux 5 (s390-tools-1.8.1 since RHEL 5.4)
SuSE Linux Enterprise Server 10 (s390-tools-1.6.3 since SLES 10 SP2)
SuSE Linux Enterprise Server 11 (s390-tools-1.8.0)

The latest version 1.10 will be released shortly.

Website: http://www.ibm.com/developerworks/linux/linux390/s390-tools.html

The Content

CHANGE

- chccwdev
- chchp
- chreipl
- chshut
- chzcrypt
- chmem

DISPLAY

- lscss
- lschp
- lsdasd
- lsluns
- lsqeth
- lsreipl
- lsshut
- lstape
- lszcrypt
- lszfcp
- lsmem

DASD

- dasdfmt
- dasdinfo
- dasdview
- fdasd
- tunedasd

MONITOR

- mon_fsstatd
- mon_procd
- ziomon

NETWORK

- ip_watcher
- osasnmpd
- qetharp
- qethconf

TAPE

- tape390_crypt
- tape390_display

DUMP & DEBUG

- dbginfo
- dumpconf
- zfcpdump
- zfcpdbf
- zgetdump
- scsi_logging_level

z/VM

- vmconvert
- vmcp
- vmur

MISC

- cpuplugd
- iucvconn
- iucvtty
- ts-shell

BOOT

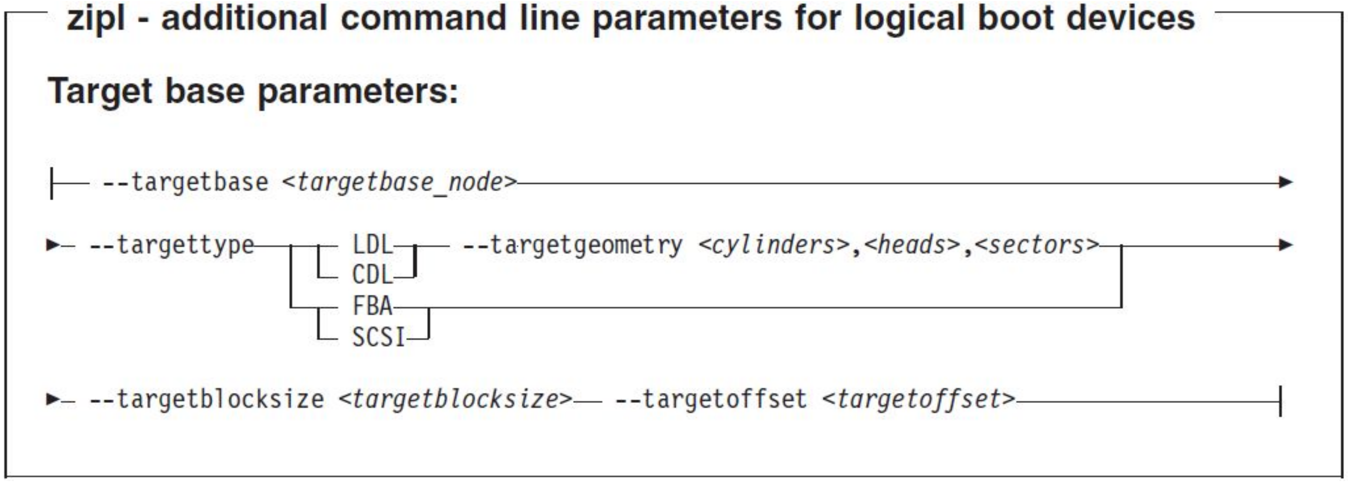
- zipl

Future Linux on System z Technology

Software which has already been developed and externally published – but is **not** yet available in any
Enterprise Linux Distribution

Support for device mapper devices: zipl (1.8.3)

Lately zipl allows installation of and booting from a boot record on logical devices, i.e. devices managed by device mapper, e.g. multipath devices.



zipl - additional command line parameters for logical boot devices

Target base parameters:

```
├── --targetbase <targetbase_node>──────────────────────────────────────────►

►── --targettype──┬─┬─ LDL─┬── --targetgeometry <cylinders>,<heads>,<sectors>─┬────►
                  │ └─ CDL─┘                                                   │
                  └─┬─ FBA──────────────────────────────────────────────────────┘
                    └─ SCSI─┘

►── --targetblocksize <targetblocksize>── --targetoffset <targetoffset>──────┤
```

- A physical device is of type DASD or SCSI
- All of the devices which contains the directory must be located on a single physical device (which may be mirrored or accessed through a multipath target) only linear, mirror and multipath targets are supported
- The boot directory is located on a device consisting of a single device-mapper target

Support for device mapper devices: zipl (cont'd)

Example for using additional parameters

```
root@larsson:~>  zipl --dry-run -i /boot/image -r
/boot/initrd -p /boot/parmfile  -t /boot --targetbase /
dev/dm-0 --targettype CDL --targetgeometry 3339,15,12
--targetblocksize=4096 --targetoffset 24 -V
```

Example section in the zipl configuration file

```
[boot5]
image=/boot/image
ramdisk=/boot/initrd
parmfile=/boot/parmfile
target=/boot
targetbase=253:0
targettype=CDL
targetgeometry=3339,15,12
targetblocksize=4096
targetoffset=24
```

Release 1.9.0

The s390-tool Release 1.9.0 was published on 2010-05-28

The following new features are included, besides changes to existing tools and bugfixes:

- **cmsfs-fuse:** Use the cmsfs-fuse command to read files stored on a z/VM CMS disk. The cmsfs-fuse file system translates the record-based EDF file system on the CMS disk to UNIX semantics. It is possible to mount a CMS disk and use common Linux tools to access the files on the disk.
- **lsmem:** Display the online status of the available memory.
- **chmem:** Set hotplug memory online or offline.
- **ttyrun:** Depending on your setup, Linux on System z might or might not provide a particular terminal or console. The ttyrun tool safely starts getty programs and prevents respawns through the init program, if a terminal is not available.

CMSFS user space file system support

- Allows to mount a z/VM minidisk to a Linux mount point
- z/VM minidisk needs to be in the enhanced disk format (EDF)
- The cmsfs fuse file system transparently integrates the files on the minidisk into the Linux VFS, no special command required

```
root@larsson:~> cmsfs-fuse /dev/dasde /mnt/cms
root@larsson:~> ls -la /mnt/fuse/PROFILE.EXEC
-r--r----- 1 root root 3360 Jun 26 2009
/mnt/fuse/PROFILE.EXEC
```

- By default no conversion is performed
  - Mount with '-t' to get automatic EBCDIC to ASCII conversion

```
root@larsson:~> cmsfs-fuse -t /dev/dasde /mnt/cms
```

- Write support is work in progress, almost completed
  - use “vi” to edit PROFILE.EXEC anyone ?
- Use fusermount to unmount the file system again

```
root@larsson:~> fusermount -u /mnt/cms
```

lsmem - Show online status information about memory blocks

The lsmem command lists the ranges of available memory with their online status.

- The listed memory blocks correspond to the memory block representation in sysfs.
- The command also shows the memory block size, the device size, and the amount of memory in online and offline state.

The output of this command, shows ranges of adjacent memory blocks with similar attributes.

```
root@larsson:~> lsmem
Address range                        Size (MB) State Removable Device
=========================================================================
0x0000000000000000-0x000000000fffffff 256       online   no    0
0x0000000010000000-0x000000002fffffff 512       online   yes   1-2
0x0000000030000000-0x000000003fffffff 256       online   no    3
0x0000000040000000-0x000000006fffffff 768       online   yes   4-6
0x0000000070000000-0x00000000ffffffff 2304      offline   -    7-15
Memory device size : 256 MB
Memory block size : 256 MB
Total online memory : 1792 MB
Total offline memory: 2304 MB
```

chmem - Set memory online or offline

The chmem command sets a particular size or range of memory online or offline.

- Setting memory online can fail if the hypervisor does not have enough memory left, for example because memory was overcommitted. Setting memory offline
- can fail if Linux cannot free the memory.
- If only part of the requested memory can be set online or offline, a message tells you how much memory was set online or offline instead of the requested amount.

This command requests 1024 MB of memory to be set online.

```
root@larsson:~> chmem --enable 1024
```

This command requests the memory range starting with 0x00000000e4000000 and ending with 0x00000000f3ffffff to be set offline.

```
root@larsson:~> chmem --disable 0x00000000e4000000-
0x00000000f3ffffff
```

hyptop - Display hypervisor performance data (1.10)

The hyptop command provides a dynamic real-time view of a hypervisor environment on System z.

- It works with both the z/VM and the LPAR PR/SM hypervisor.
- Depending on the available data it shows, for example, CPU and memory information about running LPARs or z/VM guest operating systems.

The following things are required to run hyptop:

- The debugfs file system must be mounted.
- The hyptop user must have read permission for the required debugfs files:
  - z/VM: <debugfs mount point>/s390_hypfs/diag_2fc
  - LPAR: <debugfs mount point>/s390_hypfs/diag_204
- To monitor all LPARs or z/VM guest operating systems of the hypervisor, your system must have additional permissions:
  - For z/VM: The guest must be class B.
  - For LPAR: On the HMC or SE security menu of the LPAR activation profile, select the Global performance data control checkbox.

hyptop - Display hypervisor performance data (cont.)

```
12:32:21 | CPU-T: UN(16) ?=help
system          #cpu cpu Cpu+ online memuse memmax wcur (str) (#) (%) (hm) (dhm) (GiB)
(GiB) (#)
T6360004        6    100.31    959:47    53:05:20  1.56     2.00     100
DTCVSW1         1    0.00      0:00      53:16:42  0.01     0.03     100
T6360002        6    0.00      166:26    40:19:18  1.87     2.00     100
OPERATOR        1    0.00      0:00      53:16:42  0.00     0.03     100
T6360008        2    0.00      0:37      30:22:55  0.32     0.75     100
T6360003        6    0.00      3700:57   53:03:09  4.00     4.00     100
NSLCF1          1    0.00      0:02      53:16:41  0.03     0.25     500
PERFSVM         1    0.00      0:53      2:21:12   0.04     0.06     0
TCPIP           1    0.00      0:01      53:16:42  0.01     0.12     3000
DIRMAINT
DTCVSW2
RACFVM
75 101.5
```

```
12:15:51 MYLPAR CPU-T: IFL(18) CP(3) UN(2) ?=help
cpuid type cpu mgm visual
(#) (str) (%) (%) (vis)
0 IFL 99.84 0.02 |########################################
1 IFL 99.85 0.02 |########################################
2 IFL 0.00 0.00 |
3 IFL 0.00 0.00 |
=:V:N 199.69 0.04
```

Current Linux on System z Technology

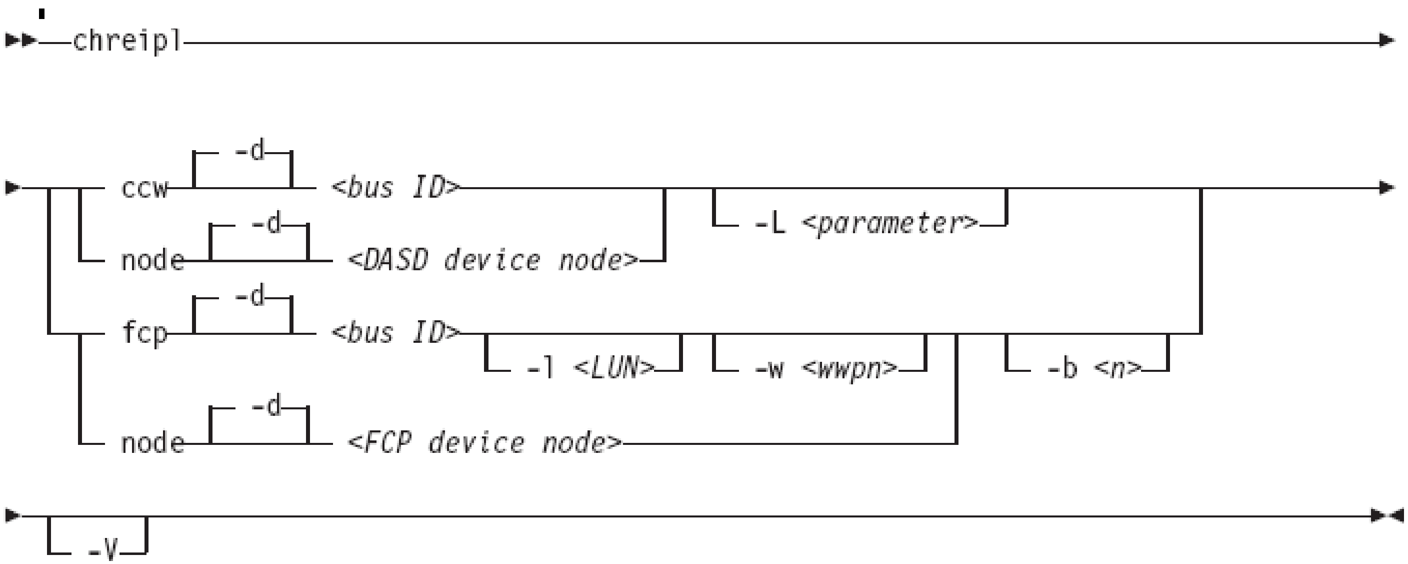
Features & Functionality contained in the Novell & Red Hat Distributions

Shutdown action tools

chreipl: Configure a disk or change a an entry in the Boot menu for the next boot

11

5.4



chreipl

ccw -d <bus ID>

node -d <DASD device node>

-L <parameter>

fcp -d <bus ID>

-l <LUN>

-w <wwpn>

-b <n>

node -d <FCP device node>

-V

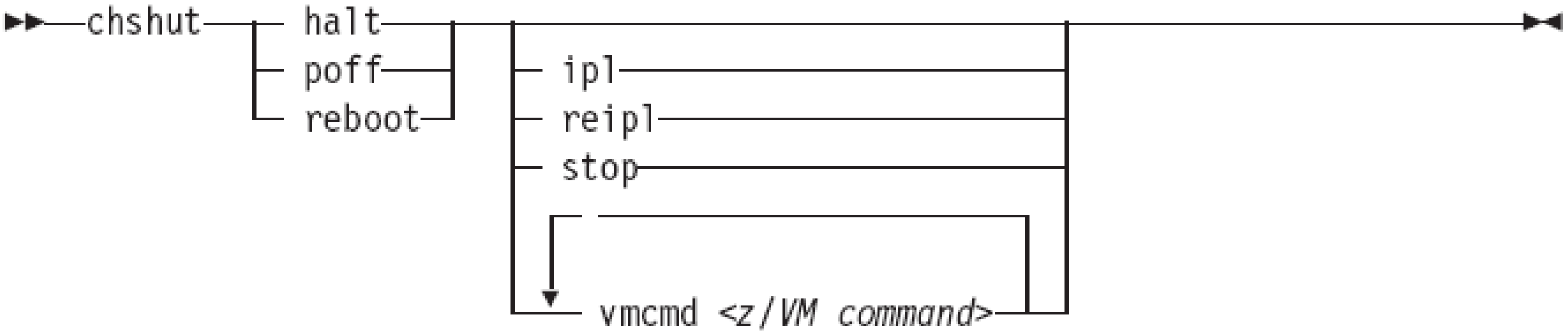
```
root@larsson:~> chreipl node /dev/dasda
root@larsson:~> chreipl node /dev/sda
root@larsson:~> chreipl ccw -d 0.0.7e78 -L 1
root@larsson:~> chreipl fcp --wwpn 0x500507630300c562 \
--lun 0x401040B300000000 -d 0.0.1700
```

Shutdown action tools (cont.)

chshut: Change the entries in */sys/firmware* to configure the shutdown behaviour

11

5.4



```
root@larsson:~> chshut halt ipl
root@larsson:~> chshut halt vmcmd LOGOFF
root@larsson:~> chshut poff vmcmd "MSG MASTER Going
down" \ vmcmd "LOGOFF"
```

Shutdown action tools (cont.)

11

5.4

lsreipl: command to see from which device your system will boot after you issue the reboot command. Further you can query the system for information about the current boot device.

```
root@larsson:~> lsreipl
Re-IPL type:   ccw
Device:        0.0.4bb8
Loadparm:
root@larsson:~> lsreipl -i
```

lsshut: command to see what the system should do in one of the following states.

```
root@larsson:~> lsshut
Trigger          Action
========================
Halt              stop
Panic             stop
Power off         stop
Reboot            reipl
```

lsluns

Use the **lsluns** command to discover and scan LUNs in Fibre Channel Storage Area Networks (SANs).

11

5.4

This example shows all LUNs for port 0x500507630300c562:

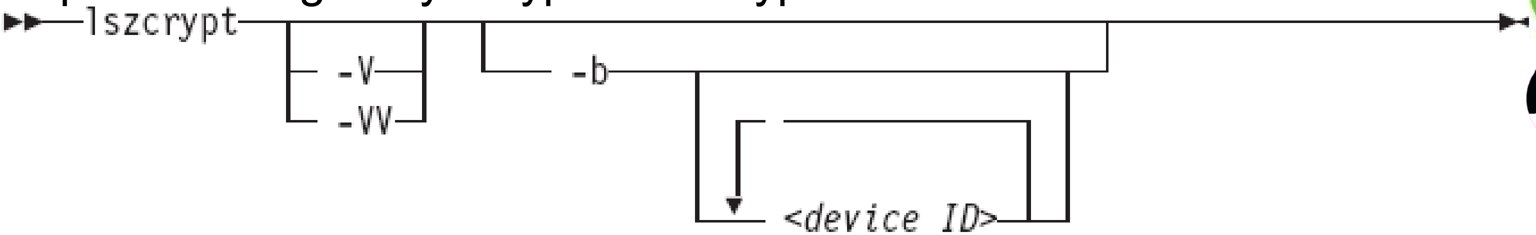
```
root@larsson:~> lsluns --port 0x500507630300c562
Scanning for LUNs on adapter 0.0.5922
at port 0x500507630300c562:
0x4010400000000000
0x4010400100000000
[...]
```

This example shows all LUNs for adapter 0.0.5922:

```
root@larsson:~> lsluns -c 0.0.5922
at port 0x500507630300c562:
0x4010400000000000
[...]
at port 0x500507630303c562:
0x4010400000000000
[...]
```

lszcrypt

Use the **lszcrypt** command to display information about cryptographic adapters managed by zcrypt and zcrypt's AP bus attributes



11

5.4

- To display card type and online status of all available cryptographic adapters:

```
root@larsson:~> lszcrypt -V
```

- To display card type, online status, hardware card type, hardware queue depth, and request count for cryptographic adapters 0, 1, 10, and 12

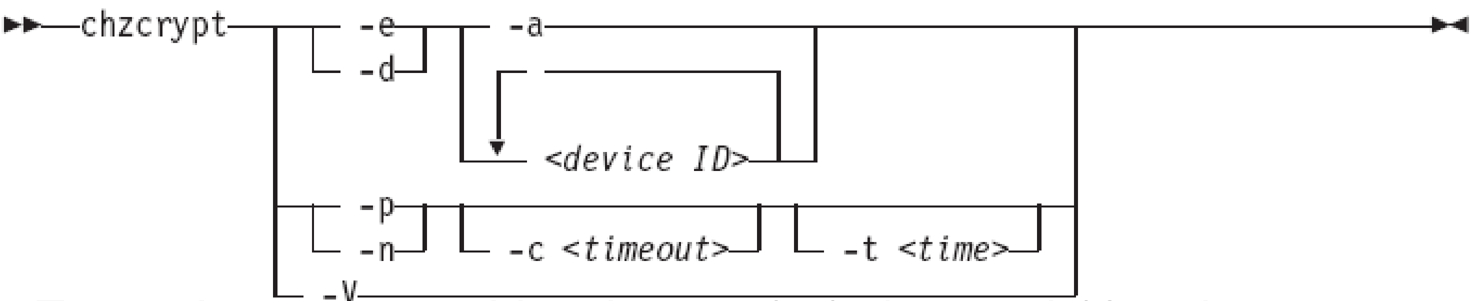
```
root@larsson:~>  lszcrypt -VV 0 1 10 12
```

- To display AP bus information:

```
root@larsson:~> lszcrypt -b
```

chzcrypt

Use the **chzcrypt** command to configure cryptographic adapters managed by zcrypt and modify zcrypt’s AP bus attributes.



To set the cryptographic adapters 0, 1, 4, 5, and 12 online:

```
root@larsson:~>  chzcrypt -e 0 1 4 5 12
```

To set all available cryptographic adapters offline:

```
root@larsson:~>  chzcrypt -d -a
```

To set the configuration timer for re-scanning the AP bus to 60

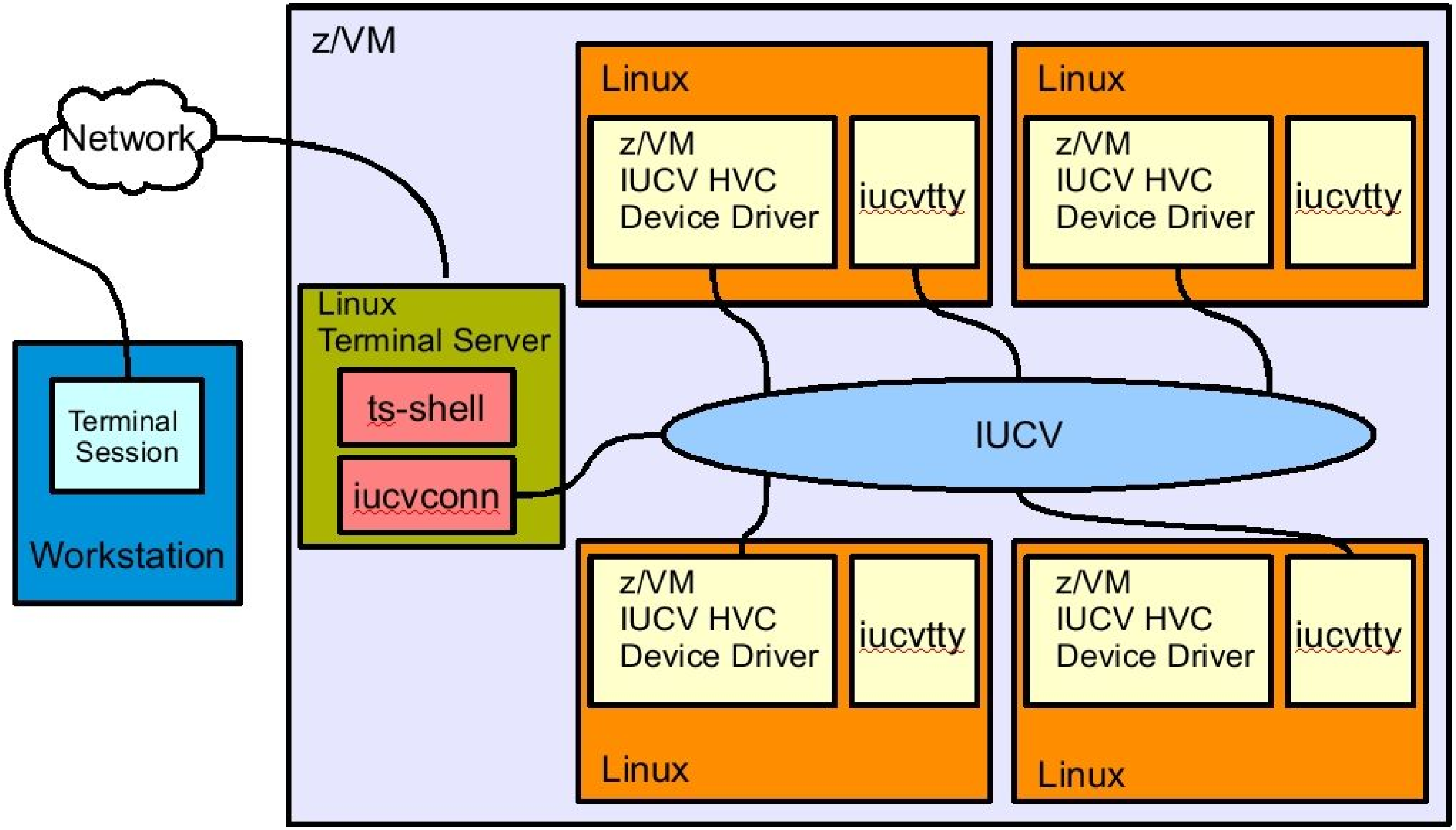
```
root@larsson:~>  chzcrypt -c 60 -n
```

IUCV Terminals

11

5.4

- Full-screen terminal access to Linux guest operating systems on the same z/VM
- Access Linux instances with no external network because IUCV is independent from TCP/IP



IUCV terminal applications

11

5.4

- The IUCV terminal applications consist of:
  - **iucvconn** – Start terminal connection over IUCV
  - **iucvtty** – Allow remote logins over IUCV
  - **ts-shell** – Login shell for terminal servers over IUCV

- Terminal access over IUCV is provided by:
  - **iucvtty**, or
  - z/VM IUCV hypervisor console device driver (Linux kernel)

IUCV terminal applications – examples

11

5.4

Using the iucvconn program: To access the first z/VM IUCV HVC terminal on the Linux instance in z/VM guest LNXSYS02

```
root@larsson:~> iucvconn LNXSYS02 lnxhvc0
```

To create a transcript of the terminal session to the Linux instance in z/VM guest LNXSYS99

```
root@larsson:~> iucvconn -s ~/transcripts/lnxsys99 LNXSYS99
lnxhvc0
```

Using the iucvtty program: To allow remote logins using the terminal identifier „lnxterm“

```
root@larsson:~> iucvtty lnxterm
```

To access the „lnxterm“ terminal on the Linux instance in z/VM guest LNXSYS01

```
root@larsson:~>  iucvconn LNXSYS01 lnxterm
```

To use /sbin/sulogin instead of /bin/login for terminal “suterm”

```
root@larsson:~>  iucvtty suterm -- /sbin/sulogin
```

cpuplugd

10.2

5.3

Use the **cpuplugd** command to:

- Enable or disable CPUs based on a set of rules. This increases the performance of single threaded applications within a z/VM or LPAR environment with multiple CPUs.
- The rules can incorporate certain system load variables.
- Manage memory under z/VM.

- Configuration file: */etc/sysconfig/cpuplugd*
- Init-Script: */etc/init.d/cpuplugd* {start, stop, restart}

cpuplugd: Example Configuration

```
UPDATE="60"

CPU_MIN="2"
CPU_MAX="10"

HOTPLUG = "(loadavg > onumcpus +0.75) & (idle < 10.0)"
HOTUNPLUG = "(loadavg < onumcpus -0.25) | (idle > 50)"


CMM_MIN="0"
CMM_MAX="8192"
CMM_INC="256"

MEMPLUG = "swaprate > freemem+10 & freemem+10 < apcr"
MEMUNPLUG = "swaprate > freemem + 10000"
```

dbginfo.sh

dbginfo.sh is a script to collect various system related files, for debugging purposes.

- It generates a tar-archive which can be attached to PMRs / Bugzilla entries
- It is similar to the RedHat tools sosreport & the Novell support-utils

```
root@larsson:~>  dbginfo.sh
Create target directory /tmp/DBGINFO-2010-02-25-22-06-20-t6345057
Change to target directory /tmp/DBGINFO-2010-02-25-22-06-20-t6345057
Get procfs entries
Saving runtime information into runtime.out
Get file list of /sys
Get entries of /sys
[...]
```

Please use the data from this tool is you open a Bugzilla (Novell/ RedHat) or a PMR

Multi Volume Dump

zipl can now dump to multiple DASDs. It is now possible to dump system images, which are larger than a single DASD.

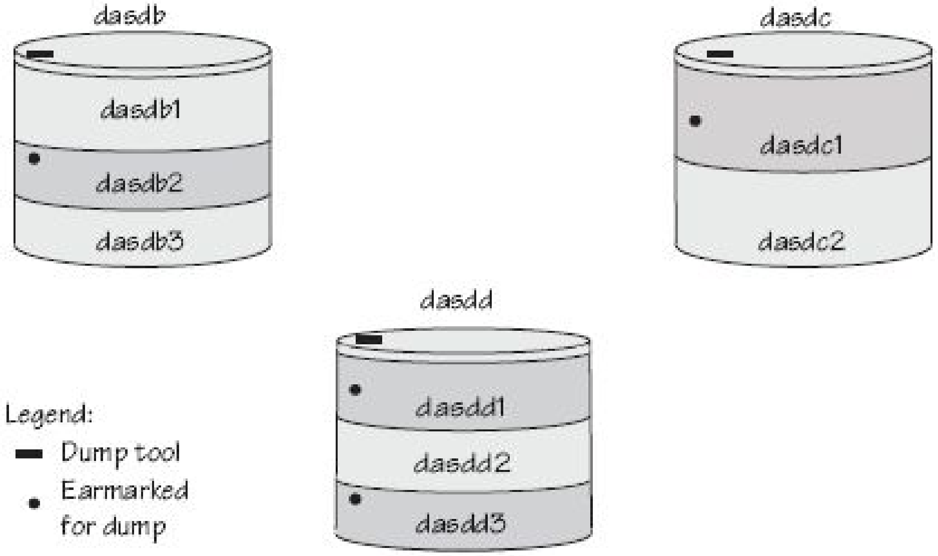
10.3

5.4

You can specify up to 32 ECKD DASD partitions for a multi-volume dump

What are dumps good for?

- Full snapshot of system state taken at any point in time (e.g. after a system has crashed, of or a running system)
- Can be used to analyse system state beyond messages written to the syslog
- Internal data structures not exported to anywhere



Obtain message, which have not been written to the syslog due to a crash

Multi Volume Dump

How to prepare a set of ECKD DASD devices for a multi-volume dump?
bit systems only).

10.3

5.4

- We use two DASDs in this example:

```
root@larsson:~> dasdfmt -f /dev/dasdc -b 4096
root@larsson:~> dasdfmt -f /dev/dasdd -b 4096
```

- Create the partitions with fdasd. The sum of the partition sizes must be sufficiently large (the memory size + 10 MB):

```
root@larsson:~> fdasd /dev/dasdc
root@larsson:~> fdasd /dev/dasdd
```

- Create a file called sample_dump_conf containing the device nodes (e.g. /dev/dasda1) of the two partitions, separated by one or more line feed characters
- Prepare the volumes using the zipl command.

```
root@larsson:~>  zipl -M sample_dump_conf
[...]
```

How to obtain a dump

To obtain a dump with the multi-volume DASD dump tool, perform the following steps:

- Stop all CPUs, Store status on the IPL CPU.
- IPL the dump tool using one of the prepared volumes, either 4711 or 4712.
- After the dump tool is IPLed, you'll see a messages that indicates the progress of the dump. Then you can IPL Linux again

```
==> cp cpu all stop
==> cp cpu 0 store status
==> cp ipl 4711
```

- Copying a multi-volume dump to a file
- Use zgetdump command without any option to copy the dump parts to a file:

```
root@larsson:~>  zgetdump /dev/dasdc > mv_dump_file
```

How to obtain information about a multi volume dumps

Display information on the involved volumes:

```
root@larsson:~>  zgetdump -d /dev/dasdc
'/dev/dasdc' is part of Version 1 multi-volume dump,which is
spread along the following DASD volumes:
0.0.4711 (online, valid)
0.0.4712 (online, valid)
[...]
```

Display information about the dump itself:

```
root@larsson:~>  zgetdump -i /dev/dasdc
Dump device: /dev/dasdc
>>>  Dump header information  <<<
Dump created on: Thu Feb  25 15:12:41 2010
[...]
Multi-volume dump: Disk 1 (of 2)
Reading dump contents from
0.0.4711...............................
Dump ended on:   Thu Feb  25 15:12:52 2010
Dump End Marker found: this dump is valid.
```

dumpconf

10.1

5.4

The dumpconf tool configures a dump device that is used for automatic dump in case of a kernel panic.

- The command can be installed as service script under `/etc/init.d/dumpconf` or can be called manually.
- Start service: `service dumpconf start`
- It reads the configuration file `/etc/sysconfig/dumpconf.`
- Example configuration for CCW dump device (DASD) and reipl after dump:

```
ON_PANIC=dump_reipl
DUMP_TYPE=ccw
DEVICE=0.0.4711
```

dumpconf (cont.)

Example configuration for FCP dump device (SCSI disk):

```
ON_PANIC=dump
DUMP_TYPE=fcp
DEVICE=0.0.4714
WWPN=0x5005076303004712
LUN=0x4047401300000000
BOOTPROG=0
BR_LBA=0
```

Example configuration for re-IPL without taking a dump, if a kernel panic occurs:

```
ON_PANIC=reipl
```

Example of executing a CP command, and rebooting from device 4711 if a kernel panic occurs (MASTER is a VM Guest Name):

```
ON_PANIC=vmcmd
VMCMD_1="MSG MASTER Starting VMDUMP"
VMCMD_2="VMDUMP"
VMCMD_3="IPL 4711"
```

Dump Tools Summary

Tool	Stand alone tools			**VMDUMP**
	DASD	**Tape**	**SCSI**	
Environment	VM&LPAR		LPAR	VM
Preparation	Zipl -d /dev/<dump_dev>		Mkdir /dumps/mydumps zipl -D /dev/sda1 …	---
Creation	Stop CPU & Store status ipl <dump_dev_CUU>			Vmdump
Dump medium	ECKD or FBA	Tape cartridges	LINUX file system on a SCSI disk	VM reader
Copy to filesystem	Zgetdump /dev/<dump_dev> > dump_file		---	Dumpload ftp … vmconvert …
Viewing	Lcrash or crash			

See “Using the dump tools” book at http://www.ibm.com/developerworks/linux/linux390/

vmcp

Using the z/VM CP interface device driver (vmcp), you can send control program (CP) commands to the VM hypervisor and display VM’s response.

```
root@larsson:~> modprobe vmcp
root@larsson:~> vmcp "q dasd"|grep T6345057
DASD 4DE0 ATTACHED TO T6345057 4DE0 R/W 0X4DE0
DASD 4DE1 ATTACHED TO T6345057 4DE1 R/W 0X4DE1
DASD 4DE2 ATTACHED TO T6345057 4DE2 R/W 0X4DE
DASD 4DE3 ATTACHED TO T6345057 4DE3 R/W 0X4DE3
```

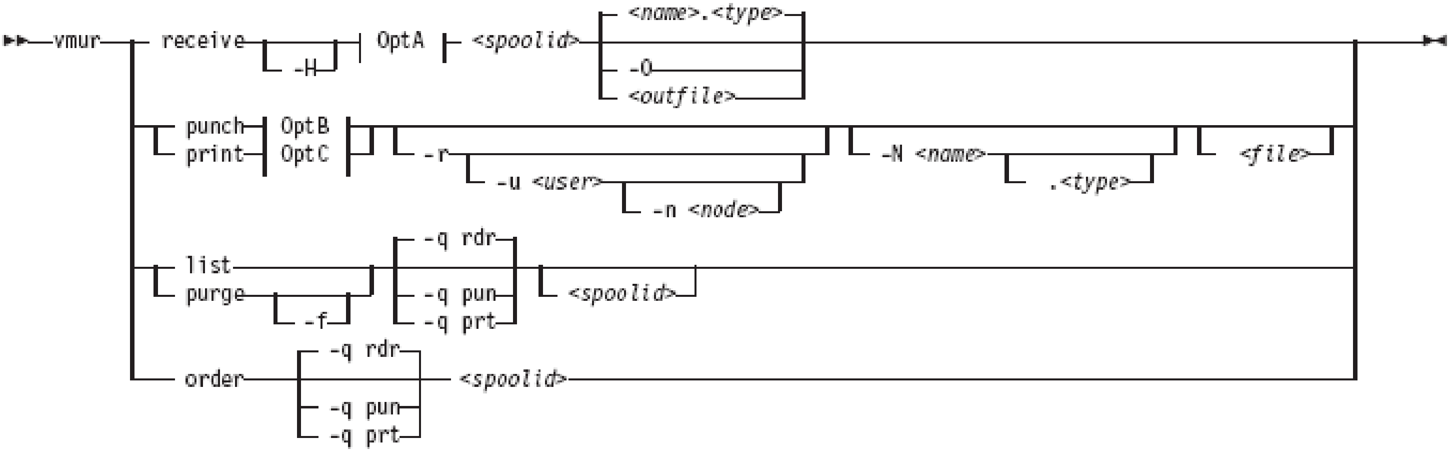
vmur

10.1

5.2

The **vmur** command provides all functions required to work with z/VM spool file queues:

- Receive: Read data from the z/VM reader file queue
- Punch or print: Write data to the z/VM punch or printer file queue and transfer it to another user’s virtual reader, optionally on a remote z/VM node.
- List: Display detailed information about one or all files on the specified spool file queue.
- Purge: Remove one or all files on the specified spool file queue.
- Order: Position a file at the top of the specified spool file queue.



vmur: Produce and read Linux guest machine dump

Produce guest machine dump:

```
root@larsson:~>  vmcp vmdump
```

Find spool ID of VMDUMP spool file in the output of the `vmur li` command:

```
root@larsson:~> vmur li
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE
DIST T6360025 0463 V DMP 00020222 001 NONE 06/11 15:07:42
VMDUMP FILE T6360025
```

Move vmdump file to top of reader queue with the vmur order command:

```
root@larsson:~>  vmur or 463
```

Read and convert the vmdump file to a file on the Linux file system in the current working directory and close the virtual reader

```
root@larsson:~> chccwdev -e 000c
root@larsson:~> vmconvert /dev/vmrdr-0.0.000c linux_dump
root@larsson:~> vmcp cl c
```

vmur: Log and read Linux guest machine console

Begin console spooling:

```
root@larsson:~>  vmcp sp cons start
```

Produce output to VM console (for example, with CP TRACE).
Close the console file and transfer it to the reader queue, find the spool ID behind the FILE keyword in the corresponding CP message.

```
root@larsson:~>   vmcp sp cons clo \* rdr
RDR FILE 0398 SENT FROM T6360025 CON WAS 0398 RECS 1872
CPY 001 T NOHOLD NOKEEP
```

Read the guest machine console file into a file on the Linux file system in the current working directory:

```
root@larsson:~>  chccwdev -e 000c
root@larsson:~>  vmur re -t 398 linux_cons
```

vmur: Prepare z/VM reader to IPL Linux image

Send parmfile to VM punch and transfer it to the reader queue and find the parmfile spool id message

```
root@larsson:~>  vmur pun -r /boot/parmfile
[...]
Reader file with spoolid 0465 created.
```

Send image to VM punch and transfer it to reader queue:

```
root@larsson:~> vmur pun -r /boot/vmlinuz -N image
```

Move image to first and parmfile to the second position in the reader queue:

```
root@larsson:~>  vmur or 465
root@larsson:~>  vmur or 466
```

Prepare re-IPL from the VM reader and boot the Linux image

```
root@larsson:~>  chreipl ccw 0.0.000c
root@larsson:~>  reboot
```

cio_ignore

When a Linux on System z instance boots, it senses and analyses all available devices. You can use the cio_ignore kernel parameter to specify a list of devices that are to be ignored.

The following applies to ignored devices:

- Ignored devices are not sensed and analyzed. The device cannot be used unless it has been analyzed.
- Ignored devices are not represented in sysfs.
- Ignored devices do not occupy storage in the kernel.
- The subchannel to which an ignored device is attached is treated as if no device were attached.
- cio_ignore might hide essential devices such as the console under z/VM. The console is typically device number 0.0.0009.
-

This example specifies that all devices in the range 0.0.b100 through 0.0.b1ff, and the device 0.0.a100 are to be ignored.

```
cio_ignore=0.0.b100-0.0.b1ff,0.0.a100
```

cio_ignore (cont.)

Display ignored devices:

```
root@larsson:~>  cat /proc/cio_ignore
0.0.0000-0.0.78ff
0.0.f503-0.0.ffff
```

Free a individual device from the ignore list

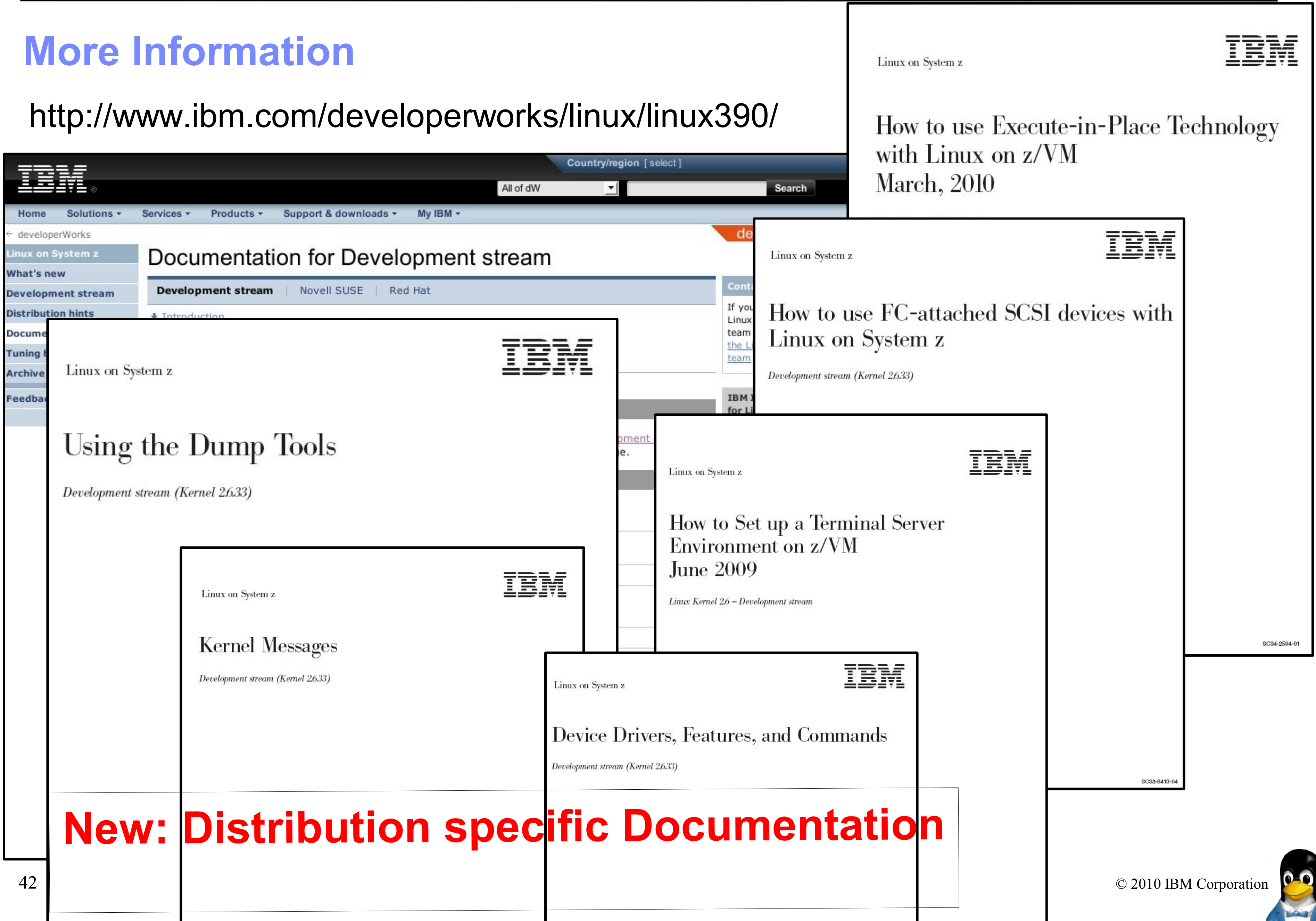
```
root@larsson:~>  echo free 0.0.4711 >/proc/cio_ignore
```

Free all devices from the ignore list

```
root@larsson:~>  echo free all >/proc/cio_ignore
```

More Information

http://www.ibm.com/developerworks/linux/linux390/



Documentation for Development stream

Development stream | Novell SUSE | Red Hat

Introduction

Linux on System z

How to use Execute-in-Place Technology with Linux on z/VM
March, 2010

SC34-2594-01

Linux on System z

How to use FC-attached SCSI devices with Linux on System z

Development stream (Kernel 2.6.33)

SC33-8413-04

Linux on System z

Using the Dump Tools

Development stream (Kernel 2.6.33)

Linux on System z

How to Set up a Terminal Server Environment on z/VM
June 2009

Linux Kernel 2.6 – Development stream

Linux on System z

Kernel Messages

Development stream (Kernel 2.6.33)

Linux on System z

Device Drivers, Features, and Commands

Development stream (Kernel 2.6.33)

New: Distribution specific Documentation

Your Linux on System z Requirements?

Are you missing a certain feature, functionality or tool? **We'd love to hear from you!**

We will evaluate each request and (hopefully) develop the additional functionality you need.

Send your input to hans@de.ibm.com

Questions?

Hans-Joachim Picht

Linux Technology Center

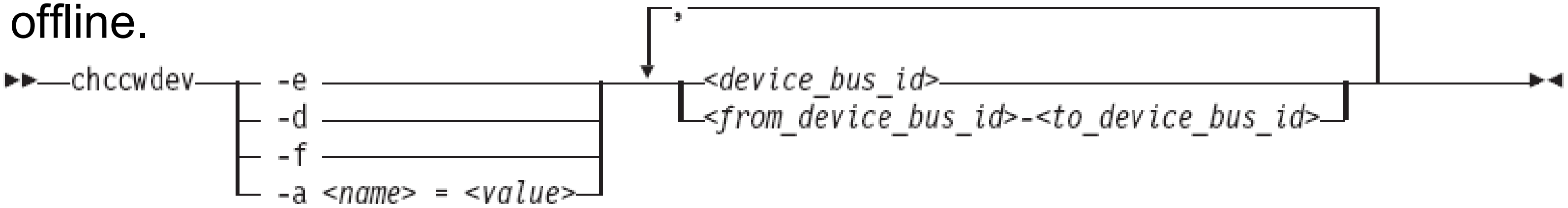
IBM Deutschland Research
& Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-1810
Mobile +49 (0)175 - 1629201
hans@de.ibm.com

Appendix

chccwdev

Use the **chccwdev** command is used to set CCW devices online or offline.



-e or --online sets the device online.
-d or --offline sets the device offline.
-f or --forceonline forces a boxed device online, if this is supported by the device driver.

- To set a CCW device 0.0.b100 online issue:

```
root@larsson:~>  chccwdev -e 0.0.b100
```

- To set all CCW devices in the range 0.0.b200 through 0.0.b2ff online issue:

```
root@larsson:~>  chccwdev -e 0.0.b200-0.0.b2ff
```

- To set a CCW device 0.0.b100 and all CCW devices in the range 0.0.b200 through 0.0.b2ff offline issue:

```
root@larsson:~>  chccwdev -d 0.0.b100,0.0.b200-0.0.b2ff
```

DASD low level format

dasdfmt formats a DASD (ECKD) disk to prepare it for usage with Linux on System z

```
root@larsson:~> dasdfmt -d cdl -b 4096 -f /dev/dasdb
Drive Geometry: 10017 Cylinders * 15 Heads =  150255 Tracks

I am going to format the device /dev/dasdb in the following way:
   Device number of device : 0xec27
   Labelling device        : yes
   Disk label              : VOL1
   Disk identifier         : 0XEC27
   Extent start (trk no)   : 0
   Extent end (trk no)     : 150254
   Compatible Disk Layout  : yes
   Blocksize               : 4096

--->> ATTENTION! <<---
All data of that device will be lost.
Type "yes" to continue, no will leave the disk untouched: yes
Formatting the device. This may take a while (get yourself a coffee).
```

DASD: Partitioning

Compared to other architectures, Linux on System z makes use of its own partitioning tool for DASD devices.

- The common Linux tool fdisk can **not** be used in this environment!

Nevertheless the handling is similar.

- The system is limited to 3 partitions per disk when using DASD

```
root@larsson:~> fdasd /dev/dasdb
reading volume label ..: VOL1
reading vtoc ..........: ok

Command action
   m   print this menu
   p   print the partition table
   n   add a new partition
   d   delete a partition
   v   change volume serial
   t   change partition type
   r   re-create VTOC and delete all partitions
   u   re-create VTOC re-using existing partition sizes
   s   show mapping (partition number - data set name)
   q   quit without saving changes
   w   write table to disk and exit

Command (m for help):
```

DASD: Partitioning (cont'd)

To create a partition:

```
root@larsson:~> fdasd /dev/dasdb
[...]
Command (m for help): n
First track (1 track = 48 KByte) ([2]-150254):
Using default value 2
Last track or +size[c|k|M] (2-[150254]):
Using default value 150254

Command (m for help): p

Disk /dev/dasdb:
  cylinders ............: 10017
  tracks per cylinder ..: 15
  blocks per track .....: 12
  bytes per block ......: 4096
  volume label .........: VOL1
  volume serial ........: 0XEC27
  max partitions .......: 3

 ------------------------------- tracks -------------------------------
               Device      start      end   length   Id  System
          /dev/dasdb1          2   150254   150253    1  Linux native
```

DASD: Partitioning (cont'd)

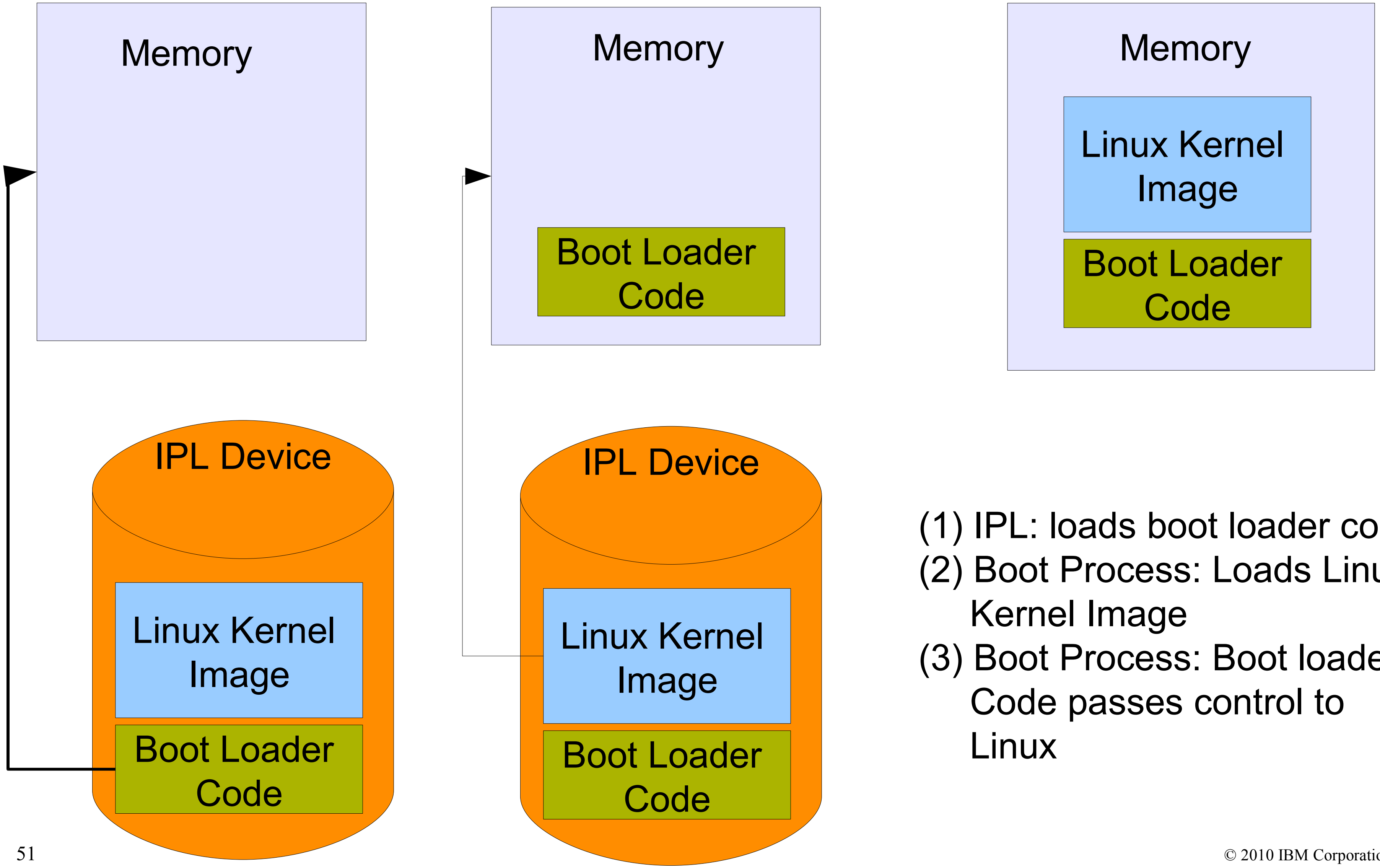
Your configuration is not completed before you write the changes to the disk

```
root@larsson:~> fdasd /dev/dasdb
[...]
Command (m for help): w
writing VTOC...
rereading partition table...
```

Now we have a new device partition (e.g. /dev/dasdb1) which can be used as any other Linux disk

```
root@larsson:~> mke2fs -j /dev/dasdb1
mke2fs 1.41.4 (27-Jan-2009)
[...]
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 28 mounts or 180 days,
whichever comes first.  Use tune2fs -c or -i to override.
```

Linux IPL (Initial Program Load)



(1) IPL: loads boot loader code
(2) Boot Process: Loads Linux Kernel Image
(3) Boot Process: Boot loader Code passes control to Linux

The zipl Bootmanager

- Zipl is the default bootmanager for Linux on System z
- It writes a bootloader to DASD or zFCP-attached SCSI disk
- Configuration file: `/etc/zipl.conf`
- zipl command must be executed after
  - altering the configuration
  - altering files referenced in configuration, e.g. initrd
- Choosing a kernel from a multi-boot configuration:
  - LPAR HMC operating system messages console
    - <number of kernel to boot>
  - z/VM 3270 console
    - #cp vi vmsg <number>

Example /etc/zipl.conf configuration file

```
[defaultboot]
defaultmenu=menu
[2.6.25]
        image=/boot/vmlinuz-2.6.25
        ramdisk=/boot/initrd-2.6.25.img
        target=/boot/
        parameters="root=/dev/disk/by-path/ccw-0.0.beef TERM=dumb"
[...]
:menu
target = "/boot"
1 = "2.6.25"
2 = "2.6.27"
default = 2
prompt = 1
timeout = 10
```

Trademarks & Disclaimer

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml: AS/400, DB2, e-business logo, ESCON, eServer, FICON, IBM, IBM Logo, iSeries, MVS, OS/390, pSeries, RS/6000, S/390, System Storage, System z9, VM/ESA, VSE/ESA, WebSphere, xSeries, z/OS, zSeries, z/VM.

The following are trademarks or registered trademarks of other companies

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries. LINUX is a registered trademark of Linux Torvalds in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC. Intel is a registered trademark of Intel Corporation. * All other products may be trademarks or registered trademarks of their respective companies.

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

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography. References in this document to IBM products or services do not imply that IBM intends to make them available in every country. Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use. The 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.